National University of Ireland, Galway
Department of Electronic Engineering

A thesis submitted to
National University of Ireland, Galway
for the degree of
DOCTOR OF PHILOSOPHY (PhD.)

Supervisor: Dr. Peter Corcoran

Advances in Automated Image Categorization: Sorting Images using Person Recognition Techniques

Gabriel Nicolae Costache

Galway, Ireland, June 2007

Contents

List of Figures

List of Tables

Acknowledgements

This thesis is the result of 3 years of work and many years of preparation, whereby I have been accompanied and supported by many people. It is a pleasant aspect that I have now the opportunity to express my gratitude for all of them.

First of all, I would like to express my deep and sincere gratitude to my supervisor, Dr. Peter Corcoran, NUIG, for supporting this work with ideas, enthusiasm, criticism and for pushing me when I needed to be pushed. His support in writing this thesis is invaluable.

I am deeply grateful to my first mentor, Prof. Inge Gavat from Politehnica University of Bucharest, Romania, for introducing me into the wonderful world of Signal Processing and Pattern Recognition research back when I was an undergraduate.

To my research colleagues: Rhys, who is working with me on the project since day one and helped me many times, Mircea, AlexC and Frank, thanks for creating a good working environment, for your comments and support.

Special thanks for the people in Fotonation for technical and financial support when I needed, and also for allowing me to continue the research into this subject after the thesis is finished: Eran, Petronel, Turlough, AlexD (for proofreading the thesis and advice), Adi, Lale and Sudhi (Sudhi helped me to implement one of the applications in the thesis).

For funding this research I would like to also thank Enterprise Ireland.

I would also like to thank my parents and my big brother, Marius, for creating an environment in which following this path seemed so natural.

And last, but certainly not least, I thank my better half, Claudia, for unconditioned support during this thesis and before. Her love and care keeps me going forward.

Thank you all!

Abstract

The core problem addressed by this thesis is to provide practical tools for automatic sorting and cataloging of a typical consumer collection of digital images. The thesis presents a complete system solution, comprising (i) automated detection of face regions in images; (ii) multiple automated face recognition modules; (iii) automated colour and texture analysis of regions peripheral to the detected face regions; (iv) a decision fusion module combining the outputs of each recognition or peripheral region analysis module and enabling an output measure of similarity between each detected face regions and a user selected reference. Each system component is implemented and tested independently. The complete system is then tested and initial results indicate that the combined performance of two independent face recognition modules (DCT and PCA) offer measurable improvement over a single recognition module when applied to typical consumer collections of images. The analysis of peripheral regions using the colour correlogram is shown to further improve the accuracy of face recognition based modules and to enable more granular sorting of the images containing a particular individual based on distinctive hair features or clothing. Techniques to improve the robustness of the system to variations in illumination and facial pose are investigated, tested and verified. A technique to significantly accelerate the retraining of basis functions f or PCA-based face recognition is presented with initial test results. Several working computer and Web applications based on, and illustrating features of the core system components are described and documented in detail.

Chapter

1

Introduction and Overview

Digital photography continues to be one of the consumer electronics industry's recent success stories. Even today, after more than a decade of continuous growth, the market and associated technologies continue to evolve rapidly. However, as users switch from conventional to digital photography, they find themselves with rapidly growing collections of digital images. As the price of electronic storage equipment has dropped significantly over the same timeframe, and continues to do so, consumers are encouraged to take more and more pictures to ensure they capture those "magic moments". But few consumers have the time, personal discipline or technical knowledge to manually catalog and organize these growing personal image collections in a manner which will enable them to easily retrieve images in the future.

This is the core problem addressed by this thesis - to provide practical tools for automatic sorting and cataloging of a typical consumer collection of digital images. The key term here is "automatic"; the goal is to perform initial sorting of the images with minimal intervention from the user.

In this work we are inspired from the human way of browsing images, which focuses primarily on the people present in an image: children, family members, friends or colleagues. This is typically how people organize and review printed images. Thus the principle pattern that will be used for organizing pictures should be based on the persons that are present in each image. Typically, when we want to retrieve images from the past, we look for certain people or certain events where the same people were present. Thus, it is intuitive to use the information specific to each person to sort and manage collections of images.

1.1 Background

The problem of automatic sorting of images is normally addressed by data processing techniques drawn from Content Based Image Retrieval (CBIR) systems. These systems sort images according to the level of similarity between either global or local properties. Global properties are those properties computed by looking at an entire image as a pattern, whereas local properties are those computed analyzing only portions of an image.

The system proposed in this thesis analyzes portions of an entire image in order to determine criteria for sorting. In particular, we employ regions of an image that are associated with the persons who are present in that image. Thus our system can be considered as a subset of the CBIR family of image sorting and retrieval systems. However, as we shall see shortly, the particular problem that we address in this work differs in subtle ways from other prior work in the field of person and face recognition. In turn this problem has unique requirements which have led us to adopt new approaches in order to address these.

The problem to be addressed in this thesis has two principle aspects: firstly we must be able to reliably detect persons in an image and secondly we must be able to extract detailed patterns from each detected person and to record this information in a compact format to enable future retrieval of the image.

The first of these problems is partly outside the scope of our research as the detection and tracking of faces in images and image sequences is a topic well suited to a thesis in its own right. Fortunately there has been significant recent progress in this field [113, 118] and we have been able to solve much of this problem through the implementation of known techniques and combining these with other research activities which are ongoing within the Consumer Electronics Research Group at NUIG. We will, however, discuss certain aspects of this problem in detail as some refinements and enhancements to known techniques were necessitated by certain approaches to the second problem and it is useful to describe and record these details in the context of this thesis.

Now because we differentiate between people from their faces it is intuitive to think that the face region within an image contains the most important information and data patterns which can be associated with that person, and subsequently employed for both recognition and classification purposes. Yet while it is very easy for us to distinguish between people just by looking at their faces, for a computer this becomes a face recognition problem, a field of research which has been problematic for many years. Even today, with significant investment during the last 5 years (i.e. since the events of Sept 11th 2001) in advanced face recognition technologies, there are still many face recognition systems which, although they can provide high accuracy, require extremely stringent input conditions with respect to face orientation and illumination conditions. These systems also require very large training datasets and significant computing power in order to be able to provide reliable retrieval results within minutes rather than hours or days.

From the perspective of consumer applications, such systems are too sensitive and require far too much experience on the part of the operator. To date there are few practical systems which can be utilized to semi-automatically detect and categorize images based on the people in those images. The reason for this is the presence of different types of variations in the face region that affect the accuracy and robustness of the face recognition systems. These types of variations include: different illumination conditions, different face orientations, variations in face size, different picture quality caused by movement, poor focus and poor conditions for image acquisition. If we consider the goal of our system, to deal with consumer image collections, and then realize that most consumers are amateur photographers, it is very probable that all kinds of image variations will be present, indeed exaggerated, in the images that we propose to analyze.

In the field of security applications, systems that require face recognition modules for: biometric access control, secure transactions or terrorist surveillance, the recognition module has to provide very high accuracy and robustness against false acceptance (intruders that are accepted as trusted by the system). This requirement has a higher priority than all other requirements such as algorithm complexity, and speed, specialized image acquisition equipment or image acquisition conditions. The number of instances of false acceptance has to be minimized even if this increases the number of false rejections (trusted people classified as intruders by the system).

The requirements of our system are significantly different when working with collections of consumer images. Of course the accuracy is important but also the complexity and speed are important especially when dealing with large collections of images when the response to the user's input has to be fast otherwise the application will not be practical for the end user.

A second requirement is that the system has to be able to work with a wide variety of images regardless of size and quality and has to be able to cope with wide variations in acquisition conditions for the photos of a collection. No preconditions regarding image acquisition should be enforced other than the image being of reasonable quality and clearly viewable on a standard desktop computer at a resolution of 640x480 or larger (i.e. better quality cellphone images are acceptable, but not of less than VGA resolution). Further, a certain level of misclassification can be tolerated by the application as long as the overall classification works well and the user has some ability to override system classification when necessary.

Even if the accuracy is not the most important factor of the system, the large variations usually present in consumer image collection still make it difficult to sort the people in the images using only information extracted from the face region.

Inspired again by how the humans browse through a collection of images usually looking for pictures from a particular event or special occasion, or pictures taken in a small period of time (holidays), we argue that we can use also information extracted from other regions surrounding the face region which can enhance the accuracy and robustness of the sorting module. In fact our research indicates that this information is actually more robust to the variations that affect the accuracy of the face recognition module. Consider, for example, that men can often be distinguished by their ties which are frequently of a distinctive colour or pattern, or women by necklaces, or the neckline of their dress or blouse.

In order to keep the automatic property of the final system, the surrounding regions have also to be defined in an automatic manner using the coordinates of the face regions.

Two surrounding regions are defined in order to extract information useful for sorting: the body regions and the head regions. Both regions are processed differently from the face region because the information from these regions is different from the type of information which is useful in face recognition. The body region contains information regarding the clothes and special jewelry worn by a person during the same event and can be used for classification, while the head region contains information regarding the hair style of the person over a period of time which can be equally useful.

This method of combining multiple type of information for classifying the same pattern is called multimodality. Multimodal systems are regarded as a solution for many pattern recognition/classification systems, especially in biometrics, as it can cope with the fact that some types of information can be affected by different factors but there is at least some information left unaffected which can be used in the classification stage.

The first multimodal aspect of the proposed system is the use of additional regions for information extraction for person sorting. The second aspect is that more than one algorithm is used to extract information from the face region and it will be discussed in **Chapter 3**, dedicated to the face recognition module of the sorting system.

1.2 Goals of this Research

The goal of this thesis is to design, implement and test an adaptable *Person Recognizer* system which achieves the practical goal of analyzing and categorizing consumer image collections.

This will involve a series of sub-goals including:

(i) a study of state of art in face detection and face recognition techniques and implementation of working system modules;

(ii) a study of alternative image classification techniques and implementation of same;

(iii) the development of classifier combination techniques to improve the performance of the system when more than one face recognition or image classification technique is employed;

(iv) the integration of the above into a single system to enable comprehensive system testing;

(v) enhanced testing and refinement of the individual techniques, particularly over a range of facial poses and illumination conditions;

(vi) the study and development of techniques to compensate for extremes of (frontal) pose and illumination.

In addition we expect that this research will lead to novel refinements of certain aspects of the facial recognition and image classification techniques we employ.

1.3 Literature Review

Although there is plenty of prior research dedicated to face detection, person recognition, and classification using the face region, there is not much prior work regarding the specific topic of sorting consumer digital image collections. In this section, we will only cover specific review relating to this main topic, which is central to the theme of this thesis. Additional literature review on face detection, face recognition and other pattern recognition and image classification techniques will be given at the start of the relevant chapters of this work.

In [119] Zhang et al. describe a system for face annotation (associating names with face regions) in family photo albums. It uses similarities between people in order to show to the user a list of possible names in a sorted order of their similarity. The user associates a name with a given face just by clicking the correct name in the list.

In order to compute the ranked list of possible names it uses an image database prior labeled with names and computes the similarities of the given face with all faces in the database.

Their system also uses extra features along with the features extracted from the face region. These extra features are extracted from a region that includes the entire face and body regions, and some surrounding parts belonging to the background. The features extracted from each region are combined, in order to compute similarities between people, using a Bayesian framework formula. Features extracted from the face region are used in the final classification only if the detected face is reported as frontal, otherwise the face region features are treated as missing.

The similarities between our work and the research described in this paper lie in the purpose of the application which is to organize consumer collection of images and also in the idea of using extra information along with the information extracted from the face region to differentiate between people.

The differences start with the implementation of the person recognizer module: they are using person similarity to assign a name to each face in the images so it can be integrated with a keyword searching engine. For that the user has to manually assign names for every face. Our proposed system performs a blind training over the images without user intervention, and when the user is searching for a person he only has to show an example to the system. The system will retrieve images with people similar to the one given as an example. The way that the regions are defined is completely different: they are analyzing an extra large region of the image for additional information, a region that contains the face region along with a large part of the surrounding environment which can have a negative influence on the classification. Our proposed system uses different distinct regions for analysis, regions that have high probability of being present in the image without obstacles.

The method of combining the information is completely different between the systems. They are combining the features extracted from different regions in order to compute the similarity between persons. Our system computes similarities between all the regions and combines these similarities in order to determine a global similarity score. This score is used to sort each image relative to all of the other images in the collection. Every region is processed so the important information representative to the region is extracted for classification.

In [43] Girgensohn et al. present an image-organizing tool, which can group images according to different combinations of factors, such as events, places and people. In the case of their people grouping function, a classical face recognition module is used in order to find images with people that are similar in order to group them in the same folder. Giving a pre-trained pre-labeled collection of images the system will try to group in the same folder new images that are imported in the collection based on face similarity with faces in the collection. The user supervises this grouping procedure and he can overwrite the system decisions. In the end each person in the collection will have a separate folder with images where the person is present.

One common point with the proposed system is the goal of the system which is to organize images in a collection using the people in the image. However, the processing used for people classification is different from our system. These authors employ classical face recognition in order to cluster similar people without taking into consideration any other information. As stated before, due to the presence of large variations usually encountered in consumer collections of images, it is really difficult to implement a face recognition system with a high accuracy under these circumstances. This will become apparent from some of the experiments detailed later in this thesis.

Other interesting recent image managing systems that use face recognition algorithms to annotate/sort images are reported by companies like Fujifilm in [1] and Myheritage.com in [2]. We couldn't obtain technical details about their systems but they are using also classical face recognition algorithms.

An interesting system very recent was reported in [3] by Riya. They are using face recognition combined with other features including body/clothes features in order to compute similarities between people. No other details could be obtained.

For more a comprehensive literature review of each component of the *Person Recognizer* please see the review sections of the **second, third and forth Chapters**.

1.4 Structure of the Thesis

The thesis is structured in ten chapters starting with the introduction and concluded with the final chapter dedicated to the conclusions and future directions of research. Additionally there is one appendix at the end dedicated to one interesting aspect discovered during this thesis.

Chapter 1 has provided introductory and background information on the field of research and the principal goals of this thesis.

Chapter 2 provides an overview of Face Detection techniques and explains why certain techniques were selected and are employed as the first component of the sorting system. It starts by explaining the role of this module and elaborates on the detailed requirements that have to be met by the detection algorithm in the final system. It also details the problems that have to be addressed in order to build a robust face detection algorithm and a literature review of reported methods is given, together with a summary of their characteristics, advantages and disadvantages. A detailed description of the face detection algorithm that was eventually selected for use in the final sorting system is given and some practical aspects of its implementation are discussed. Finally, some potential improvement for the algorithm is presented. These suggestions could make the algorithm faster and eliminate some of the false regions that are occasionally reported as faces.

Chapter 3 covers Face Recognition techniques. In particular it describes the core module of the system that analyzes the detected face regions in an image, computes feature vector patterns from them and subsequently determines similarities between the extracted patterns and the patterns of other face regions in the same image collection. It starts with a description of the main requirements that must be satisfied in order for the recognition algorithm to be used in the final system. An overview of the principle difficulties that have to be overcome by the module follows. These difficulties arise because our Person Recognizer must be able to work with consumer collection of images and to function automatically without intervention from the user.

A number of categories of recognition systems are discussed and reviewed in the next section. Because we are working with consumer image collections many of the recognition systems reported in the literature cannot be employed in our application. These systems are described briefly and their use is discounted. We then focus on several of the remaining techniques which have good performance and are suitable. The chapter continues with a detailed description of the mathematical algorithms that were used for computing face similarities, namely (i) discrete cosine transform (DCT), (ii) principal component analysis (PCA) and wavelet decomposition along with (iii) preprocessing techniques used in the sorting system. The advantages and disadvantages of each of the described methods are presented.

In **Chapter 4**, the analysis of peripheral face regions is described. These regions are included because our research indicates they can significantly improve the accuracy and contextual relevance of the final retrieval system. This chapter begins with an overview of the reason for using such additional information and what advantages its use will bring. It continues with a description of how these additional regions are defined, and provides a detailed description of several of the algorithms that can be used to extract such information. We focus on histogram and colour correlogram based feature extraction. Several algorithms, including the colour autocorrelogram and the banded colour correlogram have been tested and are described here. This chapter also describes in detail a fast algorithm for computing the colour correlogram features.

Chapter 5 is focused on the combination of multiple classification techniques to provide a single measure of similarity between detected face regions. This chapter is dedicated to the multimodal aspects of our proposed Person Recognizer system. It begins with an overview of generic multimodal systems and explains how the multimodality of a system can increase the accuracy of recognition/classification rates. Examples review of multimodal systems reported in the literature are given in the next section, continuing with a detailed presentation of the multimodality of the proposed system. The final part of this chapter is dedicated to the algorithm that was used in our system in order to combine similarities from different classification methods.

In **Chapter 6** we present initial results obtained using the retrieval system. This chapter begins by describing the database of images that was used for initial testing and the combinations of parameters used in order to achieve optimal configuration of the retrieval system are given. It continues with details of further tests and fine tuning required to obtain a final configuration of the multimodal face recognition module. Details are also given of various tests that were performed in order to analyze the performance of the face recognition module alone, using four standard face recognition databases.

Improvements to the face recognition module are presented in **Chapter 7**. These improvements regard the normalization of the two most important image variations that affect the robustness of face recognition: illumination and face orientation. For illumination normalization we have obtained very useful results using the CLAHE method, which is described in some detail. Results are presented from tests on two face recognition databases which comprise facial images at variable levels of illumination. For pose normalization we describe the Active Appearance Models (AAM) statistical face modeling technique and present some initial results achieved on a standard face recognition database which comprises facial images in a variety of poses.

In **Chapter 8** we describe several examples of the final "retrieval applications"which were developed. The first of these was implemented in order to demonstrate a working proof-of-concept to our industrial partner. Subsequent applications were implemented in order to gain an understanding of how these techniques might be employed as client-server network applications, or as a network service for mobile phone applications. The resulting workflows and use cases help place much of the core research work of this thesis into context.

This chapter begins with a general description of the architecture of the retrieval application and shows the functionality and workflow of the principle system components.

The first application described is a desktop application designed to be used by any user on his desktop computer. It is easy and intuitive to use, the training stage is hidden from the user in the background and is executed every time new images are added to the collection. The search is performed by providing a query example.

The second application is a web based application dedicated to those users who store their pictures on remote dedicated servers. The servers are in charge of training and searching trough the image collection, and searching is performed by the user using a standard web browsing interface.

Chapter 9, "Combining PCA collections" discuss the problem of updating the PCA collection of data when the format of the collection changes. It is known that PCA is a data dependent method, meaning that every time new data is added to the collection the analysis algorithm has to be re-applied to update the coefficients. In this section we present and analyze an incremental method of updating the PCA data which does not require a full re-training of basis functions for an image collection when new images are added, or when two or more collections of images are merged into a single super-collection.

Chapter 10 concludes the thesis by drawing important conclusions from the work performed and outlining possible future research directions that will improve our Person Recognition system and associated tools for the management and organization of consumer digital image collections.

The appendix, "Face recognition - will extra knowledge about the training data always help?" analyzes a possible explanation of why some face recognition techniques perform badly when they incorporate extra information regarding the cluster distribution in the training data. These problems appeared when the Fisher face method gave worse results than DCT and PCA approaches.

Finally we present a comprehensive bibliography which includes all of the references and prior art that were reviewed in the course of this study and that provided the technical and scientific basis on which this thesis has been built.

1.5 System Overview

The goal of our Person Recognizer is to sort and organize consumer collections of digital images in an easy, fast, and intuitive way using the people in the images. Like most retrieval systems, it has two main modules that share several common components. The first system component, known as the training subsystem, analyzes all images in a collection extracting the information that will be used by the second module, known as the retrieval or sorting subsystem, in order to find images of interest to the end user. In order to extract the necessary information during training, the first component of the training subsystem has to be an automatic face detection module. The second component of the system is a pattern extraction module which implements several distinct pattern matching/extraction algorithms; the extracted pattern features are stored in a dataset associated with the collection of images.

Note that each image is analyzed and categorized in the context of an associated image collection, but naturally it can be placed into a separate, distinct collection and analyzed independently in that alternative context. This gives rise to the idea of sub-collections and super-collections of images and the possibility to analyze images and categorize images within different contexts, as each collection will have a different mean pattern and a different distribution of pattern features. Furthermore, in the case of certain feature patterns, e.g. PCA components, each collection will have a distinct set of basis functions and thus the same image may have a very different pattern set associated with it for each distinct image collection context within which it is analyzed and categorized. These ideas will be discussed in greater detail later in this thesis, and we will consider how distinct image collections can be merged to achieve a combined mean pattern set and allow images from either collection to be categorized in the context of the combined super-collection without a need for retraining the combined super-collection. This is quite a powerful concept and can greatly simplify the use of a categorization tool from the perspective of an end user. We will return to this topic in **Chapter 3** and **Chapter 8**.

The retrieval module takes an example input from the user, typically a face region (selection of multiple face regions is also supported), detected by the same software module used for training the collection. The same pattern extraction algorithms are applied as those used by the training subsystem, and based on the extracted pattern features one or more classification sorting components are called in order to arrange the images in the collection according to their similarity with respect to the query example. A more detailed description of the system is given in **Chapter 8** prior to describing some actual applications implemented using the core system software.

The requirements for the training module are:

(i) the speed of the training stage has to be fast in order to be viable for practical use,

(ii) training should be fully automated and ideally should be hidden completely from the user,

(iii) user input should be minimal and ideally non-existent so that any consumer can use the system without technical knowledge,

(iv) the size of the extra information/metadata that must be stored after training must to be kept to a minimum so it will not affect the storage capabilities of the user's system to a noticeable extent,

(v) the system must be able to process any type of images that the user may collect.

For the sorting stage, the most important requirements are related to the speed of retrieval, which should naturally be relatively fast, and of course the perceived accuracy of retrieval. Searching in a moderately large collection (typically several hundred images) has to be fast in order to show the results to the user within an acceptable timeframe of a few seconds. Even when dealing with large collections of images (typically thousands of images) the retrieval lag should not be more than 10-15 seconds or users will feel the system is not practically usable.

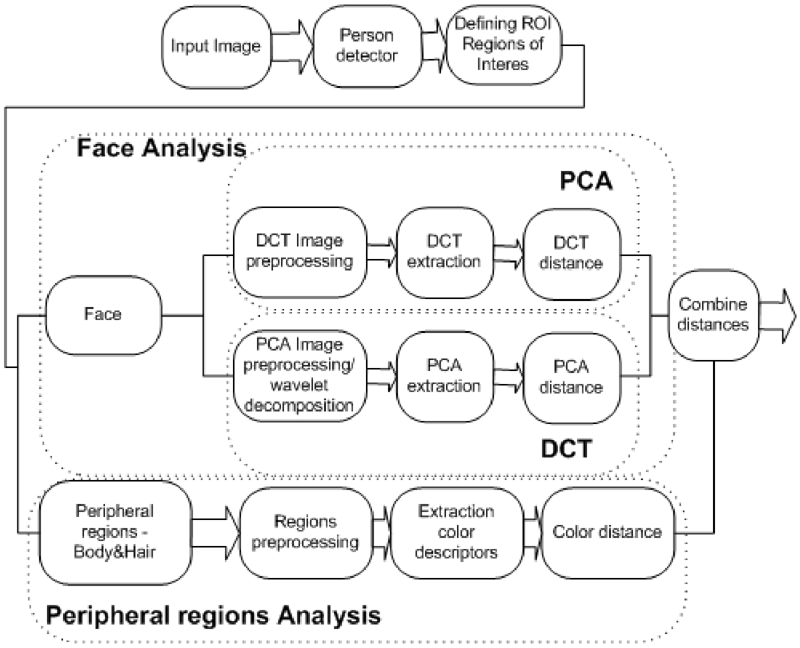


Figure 1.1: System overview

Figure 1.1 represents the general architecture of the system. Each system component is implemented and tested independently in different chapters in the thesis. The algorithms in some of the modules are chosen based on extensive review of what is available in the research world and extensive testing on different data collections (face detection, extracting information from additional regions). All refinements and improvements to these algorithms are backed up by more testing in order to validate them. Algorithms used in other modules were developed during this thesis to better suit our goals and were also tested on multiple data collections for validation (face recognition, combining scores from multiple algorithms)

For each image the system initially detects all face regions. After detection we define regions of interest (ROI) from the relative position returned by the automated face detector described in **Chapter 2**.

The face regions are processed differently from the peripheral regions. The preprocessing techniques and the feature extraction algorithms applied to the face regions are described in detail in **Chapter 3** and the algorithms applied to the peripheral regions are described in **Chapter 4**.

In the training stage the computed features are stored in the collection database. In test mode the same processing techniques are applied to the input image and the computed features are compared with the features stored in the collection database. A statistical approach, detailed in **Chapter 5**, is used to combine the scores from each region in order to obtain a final similarity score. This score will be used by the retrieval module to return to the user the images which are most similar to the given example.

Chapter

2 Face Detection

This chapter is concerned with the first major system component of any fully automatic face recognition system - the face detection subsystem.

Section 2.1 provides an historical overview of face detection, starting with a general definition of the process and continuing in section 2.1.2 with a description of some of the main approaches reported in the literature and the current state-of-art in this domain. Section 2.1.3 describes different issues associated with the detection process, which makes the face detection domain still a very challenging field of research. Evaluation techniques and the definitions of false positives, false negatives, detection rates and speeds are introduced in section 2.1.4 which is concerned with defining the metrics to be used in determining how accurate the detection method is.

In section 2.2 a detailed description of the detection algorithm used in the final system is given. This begins with a description of the features used for detection in section 2.2.1, continues with the presentation of a fast algorithm used to compute the features in section 2.2.2 and a description of how an optimized set of classifiers is determined from the very large set of all possible classifiers is given in section 2.2.3. Section 2.2.4 explains how optimal classifiers are organized into a cascade and implementation details and various tweaks to the basic algorithm, together with some detection results are presented in section 2.2.5.

Finally the chapter ends with section 2.2.6 where we describe several improvements to the detection algorithm tested during this thesis.

2.1 Introduction

Face detection is a vital component in almost any human-computer interaction system. Although there are many reported methods for implementing such systems which assume that the presence and location of the face is apriori known, it is essential for building a fully automated system to include a face detector module prior to recognition process.

Applications such as face recognition, face tracking, pose estimation, and expression recognition are based on analyzing as input only the region from the image containing a face. More complex applications that require information about the location of the face in a captured image include supervised driving where the state of the driver is analyzed using captured images while driving [4] and auto-focus in digital cameras where the focus control of the camera is automatically adjusted using the position of the face in images [5].

2.1.1 Definition

One definition of face detection is:

> *Given a still image, the goal is to (i) decide if there are faces present in the image and if so (ii) to return their one, or more, locations.*

Technically, face detection involves having a series of raw data samples corresponding to the input image or sequence of images, analyzing these data and returning those groups of samples that belong to the face regions in the image or sequence of images.

A very general representation of the face detection process is given in Figure 2.1.

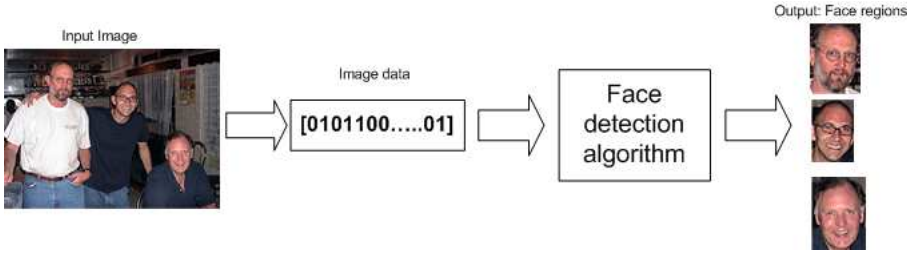


Figure 2.1: Face detection process

Ideally, the face detector should return all the faces present in the image without any false region to be reported as face. Such a face detector should give perfect results independently of people's characteristics such as sex, colour, age, appearance and independently of image characteristics such as: illumination, focus or distance to subject.

In practice, the only ideal face detector, which can be relied on to give perfect results, is the human method of detecting faces.

It can be noticed from Figure 2.2 that even in a single image multiple types of variations can be present in the face regions. These variations can make it very difficult to implement an automated face detector close to the ideal requirements. This problem will be discussed in some detail in Section 2.1.3

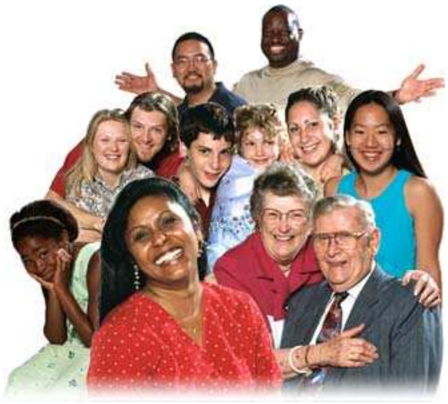


Figure 2.2: Face variations

2.1.2 Face Detection Literature Review

It is not clear how we humans detect faces in an image. In the literature there is no clear indication of which are the predominant factors that validate a region as a face. Various researchers suggest it can be the shape of the face, its colour (the human skin colour), or the features that are always present in human face regions such as the eyes, mouth or nose. Most likely it is a combination of all these factors that makes us detect faces without problems.

Inspired by these categories of facial characteristics researchers have tried to emulate the face detection process by integrating algorithms that search in an image for regions that satisfy some of the above conditions. Other approaches employ ”machine learning” techniques where large databases of images are fed to algorithms that can ”learn” the patterns associated with the selected training data. A more detailed review of some of these prior art will be given shortly.

From an engineering point of view we remark that it is easier to implement an algorithm that verifies if a given image region is a face than to search for a face in a full image.

In order to detect all faces in a given image using such a face verifier, we have to split the full image into possible candidate regions and to pass them through the face verifier for validation. A representation of this idea is given in Figure 2.3 and represents another common approach to face detection.

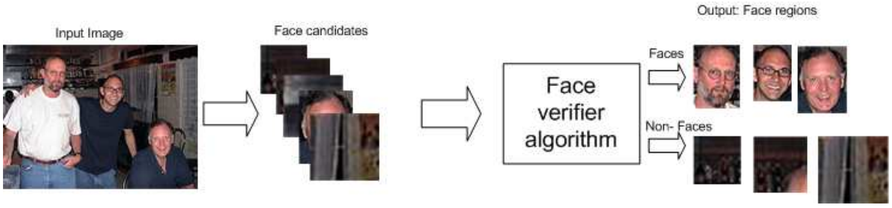


Figure 2.3: Face detection using face verifier

Usually the splitting of an image consists of using an overlap-sliding window of different sizes for considering all potential candidate regions. The window characteristics (the size and overlapping step) influence the overall detection system regarding the detection time and also characteristics like minimum and maximum sizes of the detected faces. The other characteristics of the detection system are determined by the face verifier module.

A simple way to verify if an image is a face is to use apriori information from human knowledge like basic rules satisfied by different parts of the face. These types of systems are classified as **Knowledge-based** methods. There are many possibilities for defining these rules; the simplest is using the symmetry of the eyes along with the position of the nose and mouth. The most representative technique of such knowledge-based approaches is that developed by Yang and Huang in [116], where three levels of rules are applied in order to validate a face region. The rules at a higher level are general descriptions of face geometry while the rules at lower levels rely on details of facial features. Kotropoulos and Pitas in [61] analyzed the vertical and horizontal distributions of the intensity in the image and applied rules regarding the intensity distribution of different facial feature (such as hair, eyebrows, mouth, nose) in order to detect possible face regions. These rules are based on the fact that due to the different colour distributions of facial feature, intensity transitions have to be reported when passing from facial features to facial skin regions. Detection of other facial features is further used to validate the candidate regions. The problem with this approach is that if the defined rules are too strict the system will fail to detect face regions and if the rules are too general, many false positives will be generated. It is very difficult to define generic rules which can incorporate information from all possible face orientations.

Another category of approaches are represented by Feature invariant methods. Numerous methods have been proposed to first detect different facial features and then validate face regions. Facial features like eyebrows, eyes, mouth, nose and hair line are commonly extracted using edge detectors and based on these features a statistical model can be built to describe the relationships between them. In the end the model can be used to verify the presence of a face.

One of the most common approaches from this category is to detect in the image regions with human skin colour. The simplest way to define a skin model is to define a space region in the colour space where the skin tone lies based on a statistical analysis of marked skin pixels. Many methods for detecting human skin have been proposed using different colour spaces: RGB [56], normalized RGB [71], HSV [97] , YCrCb [53], YIQ [34], CIE LUV [117].

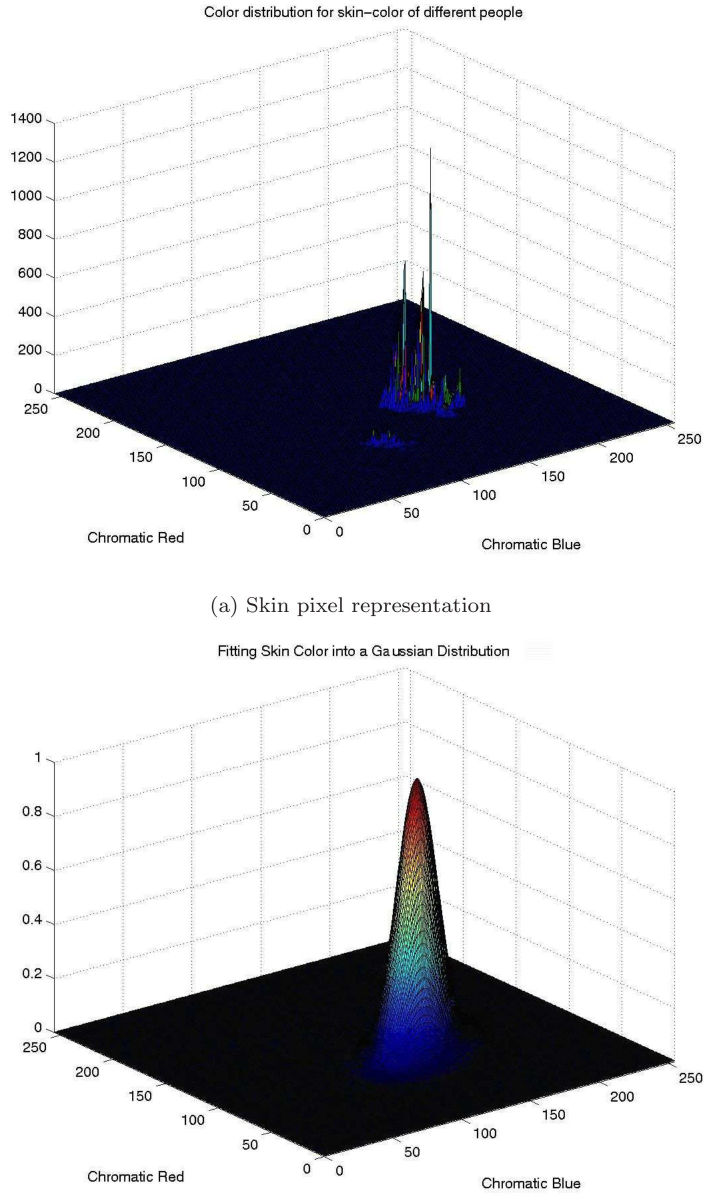
Figure 2.4 represents a histogram for skin pixels in CrCb space and an example of fitting a Gaussian in the representation to form a skin model [6].

Even if the skin detector is perfect, additional rules have to be applied in order to validate a region as a face, otherwise other human body components are reported as faces. But any skin detector has disadvantages. If the skin colour model is defined too loosely than many objects close to skin colour will be reported as faces. If the skin model is too restrictive then due to the variations in illumination some or all of the face skin could easily be projected outside the defined skin model. This will cause the detector to skip face regions.

Another feature invariant approach is based on detecting regions with texture close to the face texture. Second order statistical features used by Haralick et al. in [48] can be used to characterize such texture for face detection. Again it is very difficult to define a texture model which defines only face regions. Other objects can have texture close to human faces.

Facial features can be used for face validation but in order to use this method first we have to detect the features. In [20], Canny used his detector to build an edge map in order to separate the face from the background and an ellipse was fitted to the boundary between the head and the background, and in [44], Graf et al. used band pass filters followed by morphological operations applied to greyscale images in order to detect certain shapes like eyes for detecting the face region. One problem with such approaches is that it is not trivial to detect these facial features especially when they are corrupted with noise, occluded or viewed with different illuminations. As the facial feature detection directly influences the face detection, any undetected feature will result in an undetected face. Usually these approaches are used as post processing techniques for other face detection algorithm in order to validate the face regions.

A wide range of face detection algorithms are based on **Template matching**. This type of methods involves defining a standard face pattern, then con-



(a) Skin pixel representation

(b) skin model representation using Gaussian Distribution

Figure 2.4: Skin representation in CrCb space

volution values between this pattern and different regions from the image are computed. Based on these values the presence of the face is determined. Multiresolution and multiscale templates, subtemplates, and deformable templates have subsequently been proposed to cope with variations in scale and shape. We can use multiple predefined templates for different features like: eyes, nose, mouth or face shape taken separately for detecting candidates. The templates can also be used to validate candidate regions.

Deformable templates can be used as well. This involves defining an elastic model for the face controlled by parameters and an energy function which has to be minimized varying these parameters in order to obtain the best fit model for the detected face. Active shape models ASM developed by Cootes and Taylor in [26] incorporate shape and intensity information in order to build a face model from a given database of faces. The model will fit into a new face by minimizing the error of the reconstructed face.

The **Appearance based** methods represent the biggest category of face detection algorithms. Compared with the template matching approach where the templates where predefined by humans, in this approach the templates are learned from examples using techniques like statistical analysis or machine learning to find relevant characteristics in face and non face examples. One of the most common face detection algorithms is based on Principal Component Analysis (or Karhunen-Loeve transform or eigenface method)and will be described in detail in **Chapter 3**. In [109] Turk and Pentland used PCA for both face detection and face recognition. PCA for face detection is based on computing a set of eigenvectors (eigenfaces) from a set of face/non-face examples and then project each candidate region onto the space defined by these eigenvectors. This projection will give a measure of how close is that candidate region to being a face/non-face.

Support Vector Machines (SVM) were used for face detection in [77] by Osuna et al. SVM is a binary classification method which tries to find an optimal hyperplane to separate two patterns by minimizing the expected classification error. By training the SVM with faces and non faces examples we can use it as a verifier for unknown regions.

A related approach uses neural network techniques for face detection. By training any neural network with faces/non faces example we can use it for verifying face regions. The most representative neural network based face detection system is the one presented in [94] by Rowley et al. Many other methods of face detection using neural networks have been proposed, example: hierarchical neural networks [13], multilayer neural networks [47], Kohonen self organizing maps [19], associative neural networks [36], probabilistic decision neural networks [68].

In [96] Samaria used Hidden Markov Models (HMMs) for face detection and recognition. The support of using HMMs for detection is that the facial appearance should have the same succession when scanning the face region the same way. For example if scanning from top to bottom the succession is: hair, forehead, eyes, nose, mouth and chin. By modeling this succession as a statistical process we can use HMM to determine how close a given region is to a face.

The accuracy of the detection when using the appearance based methods is dependent on the size of the collection of faces/non-faces examples. This collection has to be big enough in order to capture all possible variations inside the face region.

The main problem that has to be addressed for most of these algorithms is the time required to detect all the faces in a given image. When using a face verifier approach to validate all possible candidates in the image the number of candidates can be extremely high. Typically a face verifier will need to scan the entire image across a wide range of scales - for a large size of image 10 or more scales may be required. For the smallest scales the computational costs can be extremely high. Consider, for example, where each scale is 0.8 times smaller than the preceding size of candidate region. The 8th scale will be 0.17 times the length of the 1st scale, or less than 3% of the area. Thus more than 30 times the amount of computation will be required to cover the image at this scale. For the 9th scale the area will be less than 2%, requiring 50 times the amount of computation, and so on. (Note that face classifiers are typically of a fixed pixel size, e.g. 24x24, or 32x32, and candidate regions must be re-sized to the same size as the classifiers).

Even if the validation stage is very fast, the processing of all candidates could take very long if each candidate is considered separately. The face detection used in this thesis has the advantage that even though it uses the same verification stages, it computes the features for all candidates extremely quick through a single recursion of the image to create what is known as an integral image. That, when combined with the fact that it uses simple and fast classifiers makes it very fast to detect faces even for large images.

2.1.3 Challenges for Automatic Face Detection

The principle challenges associated with automated face detection techniques are determined by the following factors:

(1) Face orientation (pose). The appearance of the face may differ in many ways when the orientation of the face changes from frontal to profile or extreme

view angles, where face components like the eyes, the nose or the ear may be occluded.

(2) Changes in face appearance. From person to person different features in the face appearance may be present or absent. Examples include beards, moustaches or glasses. Women may use make up which can significantly alter the face colour and texture. These factors together with the potential for variability in shape, size or colour of the face makes face detection a very challenging task for researchers.

(3) Facial expression. The appearance of the face is directly affected by the person's facial expression. Some people are more affected than others.

(4) Occlusions. Different components from the face may be occluded in the image by other objects and in group photos some faces may be occluded by other faces.

(5) Capture conditions. Factors that are involved in capturing the image like: illumination conditions, camera characteristics or quality of the captured image have a huge influence in the detection process.

All these factors have to be addressed when building a face detector, otherwise the system will not be useful in real world applications where it is very likely that such variations will frequently occur.

It is possible to make the system capable of detecting faces over large variations of different factors by loosening certain constraints regarding variations inside the regions. But this can cause it to detect regions that are not part of faces. A compromise must therefore be reached between the variations that the system can cope with, which influence the detection rate of the system, and the number of false regions reported as faces by the detection system.

2.1.4 Metrics Used to Evaluate Face Detection Algorithms

The evaluation measures that are used when comparing different face detection systems are:

- **Detection rate**: which represents usually how many faces were detected by the system compared with how many a human can notice. An image region identified as a face by a classifier is considered to be correctly detected if the image region covers more than a certain percentage of that face in the image.

- **False positives rate**: which represents how many non-face regions were classified as face regions by the system.

- **Detection speed**: represents how much time does the system require to detect the faces in an image of a certain size. Even if from a research point of view this measure it's not as important as the other two, for a practical implementation it is very important how fast and how complex is the algorithm.

The performance of face detection algorithms can be evaluated by plotting the receiver-operating characteristic (ROC) curves. The ROC graphs plot the detection rate against the false positives rate.

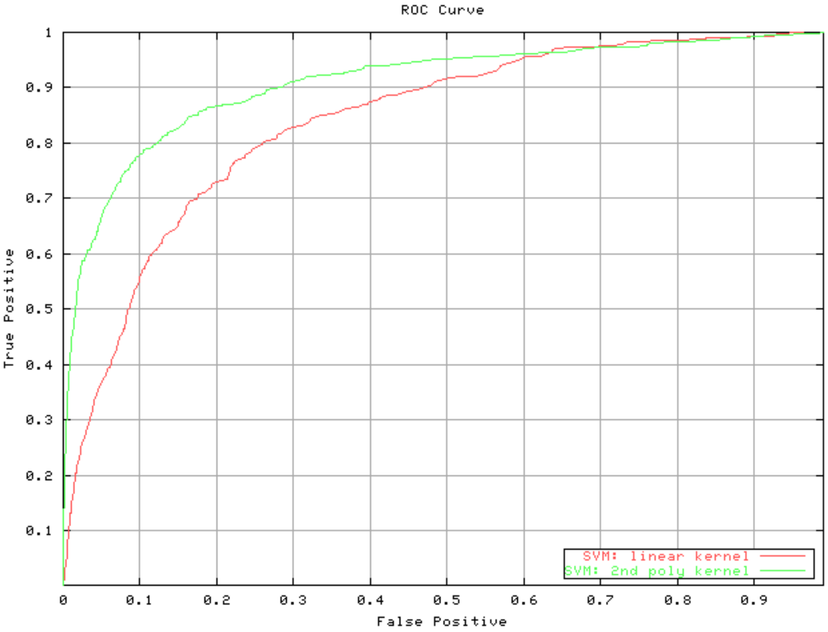


Figure 2.5: Typical ROC curve

The main requirements for the face detector module from our application point of view are that, firstly the detection has to be fast because we want to work with large collections of images with a high number of people present and secondly we want the detection rate to be as high as is practical even if that means significant increases in the number of false positives returned. If we have an accurate face similarity measure implemented in the classification stage, these false positives should not unduly influence the retrieval results and thus we can tolerate higher false positive rates.

2.2 Description of Face Detection Method Used in the System

As we discussed in the previous section, most face detection algorithms have the disadvantage that they require a significant amount of computation in order to process all potential face candidates in an image. In turn such algorithms are quite slow unless they can be run on specialized parallel-processing hardware systems.

The face detection method we chose [113] takes advantage of using simple features for classification. These features can be computed very quickly using recursion and an intermediary image known as the integral image. Also by using a cascade of classifiers from simple to complex, each candidate is processed in as short a time as possible, making the overall detection very fast. By using a large collection of training images with face and non-face examples the detection algorithm reaches a very high detection rate which is very important for the final classification system. The face detection method is described in detail by Viola and Jones [113] and is part of the last category of face detector - appearance based detectors.

This detector is the fastest face detector reported in the literature, an essential requirement for our system and the reported results are very promising. The algorithm has been well proved in recent years as being the fastest face detection algorithm reported and is presently the gold standard against which other face detection techniques are benchmarked. Most of the improvements to this algorithm relate to methods of improving the time taken to train classifiers [92], or the efficiency of the classifiers themselves [69]. For the purposes of this thesis the original algorithm is more than adequate both in terms of overall accuracy and speed.

The algorithm is based on using a set of features called Haar-like features with a cascade of classifiers from simple to complex. Each classifier will discard candidate face regions and the candidates that remain at the end of the cascade are reported as face regions.

Two main ideas in the algorithm contribute to making it very fast and reliable. First, the use of a new representation of the image called integral image facilitates a fast computation of the features in the image, and second, in order to train the cascade of classifiers a variant of the *AdaBoost* [39] algorithm is used, which selects only a small number of important features inside the candidate regions. Also, using the cascade of classifiers from simple to complex, which means that the first classifiers are very simple and fast, will eliminate most of the candidate regions in the early steps of the cascade and thus contribute to an increase in the speed of the algorithm.

A detailed description of the algorithm is given in [113] by Viola & Jones. (Hereafter the algorithm will be refered to as the VJ algorithm). Additional refinements were added by Lienhart in [92]. For the convenience of the reader an overview of the main elements of the algorithm is given below.

2.2.1 Features

The algorithm uses features related with the Haar basis functions which have been introduced in [103]. Three types of features can be defined, as shown in Figure 2.6:

- two-rectangle feature value computed as the difference between the sums of the pixels within two rectangular regions. The regions have the same size and shape and are horizontally or vertically adjacent.
- three-rectangle feature computes the sum within two outside rectangles subtracted from the sum in a central rectangle.
- four-rectangle feature computes the difference between diagonal pairs of rectangles.

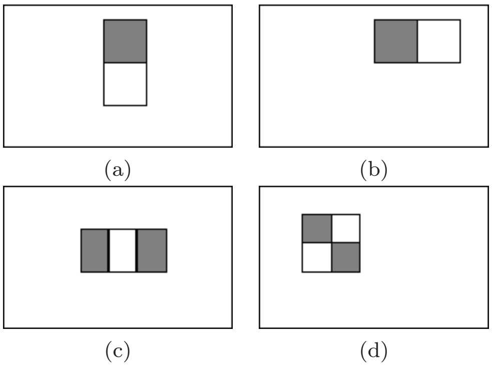


Figure 2.6: Feature examples. The sum of the pixels which lie within the white rectangles are subtracted from the sum of pixels in the grey rectangles. a) and b) two-rectangle features, c)three-rectangle feature d)four-rectangle feature.

2.2.2 Integral Image

A naive algorithm for computing all features inside a candidate region will be very inefficient due to the large number of possible features (for a 24x24 pixel detector the number of rectangle features is 45,396).

The notion of *integral image* is introduced, representing for a certain pixel the sum of all pixels above and to the left of the given pixel. The dimension of the integral image is the same as the original image and it will be used for computing the sum of all pixels inside any given rectangle as it is shown in Figure 2.7.

The formal definition of the integral image for an image Q is given in eq. 2.1

$$I_Q(i,j) = \sum_{i' \leq i, j' \leq j} Q(i^{'}, j^{'}) \tag{2.1}$$

where $I_Q(i,j)$ is the value of the integral image for pixel (i,j) and $Q(i,j)$ is the value of the original image for pixel (i,j).

Using the recurrences defined in eq 2.2 and eq. 2.3 the integral image can be computed in one pass over the original image:

$$S(i,j) = S(i,j-1) + I_Q(i,j) \tag{2.2}$$

$$I_Q(i,j) = I_Q(i-1,j) + S(i,j) \tag{2.3}$$

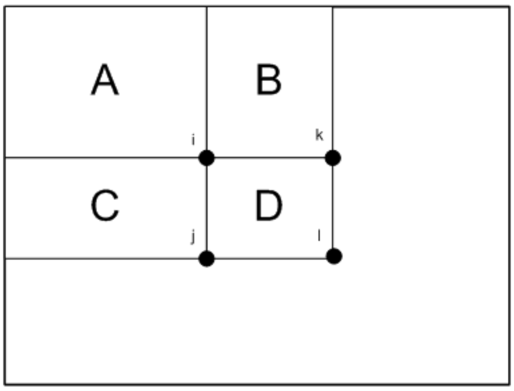


Figure 2.7: Using the integral image to compute the sum of all pixels inside rectangle D

Using the representation in Figure 2.7, if we want to compute the sum of all pixels in rectangle D we can use the pixels in rectangles A, B and C and the integral image values in the right-bottom corner as follows:

$$D = I(l) - (A + B + C)$$
$$\text{but } A = I(i),\ B = I(k) - I(i) \text{ and} C = I(j) - I(i)$$
$$\text{which makes } D = I(l) - (I(k) + I(j) - I(i))$$

This result means that the sum of pixels in any rectangle in the image can be computed using four array references of the integral image. Also the difference between two rectangular sums can be computed using eight references. Since the two rectangle features defined above involve adjacent rectangular sums they can be computed in six array references, eight in the case of the three-rectangle features, and nine for four-rectangle features. As all these operations are simple additions and subtractions there are significant computational benefits to the algorithm.

In order to detect faces of different sizes, a standard size for the detection window is defined, the image is scaled at different sizes and the detector is applied on each candidate using an overlapping sliding window of standardized size. The features and classification functions were obtained from [7]

2.2.3 Classification Functions

In the previous section we concluded that the features from each candidate region can be computed very efficiently. If we assume that the candidate regions which represent sub-windows of the image have the dimension 24x24 pixels, than the number of rectangular features inside a candidate region is over 45,000. Even if they are computed very efficiently, classifying them would require a huge amount of time.

A key innovation in the VJ algorithm is to use in the classification stage only those Haar features which contribute to a positive classification and ignore all others.

A boosting technique is used to select features and train the classifier cascade. A cascade of classifiers represents multiple single classifiers which are serialized. This boosting technique is a variant of the *AdaBoost* method which is based on forming a strong classifier by combining a collection of weak classifiers.

The *AdaBoost* was originally used in [39] to select from a large number of classifiers those which exhibit good classification rates. Weights are assigned to classifiers according to their overall classification rate across a large training set, and the weaker classifiers are subsequently discarded. In this approach the *AdaBoost* was used to select the features which best classify face patterns. A detailed description of how *AdaBoost* was used is given in [113] and [92].

In our case the features relate to each possible rectangle in the image and are represented by numbers. For each feature a classifier can be formed by comparing the number with a threshold. The decision of the classifier can be 0 or 1.

If we restrict each classifier to use only one feature we can select the single rectangle which best separates the positives and negatives example. For each feature, the learning function determines the optimal threshold classification function, so the number of examples that are misclassified will be kept to a minimum.

This type of classifier which uses only one single feature represents the weak classifiers because we cannot expect to obtain a good accuracy using a single feature (rectangle) from the image. By combining multiple such classifiers we can obtain a strong classifier.

It is impossible to obtain a good accuracy using a single feature. The best features that are chosen first in the process have smaller error rates (0.1-0.3) and the features chosen later have higher rates (0.4-0.5).

In Figure 2.8 are represented the first and the second features selected by *AdaBoost* as having the best classification rate.

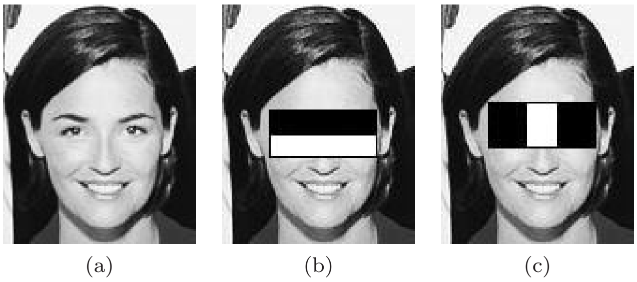


Figure 2.8: The first (b) and second (c) features selected by *AdaBoost* overlayed a face image

2.2.4 Cascade of Classifiers

If we use a single classifier, we can control the error rate by adjusting the threshold of the *AdaBoost* algorithm. A lower threshold means a very good detection rate but a higher number of false positives, if we try to minimize the number of false positives the detection rate will drop as well. Its been proven in [113] that in order to have a 100% detection rate the threshold has to be fixed so that it will also report 40% false positives which is unacceptable in any application.

The idea of using a cascade of classifiers is that each classifier will have a different threshold and will eliminate different false positives while keeping the detection rate to a maximum.

Also the first classifiers should be as simple and efficient as is practical because they will process the highest number of candidates and ideally they will reject a very large number of false candidates. The ones that pass the first classifiers will be processed by the succeeding classifiers, which are typically of increasing complexity, thus requiring more computational steps to implement.

In this way the cascade of classifiers is arranged according to classifier complexity, which in turn is optimal from the point of view of the algorithm's speed.

The candidate regions that pass through all classifiers in the cascade will be reported as being face regions at the end of the detection stage. Additional con-

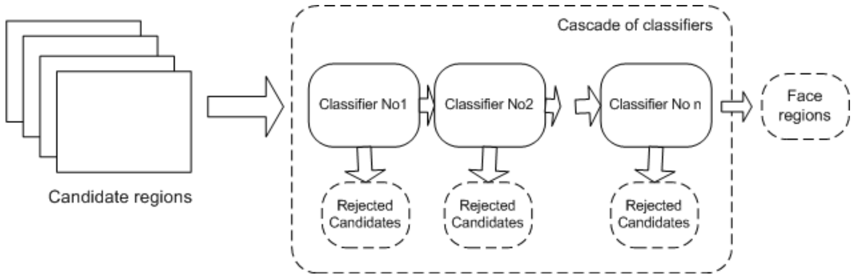


Figure 2.9: Face detection process

straints may be applied to the final candidates in order to detect only the correct face regions. For example, a simple proximity search for adjacent candidates, or co-located candidates at different scales.

2.2.5 Implementation and Preliminary Tests of the Face Detection Algorithm

The detection algorithm is implemented as part of the Intel digital image processing C++ library OpenCV [7]. This detector has been pre-trained using an impressive database of face/non-face examples.

Interfaces with Matlab and Java programs were created during this thesis in order to employ this OpenCV library for a variety of different tests.

In order to verify the accuracy of the algorithm we used multiple collections of images. Some collections had more than 1000 images and more than 1500 faces. All faces were manually marked in all images using rectangles.

The testing procedure was simple: run the detection algorithm on each image and if the detector returns any regions as faces check for manual markings around those areas. If there is a manually marked rectangle around the detected region and if the rectangle is overlapping with the region for more than 80% of the area report the detected region as a hit otherwise the detected face region is a false positive. In the end the total number of hits divided by the total number of marked faces will give us the detection rate of the algorithm. The false positive rate was measured as the total number of false positives divided to the total number of marked faces.

The average detection rate was around 80% for most of the collections with an initial false positive return of around 50%.

This false positive rate is very high even for our system which, due to the recognition component is not sensitive to false positive face regions. Nevertheless such high levels of false positives do significantly increase the overall computational time for the recognition component and also affect the storage requirements.

One explanation for this high number of false positives is the fact that the detection algorithm was trained a few years back when the size of the images was much smaller than it is now. So having higher quality images as input means having a higher number of potential candidate regions which could lead, in turn, to a higher number of false positives. Our collections have an average picture size of approximate 1600x1400 pixels which corresponds to a digital resolution of 2 Mega pixels. Several refinements were next added to the basic face detector in order to reduce this high false detection rate.

Our first refinement to the system was to run the detection algorithm on resized images. By reducing the images to half size the detection rate did not change significantly but the false positive rate dropped by an average of about 20% across the image collections used for testing. Another advantage of resizing the images was that the detection time also improved by more than 4 times. It is worth mentioning that on original size images the detection time was around 1 second per image on a normal P4 workstation.

Examples of reducing the number of false positives by resizing the original images are shown in Figure 2.10.

Another algorithm refinement that was noticed during testing was that increasing the minimum number of adjacent consecutive candidates or co-located candidates at different scales reported as faces in order to report a combined final face region, the number of false positives will decrease especially for high resolution images.

When scanning the image using a small step for the sliding windows for different scales it is possible that more than one overlapping windows will be reported as faces by the verifier. There is a parameter in the implementation's API that defines how many neighboring windows have to be reported as faces in order to validate a common face region in the image. By increasing this parameter from the default value of 2 to 3 or 4 the number of false positive also decreased by more than 20%.

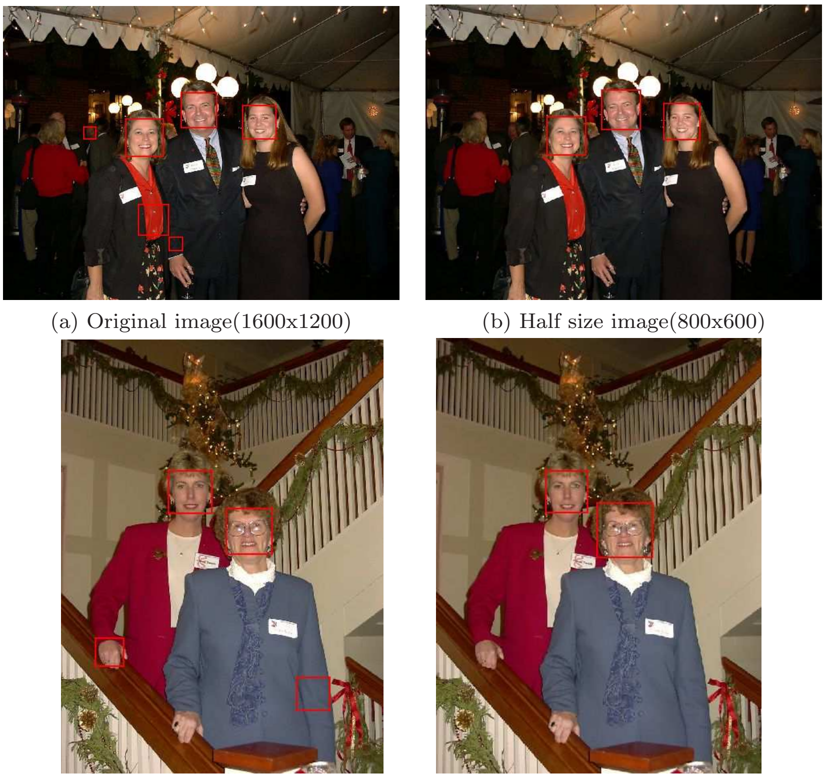
Improvements can also be made directly to the original algorithm and some of these will be discussed in detail in the next section.

Examples of detection results are given in the next figure Figure 2.11.

2.2.6 Improvements to the Algorithm

Improvements that can be applied to the algorithm can be divided according to the position of the supplemental filters into: pre-processing improvements and post-processing improvements.

The pre-processing methods include:



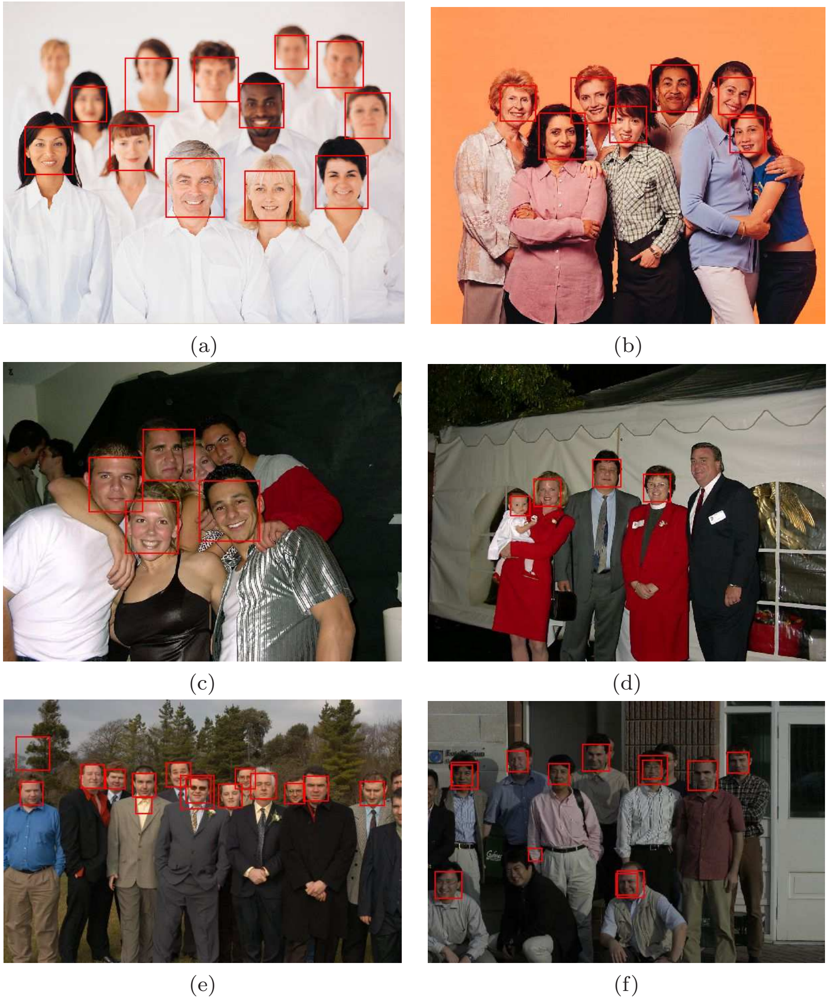
(a) Original image(1600x1200)

(b) Half size image(800x600)

(c) Original image(1200x1600)

(d) Half size image(600x800)

Figure 2.10: Eliminating false positives by resizing the images



(a) (b) (c) (d) (e) (f)

Figure 2.11: Face detection results

- applying a skin detection algorithm to the original image and use the detector only in the regions where skin regions are reported. This would decrease the overall detection time of the algorithm by excluding regions from the image where skin is not present. Such an approach was reported in [89] and the authors reported an increase of the algorithm speed by half and an improvement in the detection rate as well by using less classifiers in the cascade.

- changing the colour map where the features are computed. The original algorithm works on the greyscale images but it has been shown that using the difference between the red and green channels to compute the features of colour images, improves robustness of the face detector to shadow and illumination changes in natural scenes [69]. This method should improve the detection rate and also the false positives rate but it was not tested yet.

Post processing techniques can include additional filters, which test the candidate regions for presence or absence of different components in order to eliminate false positives from the reported face regions. These filters can search for presence of skin inside the reported regions or presence of different components of colours, or can search for different facial components like eyes, mouth and nose in order to validate face regions.

Post-processing techniques were not implemented in the final detection module used in our applications, as described in **Chapter 8**, but they were tested on multiple databases and noticeable improvements were demonstrated and broadly quantified. The reason for not using these improvements in the final implementation was not to affect the speed of the detection algorithm which, even with high false positive levels is quite efficient. Also, as stated before, the number of false positives generated by the detection is not critical to the outputs from the final system.

A histogram filter was tested based on statistical analysis on the collection of faces and multiple thresholds were applied on different colour components. The size of the histogram was set to 16 colours in order to increase speed. The number of false positives dropped around 10% using this extra filter and a small number of faces were undetected comparing with the original implementation.

Also we tested an eye detector after initial detection in order to validate face regions, and the results were similar to the results of the histogram filter.

These post-processing filters can improve the accuracy of the detection but could also affect the speed of the algorithm, which is one of the best properties of the proposed face detection system.

Chapter

3

Face Recognition

This chapter is dedicated to the main component of our Person Recognizer system: the face recognition component. The chapter is organized as follows:

In section 3.1 the problem of classifying and recognizing faces in digital images is introduced. We begin with a short overview of the problem and continue with a high level presentation of the principle approaches to face recognition which are described in the literature. An overview of the different requirements for face recognition algorithms according to the field of application is then given. This helps place our research in context, differentiating the needs of a consumer application from those, for example, of police forces or security agencies.

In section 3.2 a more detailed review of the main face recognition methods reported in the literature is given. Key points of each algorithm are presented along with an evaluation of their potential for incorporation in our application.

Section 3.3 presents the principle challenges associated with face recognition. It gives a general overview of different factors that affect the accuracy and robustness of face recognition algorithms. These factors have to be addressed when designing a face recognition system. For each factor possible solutions used in different recognition systems are presented.

In section 3.4 a high-level overview of the proposed face recognition system is presented. The system is based on combining two classical methods of face recognition: (i) the discrete cosine transform and (ii) principal component analysis. The PCA analysis is performed on the LL sub-band derived from the wavelet decomposition in order to increase the robustness of the recognition system. Section 3.5 describes the pre-processing techniques used prior to analysis, namely image resizing and enhancement.

Next, detailed presentations are given for each of the analysis methods used for recognition: (i) DCT in section 3.6 and (ii) PCA in the next section 3.7. For each method the mathematical model is initially presented and implementation considerations are given at the end of the section. Also a comparison of these two methods is given in section 3.7.3.

The last section of this chapter, section 3.8, provides a detailed description of the wavelet decomposition. Again it begins with a description of the mathematical techniques employed and continues with a discussion of practical implementation considerations.

3.1 Introduction

A general statement of the face recognition problem can be defined as follows: given an image or a sequence of images, identify or verify one or more persons in the image using a stored database of reference faces [120].

Identification or one-to-many recognition systems involves having an unknown input face, checking if the face is already in the database and if so obtain its identity. *Verification* or one-to-one recognition involves giving an input face which claims to be one of the identities in the database and deciding if the claim is true or false.

Face recognition continues to be one of the more challenging areas of the biometric based recognition/classification research and receives attention from a large community of researchers. There are many reasons which feed this interest; one of these is the wide range of commercial, law enforcement and security applications that require authentication; and second is the progress made in recent years regarding the methods and algorithms for data processing and also the availability of new technologies that makes it easier to study these algorithms.

Although other types of biometric measurement can be more reliable and often require less complex analysis (e.g. fingerprint or iris recognition) face recognition continues to fascinate because it offers a means of authentication where the user is not necessarily aware of their participation. Furthermore, given how reliable and accurate our in-built human face recognition process can be when employed, for example, in police line-ups, researchers are correspondingly challenged to develop automated systems which can be equally accurate and effective.

Although during recent years many face recognition systems have been developed in both academic [8] and industrial [9] environments which exhibit high accuracy and robustness, there is still not a common approach to the detection, analysis and classification of the face regions inside an image or a sequence of images. Each reported system has particular advantages and disadvantages that recommend them for the specific category of application they were originally designed for.

The main approaches to face recognition can be divided into local and global methods.

Local methods are based on a comparison of specific features inside the face area, or relationships between such features which are used to compute similarity measures between faces. The goal of such systems is first to detect corresponding features within detected face regions and then to compare these based on direct physical comparison of the features, or similarity measures determined from them.

Global methods are based on an overall measure of similarity between two face regions. Typically this implies projecting each face image into another mathematical space (feature space) where the distance between face projections corresponds to a measure of similarity between the original faces. The further apart two such projections are, the lower the likelihood that the two face regions match; conversely, once the two projections are within a certain critical range of each other there is a high probability that the two face regions correspond to each other.

A more detailed review of face recognition is given in the next section.

The requirements for face recognition system can vary considerably depending, in particular, on the type of application that the system has been designed for. Consider, for example:

(i) When a recognition system is to be used for high security access control or biometric based transactions the most important requirement for the system is to have a very low false acceptance rate. Such a system must apply very conservative criteria when deciding if an intruder is part of a trusted group of people, avoiding almost any possibility for false acceptances to occur.

(ii) Where a recognition system is used in surveillance type applications such as the detection of terrorists in places with large number of people, the most important factor for the system is that the false rejection rate, or failure to detect an individual that should be recognized, must be as low as possible in order to minimize the chances of such individuals being missed. Also the system has to be fast in order to cope with the huge amount of real-time data that is acquired. Naturally such a system will be susceptible to higher levels of false acceptance and will tend to generate significant number of false alarms, but in such an application it is better to be safe, rather than sorry!

(iii) In applications where large numbers of individual face regions are stored in the reference database another requirement may be that the size of data stored per face region be minimized in order to reduce the overall storage requirements for the resulting database and to optimize the access speed during searches of the database.

Other aspects of a face recognition algorithm which may need to be optimized according to the needs of a particular category of application include:

- the complexity of algorithms used in the recognition process which will influence the hardware involved in implementing the system
- the constrains needed for the algorithm to work properly, with "consequences" to the conditions that have to be satisfied for optimal results
- the level of human supervision/intervention during the recognition process which can limit the use of the face recognition system in fully automated applications.

But the principle requirements for every recognition system are obvious - high accuracy and robustness.

3.2 Face Recognition Literature Review

The topic of face recognition has caught the attention of researchers since the early days of image processing. The first approaches to automatic face recognition were based on an analysis of the geometrical properties of different face features. Kanade in [57] and Kelly in [59] used the distances between local features like eyes, nose and mouth to compute similarities between faces. These types of systems are members of the category of feature-based face recognition systems and are based on first detecting and then analyzing and comparing the geometrical properties of local features. The main disadvantage of such systems is that the corresponding local features must be accurately and repeatable detected. However trivial it may seem, this is not an easy task and face features like eyes, mouth and nose are not easy to detect consistently. The repeatability of detection for such local features will directly influence the results of the recognition process. Even slightly different synchronization between features could fool the system to take wrong decisions.

Once the local features are correctly detected, these recognition techniques can yield good results even for different illumination conditions, or for different face orientations if the relationships between facial features take changes in orientation into consideration.

More recent research into geometrical approaches was reported in [33] by Cox et al. and in [18] by Manjunath et al. These authors measured the response of different local regions to Gabor wavelet features. This response is equivalent to a measure of visual similarity between the compared regions and it has been proven that it is more robust to changes in illumination and different face poses than conventional geometric analysis of facial features.

Another important feature based approach is founded on the use of Hidden Markov Models (HMM), as described by Nefian in [74] and Samaria in [95] and [96]. Even though this approach is not based on finding the exact location of local features, it is based on the natural succession of these features. For instance when scanning the face from top to bottom the natural succession is: hair, forehead, eyes, nose, mouth and chin and we can model this succession as a statistical process. By using the HMMs to model this process one can compute the probability for a face region to belong to a certain HMM which was previously trained using face examples.

In order to test the HMM approach, different types of observation vectors extracted from the face region were used by researchers to train the HMM models. Initially the pixel values were used directly to train the model but the model was not robust to small changes in the image. Discrete Cosine Transform (DCT) and Principal Component Analysis (PCA) features were tested instead with improved results in recognition.

One of the advantages of using HMMs for recognition is that they can cope with small variations of the pattern to be modeled/classified. This robustness recommended HMM for classification and was proven by the widely usage of HMMs for speech recognition

Another important feature based approach is represented by the Elastic Bunch Graph Matching EBGM method used for face recognition by Wiskott et al. in [62] and Okada et al. in [76]. This method implies to manually define interest points (fiducial points) in the face image in the training phase, points that will become nodes in a graph. For each point features are computed (they are called bunch of jets). In the testing phase the corresponding jets are compared in order to compute similarities

A very large category of face recognition systems is represented by the holistic approaches where the face is analyzed as raw data by the recognition system. Usually this implies to project the data onto another space where the data can be separated more easily. The most common algorithm for face recognition from this group is represented by the Principal Component Analysis (PCA) method used for face recognition initially by Kirby and Sirovich in [60] and by Turk and Pentland in [109]. This approach is also called the eigenfaces method or the Karhunen-Loeve transform and will be described in detail later in this chapter.

Many variants of the eigenfaces method were developed and reported to have increased accuracy compared with these original approaches. One example is the method of probabilistic eigenfaces developed by Moghaddam and Pentland in [72].

Another well known subspace recognition approach is the Fisherfaces method developed by Swets and Weng in [105] and further refined by Belhumeur et al. in [15]. The Fisherfaces method is based on Linear Discriminant Analysis (LDA) projection which differs from PCA analysis by taking into consideration and optimizing the basis vector sets according to the variation within face classes and not just global variation across a set of faces as PCA does. LDA uses the class information and finds a set of basis vectors that maximize the between-class scatter while minimizing the within-class scatter.

One disadvantage to this approach is the fact that we must have the training database labeled with the name of the persons (we have to know the class distributions in order to compute the inner and inter class scatter matrices) which for automated retrieval applications like we intend to build it is not usable. This training requirement is needed in order to optimize the differences between classes (a class represent all images of the same person) and also the similarities between faces of the same class. Even if this requirement is fulfilled, it doesn't automatically mean that the results of the recognition will improve. In some cases it has been proved that the results of this approach are worse than the results of the classical PCA approach. This is the case especially when a small number of faces for each person are available for training.

Taking into consideration the characteristics of the images that we intend to work with, consumer images with large variations, this method that consider inner class distribution information can give worse results, as will be showed in the appendix.

Another related approach to the basic PCA technique is Independent Component Analysis ICA used for face recognition in [73] by Bartlett et al., where it has been shown that it can have more representative power than PCA for certain applications. Other studies [14] show that PCA can perform better than ICA in a face recognition set.

The difference between ICA and PCA is the fact that ICA seeks not a set of orthogonal components as PCA does for projecting the data, but a set of independent components. Independent here is used in the statistical sense. Two components are independent if any knowledge about one implies nothing about the other. This condition is stronger than uncorrelated components.

The common disadvantage of the last two approaches is that in order to obtain good results from the application of these methods the number of training faces available for each person has to be high (at least 3-4, but preferably more than 10) in order to form a good model of the most significant independent components to be used for the recognition model. This is especially obvious when large variations are present in the training collection.

Support Vector Machines (SVM) developed by Vapnik in [112] was also used for face recognition in [86] by Phillips. SVM is a binary classification method based on finding an optimal hyperplane as a decision surface between the two patterns to be classified. Different types of analysis can be used in order to extract from the face region the features used to train the SVMs models. In order to extend the binary classification to multi class recognition, different algorithms can be used based on combining multiple binary SVMs such as: one against one, one against all or Directly Acyclic Graph SVM (DAGSVM) [21].

Many neural network based face recognition approaches have been reported in the literature due to their property of generalization through learning by examples. One approach is given in [69] by Lin et al., called Probabilistic Decision-Based Neural Network PDBNN. Another approach based on evolution pursuit (EP) is given in [70] by Liu and Wechsler.

Convolution Neural Networks (CNN) have also been applied for face recognition in [65] by Lawrence et al. using self organizing (SOM) learning approach.

One disadvantage of the SVM and neural based approaches is that every time new samples are added in the collection, all the classifiers have to be re-trained, which could seriously affect the usability of the application for the common user. In the case of HMM recognition for instance, the models that have to be updated are only the ones with new samples added.

In order to combine the advantages of both approaches, holistic and feature based, hybrid techniques were developed. One example is the Modular eigenfaces presented in [85] by Pentland et al., which uses multiple views with different orientations of the same person for analysis. This way they can build a more robust model to be used for recognition. This method performs well but the requirements for training the models using multiple views are not viable for our type of application.

Another example of hybrid approach is Local Feature Analysis developed by Penev and Atick in [84] combined with PCA analysis.

We will comment further in section 3.4 on the reasons why many of these approaches are not suitable for our application: firstly, they require the apriori determination of a clustered distribution for the training set, and secondly, the wide variations in pose, illumination, and orientation which are typically encountered in the image sets of consumer collections will affect significantly the accuracy of these recognition techniques.

3.3 Face Recognition Challenges

Some of the most important factors that can influence and confound the accuracy of a face recognition system are:

Illumination: it is known [114] that variations in illumination can be more important than variations between individual characteristics, which can induce the system to decide that two different individuals with the same illumination are more similar than two instances of the same individual taken in different lighting conditions.

Two approaches can be used in order to obtain a recognition system invariant to illumination: the first one is to calculate features from the face image that are invariant to illumination, like distances between certain points of interest in the image (nose, mouth, eyes) and the second approach is to apply an illumination normalization method on the face region prior to analysis. The first approach represents one of the earliest approaches to face recognition but it has the drawback that the detection of the points of interest will directly influence the accuracy of the system and even if the distances between the points should not be influenced by the light conditions, the detection of the points is.

Second approach based on normalizing the illumination in the images is adopted in many recognition systems and the algorithm used for doing the normalization varies from simple histogram equalization to more complex algorithms like albedo maps [101] and contrast limited adaptive histogram equalization (CLAHE) [88, 27]. These algorithms are working well if the illumination variations are small, but there still isn't a common adopted method for normalizing the illumination in images which will work well in every type of illumination.

There are algorithms that are working well given some constrains regarding illumination variability, like, for example, if we consider the illumination as being uniform (no shadows) we can model the face as a Lambertian surface [66] and having a few images with controlled illumination we can model the influence of light over every face and make the new obtained faces to have the same illumination.

Pose: or face orientation is another import factor that can influence the accuracy of the recognition if it is not addressed. At the extreme, two profiles from different individuals may seem more similar than a profile and a frontal shot of the same individual. There are two known approaches that can be used in order to deal with this variation.

The first category of approach is that of multiple-view based approaches and consists in acquiring for each individual in the database multiple images with different orientations, in order to cover the entire domain of variations of the pose. If it is impractical to acquire a set of multiple images at incremental poses and we have only frontal images of each person, like a passport photo in security surveillance application, we can apply face modeling techniques in order to generate the required multiple images at incrementally different pose angles from a single image as described in [6]. The disadvantage of this approach is that we need to generate a large number of poses for training in order to cover the entire domain of pose variations and obtain a pose-invariant face recognition system. A simpler method of increasing the robustness to pose variations is to use in the test stage along with the original image also the mirrored image and to consider as the final distance the minimum between the two distances [55, 30]. The mirror imaged is obtained by swapping the values that have symmetric position from the middle of the vertical axe of the face.

A second approach, just like in the case of illumination variation, is to normalize the pose of all images in the database to have the same face orientation. In order to achieve that we can use face modelling algorithms such as 3D modelling [55] or 2D modelling algorithms such as Active Appearance Models (AAMs) [25, 28]. Using face modelling techniques the image can be transformed so that all the faces will be "straightened-up" into the same frontal orientation. We will discuss this approach in more detail in **Chapter 7**.

Size: it is difficult to compare a face image which is big with a lot of fine details with a small image where only few details are noticeable. The common approach for this is to resize all images to the smallest size acceptable for the system. By doing this, a lot of information from the face image will be lost. But that may not necessarily be a bad thing if the information that is lost correspond to noise or common face characteristics and is not useful for classification.

Quality: of the image: again it is difficult to compare high quality images with plenty of details with noisy, badly focused images. One way to deal with this kind of variability is to use features corresponding to the lower spectral frequencies of the image. In order to do that, low-pass filtering of the images prior to analysis is used [30, 63].

Another approach is to use in the classification stage only a part of the features that were calculated on the entire image. For instance, if we use the DCT coefficients for classification we can use only the first few coefficients corresponding to low frequencies and discard the later coefficients corresponding to higher frequencies. In another common approach, if the PCA-eigenfaces method [109] is used, a number of coefficients corresponding to fine details in the image are ignored in the recognition stage. Less information corresponding to fine details does make the system more robust to small variations in the image, but this information could also be useful in the recognition process. A compromise has to be made between the robustness of the system when ignoring the fine details from the face image and the accuracy of the system when keeping these details.

Number of training Images: in the training stage of the recognition process when a model is build for each person that we wish to recognize, it is important to have as many instances of the individual as is practical in order to cover a broad domain of variability in the appearance of that person. By incorporating in the model as much information as possible about the person, the model will be more robust and more accurate for recognition.

Naturally some people have more distinctive faces that others, and we have noticed that some people are easy to distinguish with only 1-2 training images, whereas others have very "average" faces and require large numbers of images to help distinguish them from others. Interestingly, these observations are somewhat independent of the actual recognition techniques used!

Expression of the face: facial expression can also be an important factor that affect the accuracy of the face recognition system. This influence can also be minimized by low-pass filtering of the face image as discussed above.

Changes in appearance: this has to be addressed especially for application designed to work over long periods of times but also over short periods because the appearance of the individuals can change very much due to facial hair, change of hair style, presence of glasses, make-up and over longer periods of time, the effects of aging.

Another important factor that can influence the results of a face recognition system is the level of automation that is required for the application. If the system is required to be fully automated, then prior to face recognition it is necessary to detect the face region inside the image. If this is done manually, eventually also marking other points of interest (like eyes, mouth), the face detection stage may help to increase the overall accuracy of the recognition system.

Every recognition system depends on how they address each of these factors, so they will have requirements/restrictions regarding acceptable illumination conditions, the angle of the face orientation, the 3-D pose of the face, the size of the face in the images, the quality of the captured image itself, the number of training images available for each individual when training the recognition system and so on.

Our goal is to find a combination of existing techniques which enables high levels of recognition combined with a lack of sensitivity to each of the above confounding factors.

3.4 Face Recognition System Description

Now that we have presented an overview of the current state-of-art in face recognition techniques we can return to a consideration of the key problem to be addressed by this research. As previously discussed there are a number of considerations in choosing the face recognition component of our system for *Person Recognition in Consumer Image Collections*.

Ideally it should be fully automatic. Thus we have taken the decision to use an automated face detection component as discussed in **Chapter 2**. It is also desirable that the training process for the face recognition component can be triggered automatically in normal circumstances. In fact the only point at which active user input is desirable is during the search/recognition phase when a user needs to select which person(s) to search for in the image collection.

A second consideration is the time needed for training the algorithm. This can be somewhat flexible as with appropriate system architecture it can be implemented as a background, or a distributed process which is hidden from the end user. Nevertheless it is desirable that the frequency of training updates and the time needed to perform each update should be minimized.

A third consideration relates to the searching/recognition phase, which should be as fast as is practical, especially when working with very large collections of images. A further related consideration is the size of the feature database which results after the training process. This should be kept as small as possible so it does not create unreasonable storage requirements or contribute to slowing down the search/recognition phase. At the same time a sufficient number of features must be stored to enable a good differentiation between individuals across a typical image collection.

Further, the images that a typical user will have in their collection will show wide variations in each of the factors discussed in section 3.2. Thus, it is also important that the system is sufficiently robust to variations in each of these factors. A series of techniques were tested in order to improve the robustness of the base recognition component, and these are discussed in detail in **Chapter 7**.

The accuracy of the base recognition component is of course a very important factor in making the retrieval system usable in applications. The accuracy of a recognition system can be described in terms of certain standardized metrics:

- **recognition rate**, which represents the number of correct decisions made by the system, and in one to one applications this rate is equal to the sum of correct **acceptance rate** (number of accepted authorized person) and correct **rejection rate** (number of rejected intruders),

- **false acceptance rate** which represents the number of intruders that are reported as authorized persons and;
- **false rejection rate**, which represents the number of authorized persons that are reported as intruders.

The last two measures represent the negative features of the system.

In a perfect system the recognition rate will be 100% and the false acceptance and false rejection rates will be zero. As we cannot reach these ideal measures, when designing the recognition system, and dependant on the type of application that the system will be used in, some of the rates will have higher or lower importance. For example in high security access control applications, the false acceptance rate is the most important factor and has to be kept to a minimum even if this means that the correct acceptance rate will drop. In such a system it is more important to prevent an intruder from gaining access than avoiding the inconvenience of an authorized user occasionally having to repeat the recognition process a number of times in order to gain access. In security surveillance applications such as searching for known terrorists in airports the false rejection rate should be minimized and the recognition rate should be as high as is practical, even if that means that "innocent" persons will frequently be falsely matched. In this system these false matches can be corrected afterwards through the participation of human supervisors in a follow-on interview process.

This compromise between acceptance and rejection rates is usually realized by setting a threshold for the similarity measure between faces like illustrated in Figure 3.1. Having the distance distributions for faces from the same person (red) and faces from different persons (blue) we can set the threshold to suit the needs of the application. In the ideal case the two curves would be completely separate.

Our retrieval application is not very strict regarding the false acceptance rate as long as the recognition rate is high enough. After we rank all the pictures in a collection according to their similarity with a test face, we want all the correct images to have a high rank, even if this means that occasionally some incorrect images will be inserted in the top page of the ranking list. A user will typically apply their own superior "face recognition" system and quickly ignore these images as being unrelated to their query. Thus as long as most of the matches are "good" these occasional "mistakes" will not unduly inconvenience a user and can be easily forgiven.

More important for our application is the robustness of the recognition system to different variations than strict restrictions regarding accuracy parameters (recognition rate and false acceptance and rejection rates). Still, these parame-

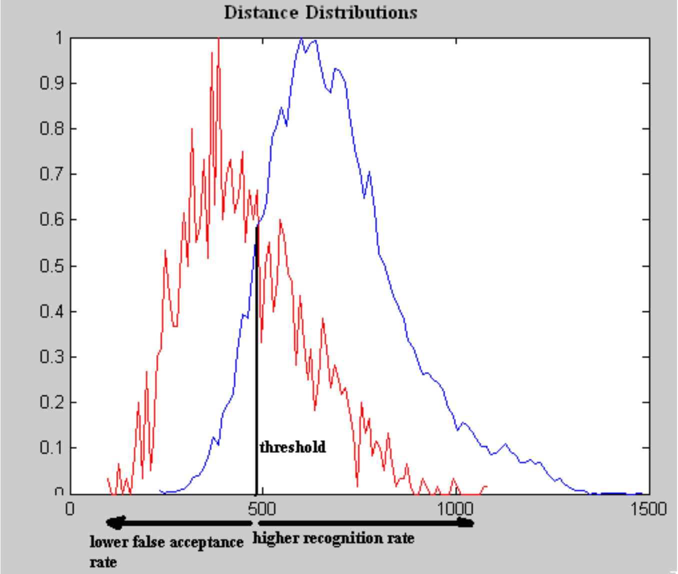


Figure 3.1: Threshold manipulation

ters cannot be ignored and will be used in order to test the functionality of the final system.

Keeping in mind that one of our key goals is to design a system which should require minimal user input, we are forced to restrict our choices for the face recognition component. Now because we cannot use any classification algorithm that requires training with pre-marked images any Neural Network based algorithms [65], Support Vector Machines [35, 50], Hidden Markov Models, Fisherfaces method [15], and other techniques that require manual image marking are eliminated. All these algorithms have been used successfully in face recognition system but are simply not suitable for our application.

As we explained before due to the nature of the images we intend to work with, it is difficult to implement a single face recognition system with good accuracy and robustness. Initially we tested different classical face recognition approaches and decided to combine two of them in order to obtain a better robustness. The first method we tested is the well known Principal Component Analysis (PCA) or the eigenfaces method. The PCA technique meets the requirements of our system regarding the training procedure (unlabelled training collection) and it is been proven to be useful in many face recognition applications.

The main drawback of the PCA approach is its low robustness to different variations in the image. We tried to minimize these variations by applying a series of pre-processing algorithms prior to analysis. These are described later in this chapter.

The second face recognition algorithm uses Discrete Cosine Transform (DCT) coefficients for computing face similarities. This method is also very well known for face recognition and has the advantages of being very fast and showing good recognition results. It is also influenced by different variations in the image and needs pre-processing techniques to improve its robustness.

By testing each of the methods separately on different databases we noticed that their results are influence by different factors. Depending on the variations present in the testing collections, the results of the two methods seemed complementary which influenced us to try to combine their results for a better robustness.

The architecture of our face recognition system is represented in Figure 3.2. The system is based on combining two well known techniques which were modified in order to obtain a more robust system as described above: Principal Component Analysis (PCA) [109] and Discrete Cosine Transform (DCT). Both methods are well researched in the literature and they have advantages and disadvantages that will be further discussed.

By combining the results of both methods we expect to obtain a higher accuracy and also a higher robustness to different variations. **Chapter 6** will describe how we reached this final configuration by testing each approach separately and unmodified. By changing configuration parameters and combining the two methods the proposed method showed optimal results.

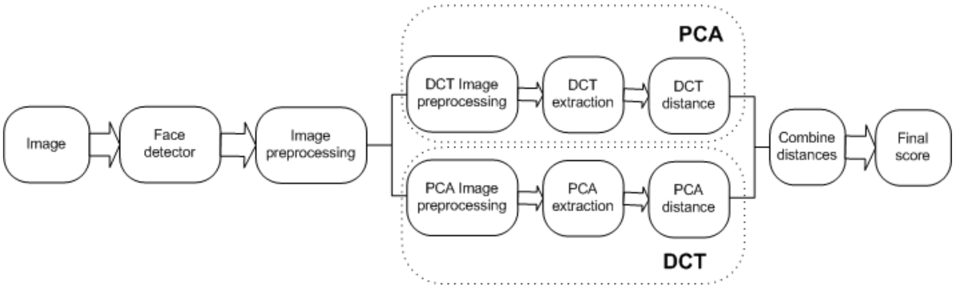


Figure 3.2: Face recognition system architecture

The first step of the system is to detect the face regions in an image. In order for the final system to be fully automatic, the detection process has to be fully automatic also. Our system uses a face detection component which is largely based on the face detection algorithm of Viola-Jones [113] as described in **Chapter 2**. This is the fastest and most accurate face detection system we tested and is recognized in the literature as the current state-of-art.

Prior to feature extraction, a series of image preprocessing techniques are applied in order to minimize the effects of different factors that may influence the accuracy of the face recognition component of our system. These techniques include:

- normalizing the size of all the face regions in the collection to a standard size
- normalizing the illumination of the regions using histogram equalization method or CLAHE - Contrast Limited Adaptive Histogram Equalization method.

Other preprocessing techniques were applied, separately for each type of feature extraction. For DCT features we transformed color images to grayscale face images. We also tested DCT on indexed images using different colormaps. The results will be presented in **Chapter 6**. **Chapter 4** has a section dedicated to changing image colormaps in order to compute different features.

Prior to PCA analysis the color images were transformed only to grayscale images and prior the first level wavelet decomposition was applied. For feature extraction only one sub-band from the decomposition was used. The sub-band used corresponds to the low-pass filtering on both horizontal and vertical directions and is called the 1LL sub-band. By using only this sub-band we ignore fine details in the images that correspond to noise and small variations of the face thus making the recognition system more robust. A detailed explanation of this techniques is given in section 3.8.

Another refinement to improve the robustness of the system to face orientation was the use the mirrored image of the face to be recognized in the testing phase prior to PCA analysis. Note that this method is not beneficial for DCT analysis because the spectrum of the original image and the spectrum of its mirrored version are identical due to the properties of the DCT transform.

Finally having obtained similarity scores from both DCT and PCA methods we combine these in order to obtain a final score. This combination is based on a statistical method which will be described later in this chapter.

3.5 Image Pre-processing Techniques

3.5.1 Image Resizing Using Bicubic Interpolation

There are many techniques that can be used to enlarge or reduce the size of an image. These methods generally realize a trade-off between speed and the degree to which they reduce the visual artefacts in the resulting image. Simple methods for resizing an image by a factor of n, are to duplicate each pixel n^2 times in order to enlarge by a factor of n, and for reducing the image by the same factor we keep every n^{th} pixel discarding the others. Of course this will lead to more pronounced jagged edges than the ones in the original image when enlarging the images, and to aliasing of high frequency components when reducing the size of the image.

More complex methods of changing the size of an image by an arbitrary amount require interpolation of the colours between pixels.

The simplest method of resizing an image using interpolation is known as *nearest neighbor interpolation*. Using this method one finds the closest corresponding pixel in the original image (i, j) for each pixel in the destination image (i', j'). If the source image has dimensions width w and height h and the destination image w' and h', then a point in the destination image is obtained using the next formulas:

$$i' = i * \frac{w'}{w} \tag{3.1}$$

$$j' = j * \frac{h'}{h} \tag{3.2}$$

where the division operation above is rounded to an integer. This form of interpolation can suffer from unacceptable aliasing artifacts when used to enlarge and to reduce images.

Another technique, bilinear interpolation, uses the value of 4 neighboring pixels in the source image in order to compute a new pixel value for the resized image.

Figure 3.3 represents the notations used in the next equations. We wish to determine the colour of every point (i', j') in the destination image. There is a linear scaling relationship between the two images, in general a point (i', j') corresponds to a non integer position in the original image. Thus its position is given by:

$$x = i * \frac{w'}{w} \tag{3.3}$$

$$y = j * \frac{h'}{h} \tag{3.4}$$

The nearest pixel coordinates (i, j) are the integer parts of x and y, and dx and dy in Figure 3.3 diagram are the differences between these, $dx = x - i$, $dy = y - j$.

The value of the pixel (i', j') in the new resized image I' using bilinear interpolation is obtained using the formula (note: $t = dx$ and $u = dy$):

$$\begin{aligned} I'(i', j') = (1-t) * (1-u) * I(i, j) + (t) * (1-u) * I(i+1, j) + \\ u * (1-t) * I(i, j+1) + t * u * I(i+1, j+1) \end{aligned} \tag{3.5}$$

The standard approach is called bicubic interpolation and it estimates the

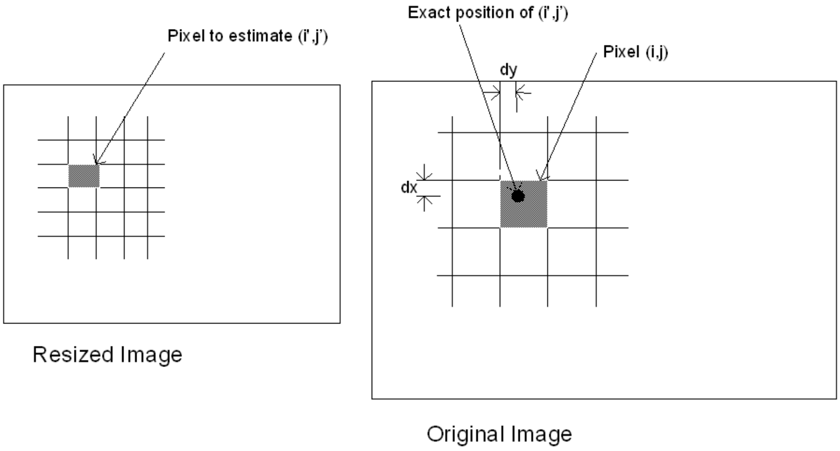


Figure 3.3: Image resizing using interpolation

colour of a pixel in the destination image by an average of 16 pixels surrounding the closest corresponding pixel in the source image. There are two methods commonly used for interpolating the 4x4 pixel, cubic B-Spline discussed further and a cubic interpolation function.

The equation below gives the interpolated value, and is applied to each of the red, green, and blue components. The m and n summation span a 4x4 grid around the pixel (i,j).

$$I'(i',j') = \sum_{m=-1}^{2}\sum_{n=-1}^{2} I(i+m,j+n) * R(m-dx) * R(n-dy) \tag{3.6}$$

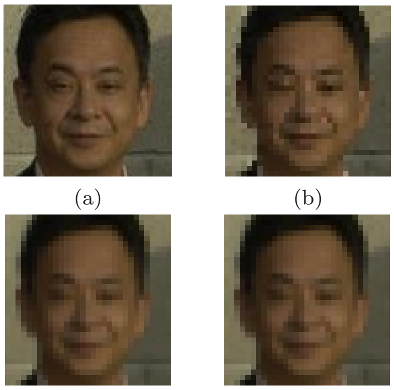
The cubic weighting function $R(x)$ is given below:

$$R(x) = \frac{1}{6}(P(x+2)^3 - 4P(x+1)^3 + 6P(x)^3 - 4P(x-1)^3) \tag{3.7}$$

and

$$P(x) = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases}$$

Figure 3.4 represents an image resized using the techniques described earlier. The dimension of the resulted image is 3 times smaller than the original one.



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

Figure 3.4: Resized images using different methods: (a) original image (105x105), (b) Resized image using nearest neighbor method(35x35), (c) Resized image using bilinear interpolation (35x35) (d) Resized image using bicubic interpolation (35x35)

3.5.2 Histogram Equalization

The histogram of a digital image with grey levels is defined as the probability for a pixel to have a certain grey level and can be computed as a discrete function:

$$p(r_k) = n_k/n \tag{3.8}$$

where r_k is the k^{th} grey level, n_k is the total number of pixel in the image with that grey level and n is the total number of pixels in the image.

Histogram modeling techniques (e.g. histogram equalization) provide a method for modifying the dynamic range and contrast of an image by altering the image such that its intensity histogram has a pre-defined shape. Histogram equalization employs a monotonic, non-linear mapping which re-assigns the intensity values of pixels in the input image such that the output image contains an uniform distribution of intensities (i.e. a flat histogram).

Histogram modeling is usually introduced using continuous, rather than discrete process functions. Therefore, we suppose that the images of interest contain continuous intensity levels (in the interval [0,1]) and that the transformation function f which maps an input image I onto an output image I' is continuous within this interval. Further, it will be assumed that the transfer law (which may also be written in terms of intensity density levels (e.g. $s = f(r)$) is single-valued and monotonically increasing (as is the case in histogram equalization) so that it is possible to define the inverse law $r = f^{-1}(s)$, where variables r and s represent the grey levels in image I and respectively I'. An example of such a transfer function is illustrated in Figure 3.5.

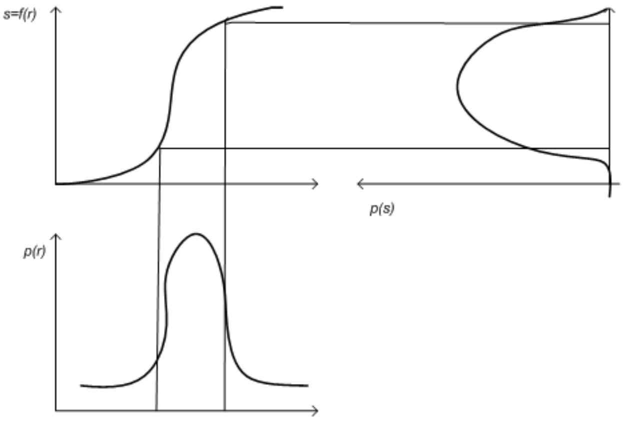


Figure 3.5: Example of transfer function for histogram equalization

If r and s are continuous variables, the original and transformed grey levels can be characterized by their probability density functions $p_r(r)$ and $p_s(s)$. From probability theory, if $p_r(r)$ and $f(r)$ are known and $f^{-1}(s)$ is single-valued and monotonically increasing, then the probability density of the transformed grey levels is:

$$p_s(s) = \left[p_r(r) \frac{dr}{ds} \right]_{r=T^{-1}(s)} \tag{3.9}$$

This enhancement technique is based on modifying the appearance of an image by controlling the probability density function of its grey levels via the transformation function $f(r)$.

Let us consider the transformation function

$$s = f(r) = \int p_r(w)dw \tag{3.10}$$

where w is a dummy variable of integration.

The last part of the equation is recognized as the cumulative distribution function of r or the cumulative histogram of image I.

From eq. 3.10 the derivative of variable s with respect to r is:

$$\frac{ds}{dr} = p_r(r) \tag{3.11}$$

If we substitute in eq. 3.9 we obtain:

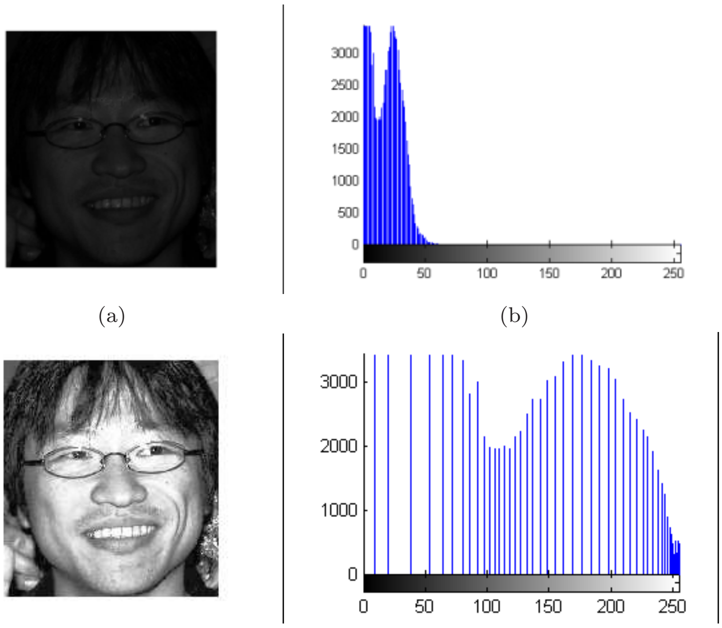
$$p_s(s) = \left[p_r(r) * \frac{1}{p_r(r)} \right]_{r=f^{-1}(s)} = [1]_{r=f^{-1}(s)} = 1 \qquad 0 \leq s \leq 1 \tag{3.12}$$

which is an uniform density in the interval of definition of the transformed variable s.

The discrete form of eq. 3.10 which can be used in digital image processing is:

$$s_k = f(r_k) = \sum_{i=0}^{k} \frac{n_i}{n} = \sum_{i=0}^{k} p_r(r_i) \tag{3.13}$$

This result indicates that using a transformation function equal to the cumulative histogram of the original image, produces an image whose grey levels have an uniform density. In terms of enhancement, this result implies an increase in the dynamic range of the pixels, which can have a considerable effect in the appearance of the image.



(a)

(b)

(c)

(d)

Figure 3.6: Example of the effects of histogram equalization

3.6 The Discrete Cosine Transform (DCT)

The DCT is part of the transform functions family that attempts to decorrelate data sequences and it is usually used in data transmissions prior to compression. This is possible because after decorrelation each transform coefficient can be encoded independently without loosing compression efficiency. Another important property of the DCT transform is its ”energy compaction” which means that most of the signal information tends to be concentrated in a small number of coefficients situated in the low part of the frequency band. This also inspired the use of the DCT transform for data classification by comparing only the useful information between data samples and ignoring redundant and corrupted information.

Just like the Fourier transform, the Cosine transform describes the frequencies spectrum of the data sequence. The difference is that the Cosine transform uses only real numbers ignoring the imaginary part.

3.6.1 The One-Dimensional DCT

The formal definition of the DCT transform of a 1-D data sequence $f(x)$ of length N is:

$$C(k) = \alpha(k) * \sum_{x=0}^{N-1} f(x) * \cos\left[\frac{\pi(2x+1)k}{2N}\right] \tag{3.14}$$

for $u = 0, 1, 2 \ldots N-1$

Similarly the inverse transform is defined as:

$$f(x) = \sum_{k=0}^{N-1} \alpha(k) * C(k) * \cos\left[\frac{\pi(2x+1)k}{2N}\right] \tag{3.15}$$

for $x = 0, 1, 2 \ldots N$

In both equations eq. 3.14 and eq. 3.15 $\alpha(k)$ is defined as

$$\alpha(k) = \begin{cases} \sqrt{\frac{1}{N}} & for \quad k = 0 \\ \sqrt{\frac{2}{N}} & for \quad k \neq 0 \end{cases} \tag{3.16}$$

It can be noted from eq. 3.13 that for $k = 0$, the first coefficient is:

$$C(0) = \sqrt{\frac{1}{N}} \sum_{x=0}^{N-1} f(x) \tag{3.17}$$

Thus the first coefficient is the average value of the sample sequence. In the literature this coefficient is named *the DC coefficient* and all the others coefficients are called *AC coefficients.*

It can be noticed from eq 3.14 that each DCT coefficient is computed as a linear combination between the data samples and cosine terms of type $(\sum_{x=0}^{N-1} \cos\left[\frac{\pi(2x+1)k}{2N}\right])$ which are called cosine basis functions. If we analyze the particular case of $N = 8$ we can plot these cosine terms varying k like in Figure 3.6. As we already stated, the first waveform for $k = 0$ renders a constant value (DC), and all the other waveforms give signals at progressively increasing frequencies.

It can be observed that these basis functions are orthogonal. Hence, any multiplication of the waveform with another waveform followed by a summation over all sample points gives a zero (scalar) value, whereas any multiplication of the waveforms with itself followed by a summation yields a constant (scalar) value. Orthogonal signals are independent, that is, none of the basis functions can be represented as a combination of other basis functions.

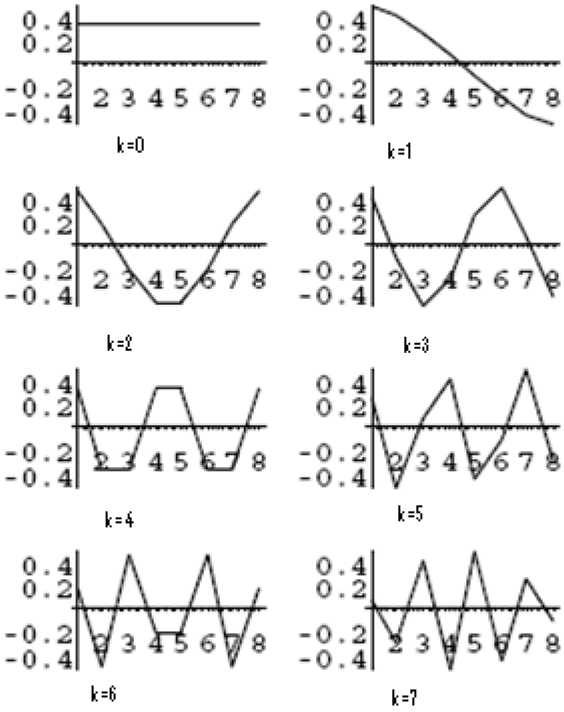


Figure 3.7: The basis vectors for 1D DCT for N=8

If the data sequence has more than N sample points it can be divided into sub-sequences of length N and DCT can be applied to each sub-sequence independently. For each sub-sequence the basis vectors remain the same, only the data samples will be changed. If the basis vectors are pre-computed and stored in memory, this will reduce the number of mathematical operations (i.e., multiplications and additions) thus significantly enhancing the computational efficiency of the algorithm.

3.6.2 The Two-Dimensional DCT

When dealing with 2-D data such as images a direct extension of the 1-D DCT to two dimensions is used. This 2-D DCT is defined as:

$$C(k,u) = \alpha(k) * \alpha(u) * \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) * \cos\left[\frac{\pi(2x+1)k}{2N}\right] * \cos\left[\frac{\pi(2x+1)u}{2N}\right] \tag{3.18}$$

for $k, u = 0, 1, 2 \ldots N-1$ and $\alpha(k)$ as defined in eq. 3.16

The inverse transform is defined as:

$$f(x,y) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \alpha(k) * \alpha(u) * C(k,u) * \cos\left[\frac{\pi(2x+1)k}{2N}\right] * \cos\left[\frac{\pi(2x+1)u}{2N}\right] \tag{3.19}$$

for $x, y = 0, 1, 2 \ldots N-1$

The 2-D basis vectors can be generated by multiplying the horizontally orientated 1-D basis functions from Figure 3.7 with the vertically orientated set of the same functions. The obtained basis vectors for $N = 8$ are illustrated in Figure 3.8. Note that the basis functions represent a progressive increase in frequency in both vertical and horizontal directions. The first function is the DC component and it is the result of multiplying the DC component from Figure 3.7 with its transpose, therefore being a constant value.

Some examples of images and their DCT spectrum are given in Figure 3.9

3.6.3 Properties of DCT Transform

In this section we describe some of the key properties of the DCT transform which recommend the technique for use in pattern recognition systems and for image analysis and compression.

- **Decorrelation:** This property is very important for data encoding because the transformation removes the redundancy between neighboring pixels so each coefficient can be encoded independently.

- **Energy compaction:** This is very useful in both data encoding and classification. By using only a small number of coefficients from the transformation where all the information from the original data is compacted, the efficiency of the encoder is highly increased, and by ignoring the rest of the coefficients the visible effect is hardly noticeable. Also in pattern recognition, by ignoring the coefficients corresponding to frequency domain where noise is most likely to be present, the robustness of the classification is improved.

- **Separability:** It can be observed if we re-arrange the formula for 2D-DCT

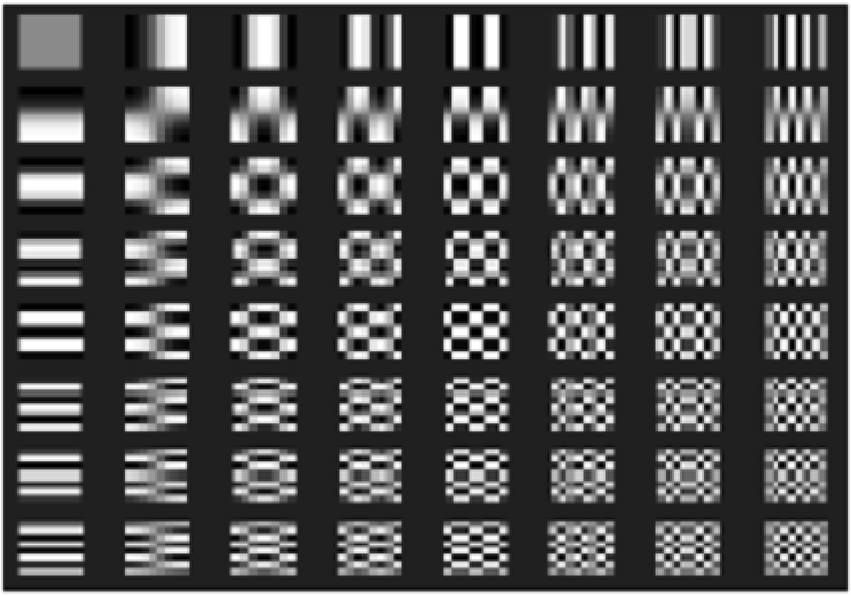


Figure 3.8: The basis vectors for 2D DCT for N=8. The highest value is shown in white.

(eq 3.18) that we can split the calculus in two steps using the formula for 1D-DCT applied first in the rows of the image and then on the columns of the image.

$$C(k,u) = \alpha(k) * \alpha(u) * \sum_{x=0}^{N-1} \cos\left[\frac{\pi(2x+1)k}{2N}\right] \sum_{y=0}^{N-1} f(x,y) * \cos\left[\frac{\pi(2x+1)u}{2N}\right] \tag{3.20}$$

for $k, u = 0, 1, 2 \ldots N-1$ and $\alpha(k)$ defined in 3.16

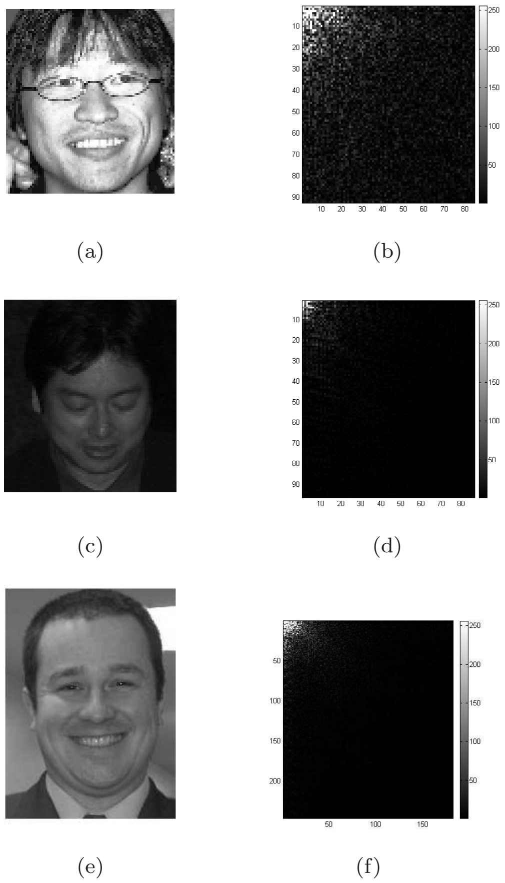
A graphical representation of DCT separability is given in the next figure. A similar algorithm is applied in order to compute the inverse transform.

This property is very useful for fast implementation of the transform.

- **Symmetry:** A separable and symmetric transform is defined by the following expression:

$$T = AfA \tag{3.21}$$

If we analyze eq 3.20 we can identify A as an NxN symmetric matrix with



(a) (b)

(c) (d)

(e) (f)

Figure 3.9: Face detection results

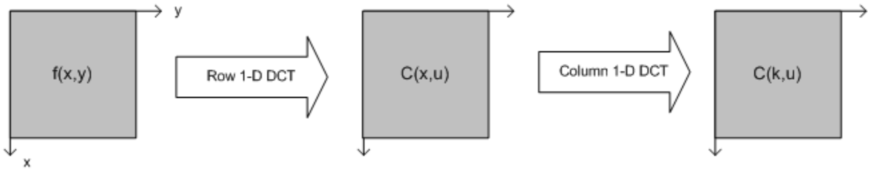


Figure 3.10: Computing 2D-DCT using 1-D DCT formula

coefficients $\alpha(i, j)$ given by :

$$\alpha(i, j) = \alpha(j) * \sum_{x=0}^{N-1} \cos\left[\frac{\pi(2x+1)i}{2N}\right] \tag{3.22}$$

and f from eq 3.21 is the NxN matrix image. If we pre-compute the transformation matrix A, the transformation will imply only matrix multiplication which will boost the computation efficiency of the transformation.

- **Orthogonality:** Taking into consideration eq 3.21 the inverse transform is defined by:

$$f = A^{-1}TA^{-1} \tag{3.23}$$

As previously stated, the DCT basis vectors are orthogonal. Thus the inverse of the transformation matrix A is equal to its transpose $A^{-1} = AT$. Further, in addition to its decorrelation characteristics, this property leads to some reduction in the pre-computation complexity.

3.7 Principal Component Analysis (PCA)

Principal component analysis (PCA) is a classical statistical algorithm that transforms a number of potentially correlated variables into a smaller number of uncorrelated variables called principal components. The objective of principal component analysis is to reduce the dimensionality, or number of variables, of the input dataset retaining most of the original variability in the data.

Thus the first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible.

3.7.1 Mathematical Model

Principal component analysis is based on the statistical representation of a random variable. Let us assume we have a random vector of n multidimensional variables $\mathbf{x}$ defined as:

$$\mathbf{x} = [x_1, x_2 \ldots x_n] \tag{3.24}$$

The statistical mean of the random variable is denoted by:

$$\mu_{\mathbf{x}} = E\{\mathbf{x}\} \tag{3.25}$$

and the covariance matrix of the data set is defined by:

$$C_x = E\{(\mathbf{x} - \mu_{\mathbf{x}}) \times (\mathbf{x} - \mu_{\mathbf{x}})^T\} \tag{3.26}$$

The components of the covariance matrix: c_{ij} represent the covariance between data sample x_i and x_j and the component c_{ii} represent the auto-variance of the data sample x_i. The covariance matrix is by definition symmetric $c_{ij} = c_{ji}$. The variance of a component indicates the spread of its values around the mean value and the covariance between two data samples gives information on the similarity between them.

From the covariance matrix which is symmetric we can calculate the orthogonal basis by finding its eigenvectors and eigenvalues. The eigenvectors $\mathbf{E_i}$ have the same dimension as the input data and can be computed using the formula:

$$C_x \times \mathbf{E_i} = \lambda_i \times \mathbf{E_i} \tag{3.27}$$

for $i = 1, 2 \ldots n$

where λ_i represents the eigenvalues of the covariance matrix and can be computed as the solution for next equation:

$$|C_x - \lambda I| = 0 \tag{3.28}$$

where I is the identity matrix with the same dimensions as the covariance matrix and operator $| * |$ denotes the determinant of the matrix. Equation 3.28 is easy to solve if n is small but becomes a non-trivial task if n is high. Other methods exist in order to solve this equation, such as using a neural network trained to obtain the wanted results.

By ordering the eigenvectors in the descending order of the corresponding eigenvalues, we can obtain an ordered orthogonal basis with the first eigenvector having the direction of the largest variation in the data set. This way we can find the directions in which the data sets have the most significant amounts of energy.

In the same way as for the DCT transform we can write the PCA transform in matrix form as:

$$\mathbf{y} = A \times (\mathbf{x} - \mu_{\mathbf{x}}) \tag{3.29}$$

where x is the input data, y is the transform data, $\mu_{\mathbf{x}}$ is the statistical mean of the input data and A is the transformation matrix which contains all the eigenvectors E_i.

We can reconstruct the input data x from y using the inverse transform:

$$\mathbf{x} = A^T \times \mathbf{y} + \mu_{\mathbf{x}} \tag{3.30}$$

Similar to the DCT transform, A is an orthogonal matrix, therefore $A^{-1} = A^T$. The original data x was projected on the coordinates axes defined by the orthogonal basis. The original data was then reconstructed as a linear combination of the basis vectors which is often referred to as the definition of PCA.

If we choose to use only a few of the basis vectors corresponding to the largest eigenvalues we obtain the new transformation:

$$\mathbf{y} = A_k \times (\mathbf{x} - \mu_{\mathbf{x}}) \tag{3.31}$$

Then the original data can be generated using:

$$\mathbf{x} = A_k^T \times \mathbf{y} + \mu_{\mathbf{x}} \tag{3.32}$$

where A_k represents the transformation matrix and includes as rows only the first k eigenvectors.

This means that we can project the original data onto a set of coordinate axes having a dimensionality of k and transforming the data back as a linear combination of the determined set of basis vectors. This minimizes the mean-square error between the data and its representation for a given number of eigenvectors.

This property is the equivalent of the energy compaction property of the DCT transform. If the data is concentrated in a linear subspace, this provides a means to compress it without losing information and at the same time we can simplify its representation by using a smaller number of coefficients. By using the eigenvectors with the largest eigenvalues, we lose as little information as possible in the mean-square sense.

Note that we can choose a fixed number of eigenvectors and their respective eigenvalues and get a consistent representation. This preserves a varying amount of energy of the original data. Alternatively, we can fix the amount of energy and we will obtain a varying amount of eigenvectors and their respective eigenvalues. This gives an approximately consistent amount of information at the expense of varying representations with regard to the dimensionality of the subspace.

Thus PCA offers an efficient way to control the trade-off between losing information and simplifying the representation of the data in the sense of the number of coefficients. On one hand, we can simplify the problem by reducing the dimensionality of the PCA representation. On the other hand we want to preserve as much as possible from the original information content so the coefficients which are discarded should be from those PCA components which exhibit the least variation across the entire data set, and thus contain the least amount of useful information.

3.7.2 PCA for Face Recognition

Having in mind the conclusion reached at the end of the previous section, we can think of using PCA transform on each face image and use for comparison only a few coefficients corresponding to the largest eigenvalues which carry useful information for the classification of face regions. In order to achieve this we have to transform each image into a vector and regard the resulting vector as a multidimensional random variable. The steps of the algorithm are:

(i) first compute the mean face of the collection of faces like the mean face illustrated in Figure 3.11, using the formula for the statistical mean ($\hat{I} = \sum_{j=1} nI_j/n$).

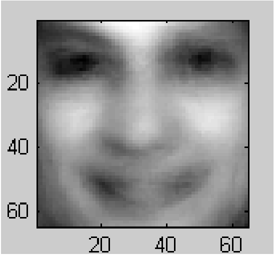


Figure 3.11: Example of mean face (64x64)

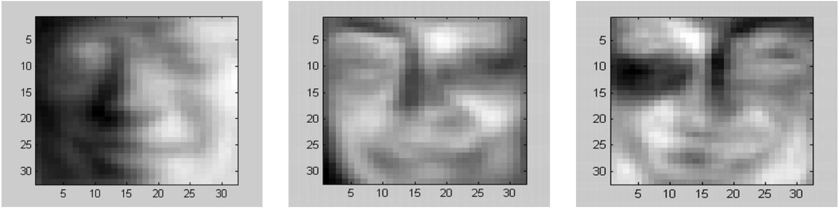
(ii) Next subtract the mean face from every face in the collection

(iii) Compute the covariance matrix using eq 3.26

(iv) From the covariance matrix compute its eigenvalues and eigenvectors using the formulas 3.27 and 3.28.

The obtained eigenvectors are commonly referred to as eigenfaces if they are transformed from vectors back to matrices and plotted as images. This is why PCA for face recognition is usually referred to as the eigenfaces method. Examples of the first few eigenfaces are given in Figure 3.12.

(v) After we compute the eigenfaces we can project each face in the databases using a certain number of eigenfaces as basis vectors. The obtained weights are called PCA coefficients and can be used in the classification stage.



(a) (b) (c)

Figure 3.12: First three eigenfaces (32x32)

Figure 3.13 represents the eigenvalues evolution across their ranks. It can be observed that the eigenvalues have a very small value just after few ranks. We can fix a threshold and discard the eigevalues that are smaller than the threshold. We can compute the threshold by conserving a certain amount of energy (95%-99%) from the entire energy in the eigenvalues.

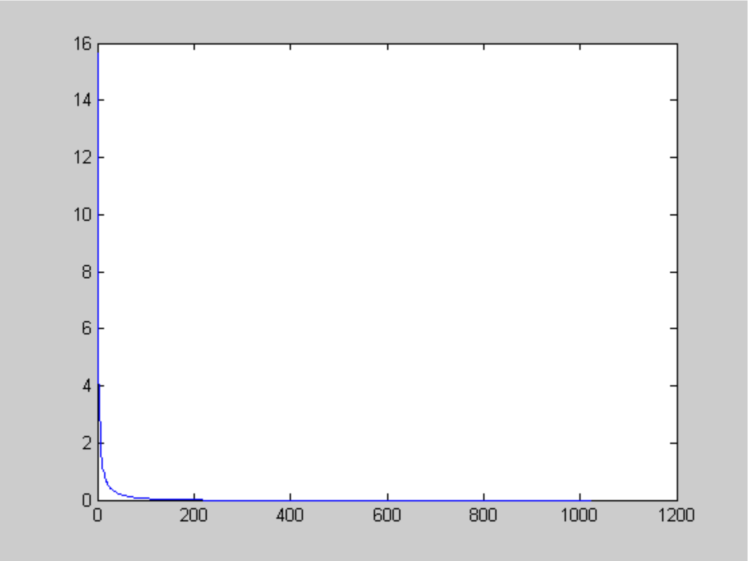
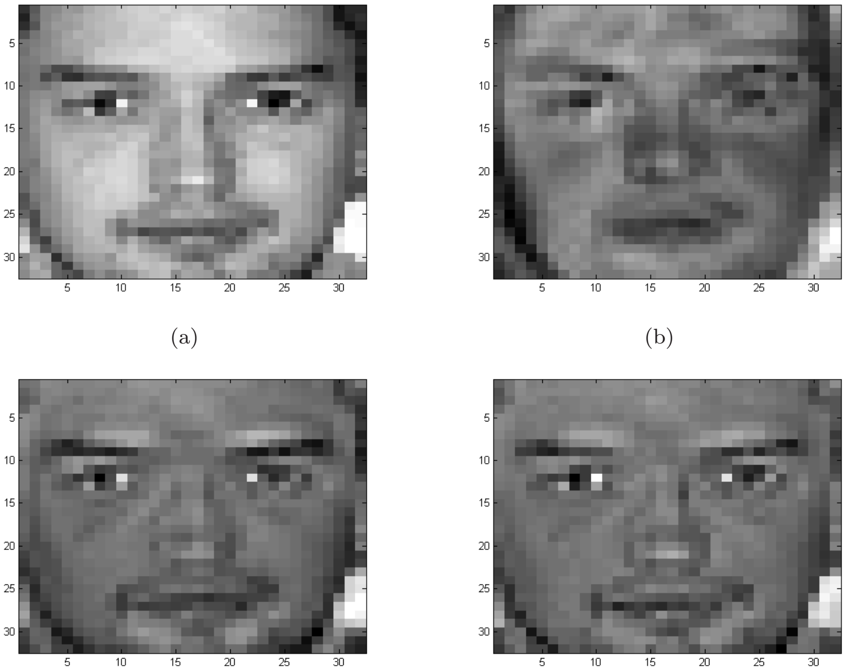


Figure 3.13: Eiegenvalue variations for images 32x32

(vi) Alternatively we can recreate the original faces as linear combination of the eigenfaces weighted by the PCA coefficients. In Figure 3.14 we can see the original image and the recreated face using different number of eigenfaces.

It can be observed from Figure 3.14 that the image reconstructed using a smaller number of eigenvectors keeps the original features of the input image. Adding new eigenvectors will improve the fine characteristics of the input image. The difference between using 20 and 50 coefficients is very small, which indicates



(a)

(b)

(c)

(d)

Figure 3.14: Image restoration using different number of eigenfaces a) Original image, b) Image using first 20 eigenvectors c) Image using first 5 eigenvectors d) Image using first 50 eigenvectors

that by determining the optimal number of coefficients we will be able to encode only the coefficients that are important for the face characteristics.

The main advantage of the PCA method is the small dimension of the feature vector used for classification. This is useful for speeding the classification process and also for decreasing the storage requirements for the feature database. The disadvantages are that, firstly, the method is data dependent which means that every time the face collection changes we have to re-compute the basis vectors; in turn this will slow down the training process; secondly, the method is known to be quite susceptible to variations in illumination and face orientation. Without proper pre-processing which address those variations the PCA method will not be very useful in our system.

3.7.3 Comparison between DCT, DFT and PCA

The PCA transform is a linear transform where each data sample can be written as a linear combination of basis vectors which are taken from the statistical properties of the data samples. It is an optimal transformation from the perspective of energy compaction, meaning that it places as much energy as possible in as few coefficients as possible.

However, the transformation kernel is generally not separable and a full matrix multiplication must be performed. In other words, PCA is data dependent and therefore a fast implementation (like FFT) is not possible. Derivation of the respective basis for each image sub-block requires significant computational resources.

Although there are reported fast algorithms for PCA, the overall complexity of the algorithm is significantly higher than DCT/DFT algorithms.

The Discrete Fourier Transform (DFT) transformation kernel is linear, separable and symmetric just like DCT. Hence it has fixed basis vectors and fast implementations are possible. It also presents good properties of decorrelation and energy compaction. However, DFT is a complex transform and therefore requires encoding of both magnitude and phase information.

In addition, studies have shown that DCT provides better energy compaction for natural images.

3.8 Wavelet Transform

It is well known [15] that the PCA recognition technique is very sensitive to small variations of the face regions. In order to increase the robustness of the technique to such variations, we can filter the high frequency components from the image prior to applying the PCA transformation. These components are commonly caused by noise added over the image, and also correspond to image components which are not useful in recognition.

Wavelet decomposition over the input image has the mathematical property of filtering the image on horizontal and vertical directions using low-pass and high-pass filters.

Applying the PCA analysis over the sub-band corresponding to the low pass filtering on both horizontal and vertical directions should increase the robustness and the accuracy of the recognition.

Fast algorithms for wavelet decomposition exist, thus eliminating the disadvantage of using the more complex wavelet filtering technique instead of simple low-pass filtering.

Compared with the Fourier transform which deals with transforming the time domain components into frequency domain components, the wavelet transform provides multi-resolution analysis using mathematical structures that provide varying time/frequency windows for analysis.

The wavelet transform has the property that it can identify frequency components, simultaneously with their locations in time. Also the number of computations is directly proportional to the length of the input signal. The complexity of wavelet transform as shown in [21] can be $O(n)$ where n is the length of the signal; when compared to the Fast Fourier transform which requires $O(n*log(n))$ computations, in some cases the wavelet transform will prove even more efficient.

Multi-resolution analysis provided by the wavelet transform solves the problem that Fourier analysis rise when analyzing non-stationary signals where frequencies are not present in the entire time domain of the signal even when using short-time Fourier analysis designed for non-stationary signals.

It is well known that if we want good resolution in time we must use a small window for analysis, and for good frequency resolution we must have a large window for analysis.

Wavelet transform solves the problem by using for temporal analysis a short low frequency version of the prototype wavelet and for frequency analysis a long high frequency version of the prototype wavelet (see Figure 3.15).

The wavelets, which represent the vectors basis for signal decomposition in the transformation, are obtained from a single prototype wavelet using dilatation, contractions and shifts.

3.8.1 Mathematical Model

Instead of using fix time (Δt) and frequency (Δf) resolution, we can let both to vary in the time-frequency plan in order to obtain a multi-resolution analysis. If we keep the ratio ($\frac{\Delta f}{f} = c$) constant, the time resolution will increase when frequency increases and frequency resolution will increase when frequency decreases.

We can compare the wavelet analysis with a filtering using a set of bank band-pass filters with constant relative bandwidth.

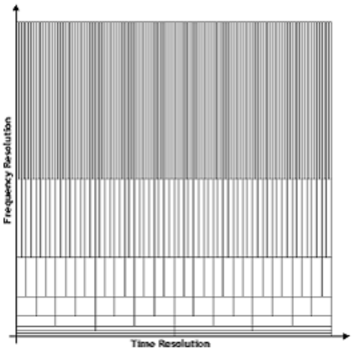


Figure 3.15: Time-frequency resolution plan

If we analyze the impulse response of the analysis filters in the filter bank, we see that they are all scaled version of the same prototype $\psi(t)$. For example, for a scale factor γ the filter response is:

$$\psi_\gamma = \frac{1}{\sqrt{|\gamma|}}\psi\left(\frac{t}{\gamma}\right) \tag{3.33}$$

Based on this, the definition of the wavelet transform is:

$$W(\tau(\gamma)) = \frac{1}{\sqrt{|\gamma|}}\int_{-\infty}^{+\infty} x(t)\bar{\psi}\left(\frac{t-\tau}{\gamma}\right)dt \tag{3.34}$$

where $\bar{\psi(*)}$ represents the complex conjugate operator of the variable ψ.

If we choose modulated windows like in eq 3.35 for the basic filter, we can see the connection with the Fourier transform

$$\psi(t) = \varphi(t)e^{2\pi f_0 t} \tag{3.35}$$

The definition of the local frequency depends on the primary wavelet and is no longer linked only to the actual frequency but is related also with the scaling, hence the terminology "scale" used in wavelet analysis instead of frequency.

The inverse wavelet transform is defined as:

$$x(t) = \frac{c}{\gamma^2}\int_{-\infty}^{+\infty}\int_{\gamma>0} W(\tau(\gamma))\psi_{\gamma,\tau}(t)d\gamma dt \tag{3.36}$$

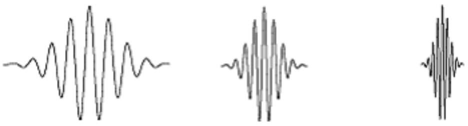


Figure 3.16: Three wavelets with different time durations and frequencies

if we define: $\psi_{\gamma,\tau} = \frac{1}{\sqrt{\gamma}}\psi\left(\frac{t-\tau}{\gamma}\right)$ as basis functions and we choose to make the scaling variable γ discrete, we do this in a logarithmic manner ($\gamma = \gamma^{-j_0}$) and we can use the Nyquist sampling rule to make variable τ discrete at any given scale. We can obtain the new wavelets using:

$$\psi_{j,k}(t) = \gamma_0^{j/2}\psi(\gamma_0^j t - kT) \qquad (3.37)$$

where j and k are integers.

The new definition for the discrete wavelet transform is:

$$W[k,j] = \int_{-\infty}^{+\infty} x(t)\psi_{j,k}(t)dt \qquad (3.38)$$

And the inverse formula for the discrete transform becomes:

$$x(t) = \sum_j \sum_k W[k,j]\psi_{j,k}(t) \qquad (3.39)$$

In practice, a common value for γ_0 is 2, which changes the formula for calculating the wavelets from the basic wavelet to:

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k) \qquad (3.40)$$

3.8.2 Wavelet Decomposition

In image processing, a very common use for wavelet transform is image decomposition using wavelets for filtering the image. The idea is to filter the image using low pass and high pass filters obtained from the wavelet filters on both vertical and horizontal directions.

A generic scheme is given in Figure 3.17 and starts with filtering the image vertically using low pass and high pass filters, continues with decimating with a factor of two the vertical directions, followed by another horizontal filtering using the same low pass and high pass filters. The final step is to decimate again with the same factor the image on horizontal direction.

We obtain this way four sub-bands each corresponding to a way of filtering noted with:

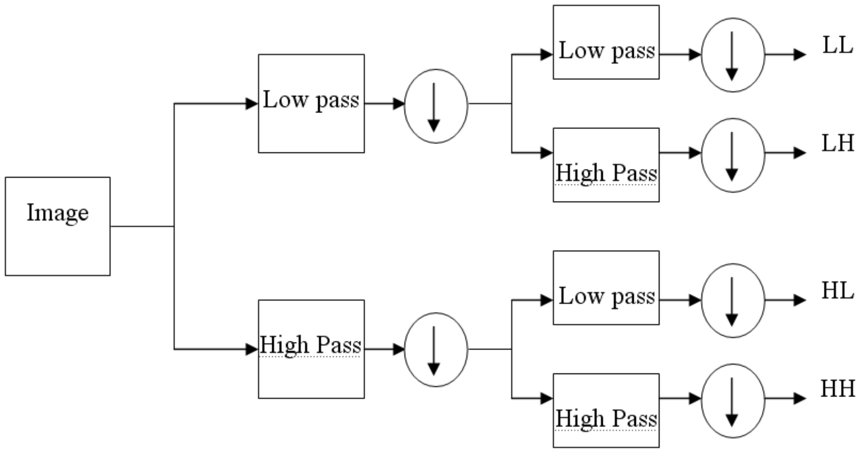


Figure 3.17: Block scheme for 1 level wavelet decomposition

- LL (Low pass filtering on vertical and horizontal directions)
- LH (Low pass filtering on vertical direction and high pass filtering on horizontal direction)
- HL (High pass filtering on vertical direction and low pass filtering on horizontal direction)
- HH (High pass filtering on vertical and horizontal directions).

This method is used for many applications in image processing like: image compression, image de-noising, image restoration and many others.

An example showing the application of wavelet decomposition to a facial image is shown in Figure 3.18 below. This example uses Daubechies D4 wavelets for decomposition.

The wavelet decomposition is used in our system for filtering the image prior to PCA analysis in order to cancel the effect of small differences in the face region like small rotations, face expressions or orientation changes of the face region.

3.9 Face Recognition Preliminary Tests

The first test we performed to analyze the accuracy of our face recognition approach was on the BioID database. This database contains 400 pictures of 20 different persons. The faces presented are frontal views with large variations in facial expression and illumination. The database is represented in Figure 3.19.

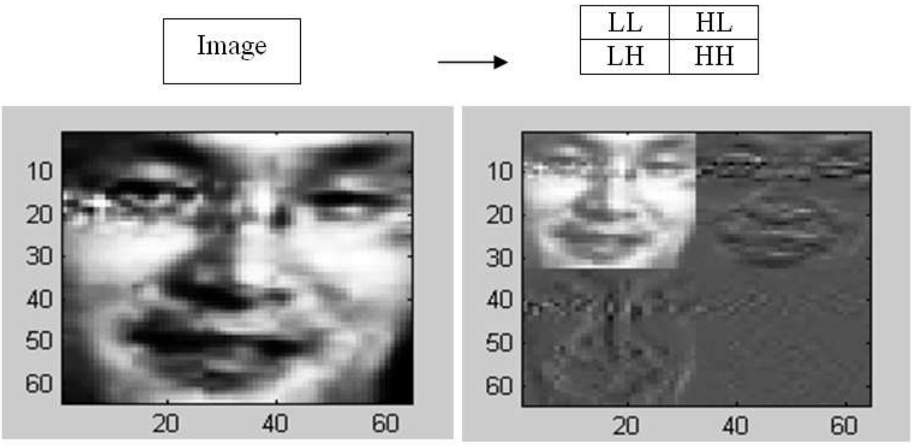


Figure 3.18: Example of wavelet decomposition applied on an image

The testing procedure was: use a different number of faces in the training stage (gallery) and the rest of them for testing (probe). Run the recognition test for gallery size from 1 to 10 images per person.

The training stage consists in: preprocessing of the face images (resize all faces to a standard size, transform them to grayscale images and apply histogram equalization), computing the PCA eigenvectors, or basis vectors of the gallery images and project both the gallery and the probe into these basis vectors after wavelet decomposition was applied on the images. Also DCT coefficients are extracted and stored for each face.

The method used for classification is the "nearest neighborhood" algorithm. This implies that the test face belongs to the cluster of the closest face in the feature space. If the closest face belongs to the same person as the test face we consider the result as a hit if not we consider it as a miss. In the end the recognition rate represents the number of correct hits divided by the total number of faces in the probe set.

The distance between faces is the Euclidean distance between the coefficient vectors (DCT and PCA). We wanted to compare the recognition rate when using DCT coefficients with using PCA coefficients and a combination between the distance using PCA and DCT. The combination algorithm is described in detail in **Chapter 4**.

For these tests the faces were normalized to 32x32 pixels size. In order to compute the distance between the DCT spectrum we used only the first 100 coefficients and the PCA coefficients used corresponded to the eigenfaces from 2 to 18 corresponding to the ranked eigenvalues. We reached this combination after a series of other tests and this configuration is optimal giving the best results (also considering the execution time). The faces were extracted using the face detector described in the previous chapter and no other manual method was used to align them.

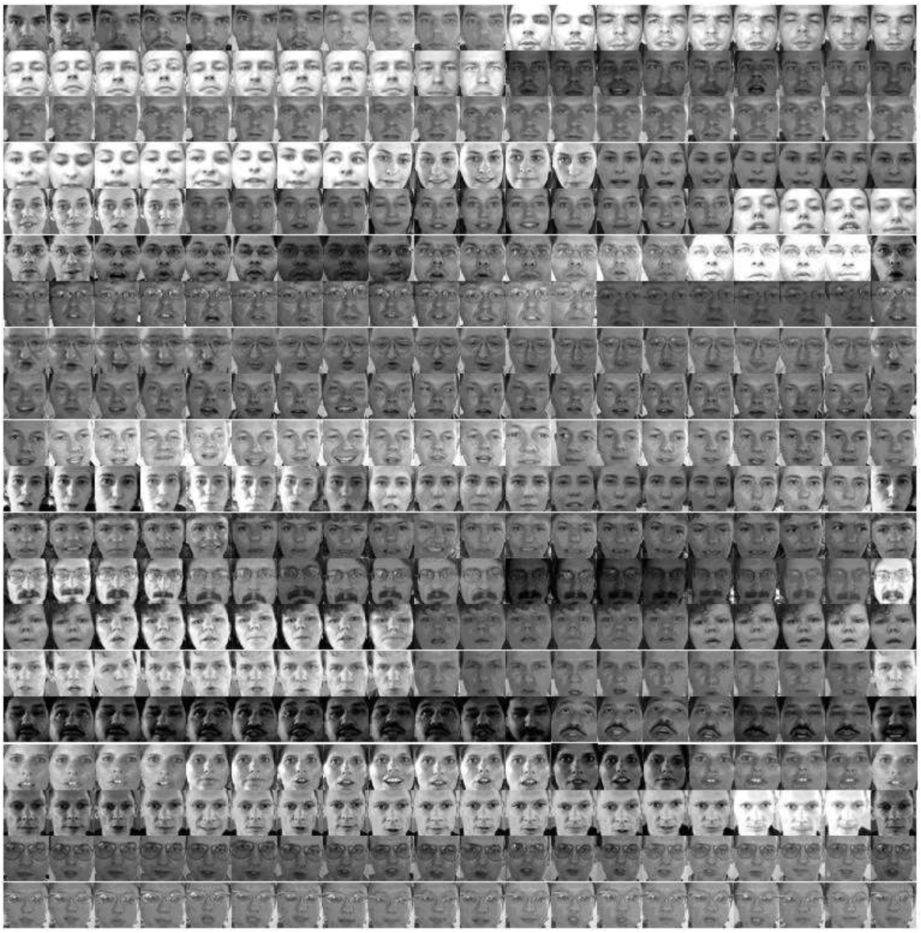


Figure 3.19: BioID database

The testing procedure used was nearest neighborhood approach based on the distances computed using PCA, DCT and their combination. The method for selecting the training and testing faces was based on their rank in the collection. When only one image was used for training, the first one was selected and all the other faces left were used for testing the algorithms. If the closest face after classification was the correct one than we count it as a match if not we count it as a miss. In the end the recognition rate is given by the number of correct matches.

The recognition rates obtained using only DCT and only PCA features along with the recognition rates using their combination are presented in next table:

Table 3.1: Recognition Rates for BioID database

DCT (%)	PCA(%)	Combination (%)	No faces train/test
50,51	47,9	59,7	1/19
70	71.6	74.7	2/18
76.7	77.6	82.35	3/17
79.68	83.44	83.43	4/16
84	86	87.6	5/15
84.64	87.14	88.21	6/14
86.54	88	90	7/13
90	92	91.67	8/12
95.45	92.27	94.55	9/11
98	94	97	10/10

A graphical representation of the evolution of the recognition rate against the number of faces used for training is given in Figure 3.20.

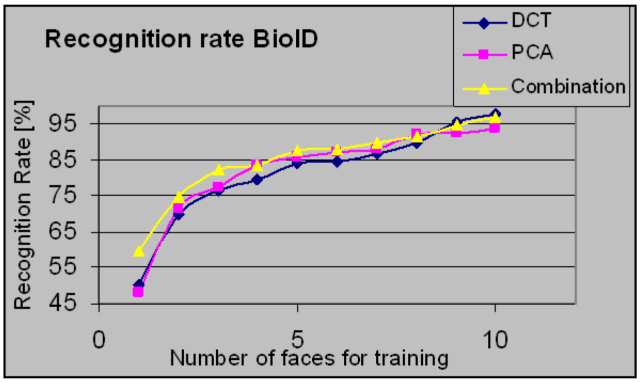


Figure 3.20: Results for BioID database

It can be noticed that the recognition rates are better for the combined approach especially when only a few images are available for training and as expected both algorithms and their combination perform better when using more instances of the same person in the gallery set for training.

The DCT approach seem to outperform the PCA approach but this was not always the case in our tests as it will be shown in the results dedicated chapter.

These recognition rates are comparable with the results from other reported face recognition approaches but it has more advantages for our type of application (speed, simplicity, robustness, training data etc).

For more information about the testing procedure and more rigorous experimental results on different databases please see **Chapter 6**.

Chapter

4

Face Periphery Analysis

This chapter describes the use of additional information in the classification process, information extracted from automatically defined regions surrounding a detected face region. This information is more robust to factors such as variations in pose and illumination that can unduly influence the accuracy of conventional face recognition algorithms.

The chapter begins with an overview of how such additional information can be extracted and used to assist in classifying people in an image. The advantages and disadvantages of using this information are discussed in section 4.1. In section 4.2 we provide detailed descriptions of algorithms that were tested for their ability to extract colour information from image regions: histogram techniques are described in section 4.2.1 and colour correlogram techniques in subsection 4.2.2 along with the description of a fast algorithm for computing the colour correlogram of an image in section 4.2.3. The latter was the technique we eventually adopted in our recognition system.

Section 4.3 shows how the peripheral regions can be automatically defined in order to extract the maximum amount of useful information. The dimensions and geometry of these regions were initially determined empirically from observations and tests on a number of moderately sized face databases. They provide useful information on the subjects hair and upper body clothing with a low risk of confounding data being captured from the image background. Subsequent testing on larger databases has proved their robustness and usefulness in practice. In the same time this information must describe only region that is part of the subject region, without any noise that may affect the classification. This noise can be caused by obstacles or parts of another person that are positioned in front of the subject when the picture is taken.

Finally section 4.4 of this chapter provides some conclusions on the benefits and potential drawbacks of using peripheral region information for the classification of people in image sets.

4.1 Overview of Using Additional Information for Person Classification

Although there are many face recognition systems reported in the literature with very high recognition rates, the problem of recognition robustness with respect to the acquisition conditions is still largely unsolved. Classical databases that are used to evaluate face recognition systems are built using constraints to the environment (e.g. lighting conditions, image background) where the pictures are captured and also to the subjects that are photographed (e.g. facial pose and orientation). These databases are useful for comparative testing of algorithms and there are benefits to using a database where the only image variability across face regions is, for example, based on luminance levels. Such a database allows different algorithms to be tested and compared for robustness to variations in luminance.

Nevertheless, when dealing with typical consumer image collections, we have to consider that we cannot make any assumption about the conditions under which the pictures were acquired. Thus it is very difficult to implement a face recognition system that will have good results is every condition. It is well known that face regions are sensitive to variations in illumination, so face recognition results can be strongly influenced by the lighting condition when a picture is captured.

Another factor that can influence the results of face recognition is the quality of the picture. It is a common thing for amateur photographers to take pictures with incorrect focus or aperture settings. Even with the supposedly "smart" scene algorithms in modern digital cameras it is very easy for these algorithms to be "tricked" by the ambient conditions. Indeed, as consumers implicitly trust the automated algorithms in a digital camera to "get it right" they often tend to ignore the conditions under which images are captured, shooting directly into the sun or under low lighting conditions where handshake is likely to blur the image. Other common problems include not allowing enough time for the autofocus algorithm to work; focussing away from the main subject; shooting a high contrast image with strong shadows/light; capturing images with strong specular reflections from metallic surfaces or windows, and so on. Thus many of the images in a consumer image collection can be of somewhat variable quality.

Also the size of the face could influence the result if the face is too small for the features that are extracted from it to represent its related characteristics.

Bearing all these factors in mind we can try to find other elements of the image, other than the face that can carry useful information about a person. These elements should be able to separate that person from others, and provide information that is potentially more robust to the various factors enumerated above.

We can begin with the observation that when taking a set of pictures during the same social event or during the same day people will usually wear the same clothing and jewelery. People will also tend to "get their hair done" for special events - and it is at such events that we capture many of our images. We also remark that when on vacation most people have favorites shirts, dresses, or outfits; younger people have their "clubbing gear" and every woman remembers her wedding dress. In general, we are more likely to take pictures of ourselves when we are "dressed up".

This suggests that it would make sense to implement a system that will look not just for similarities between faces but also similarities between people's clothing, /jewellery and hairstyles when searching and categorizing images. The advantages of such a system will be: (i) usually the colour of clothing is significantly less sensitive to variations in luminance; (ii) colour is not influenced by the quality of the picture (focus, motion blur), and (iii) the region which represents the clothes is larger than the face region and can provide useful information even when the face region is too small to be properly distinguished.

A further advantage of using such information is that it provides a useful means of categorizing people according to different events or periods of time thus supplementing and improving on the categorization of people based on their face region alone. In brief, we can use face information to sort the people; peripheral regions information, in addition to supplementing our base sorting of images based on the people in them, also allows us to further categorize based on particular times or events.

There are some drawbacks to using such descriptors. In some cases the region that contains the clothes of the person may not be entirely present in an image. This can happen for two reasons: (i) the person is located at the margin of the picture so most of the region will be located outside the image and (ii) some obstacle could occlude the region, so instead of finding features that describe the clothes, we will also have some features that describe the obstacle, which can produce errors.

The first drawback can be overcome by restricting the algorithm to use only face features when the face is at the margins of the picture. The second case is more problematic, but it can also affect the detection of face regions where one person's face partly occludes another's. In fact the problem of detecting and handling partial occlusions in object detection and object tracking systems could form the basis for an entire thesis in its own right, thus we will not discuss it further here.

For both drawbacks we have to make a compromise regarding the size of the region that contains the clothes. Larger regions mean more information to be compared, but is then more likely that the region will lie outside of the image or that an obstacle may partly occlude it. Smaller regions imply less information is captured and may affect the granularity of classification that can be achieved.

The second region than contains useful information about a person is the region above the face which gives details about the person's hairstyle and hair colour. This region has similar advantages to the clothing region but is far less likely to lie outside the main image or to suffer from significant occlusion. The only drawback is that this region contains less useful information than the face or the clothes regions and is generally not as useful to distinguish between people.

From this point forward we refer to the clothing region as the *body region* and to region above the face as the *hair region.* An example with these regions displayed over an image is given in the next figure

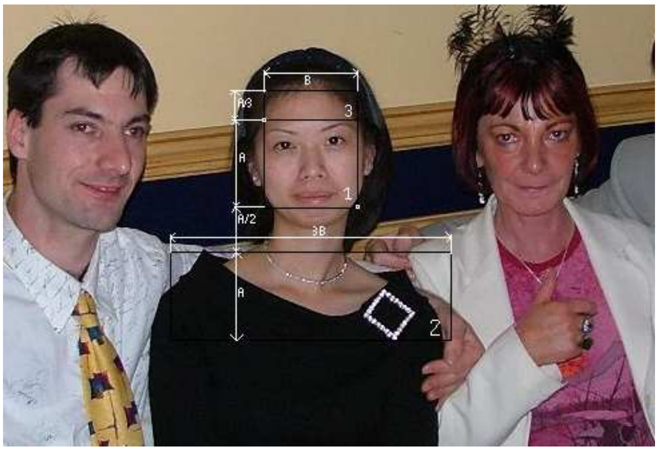


Figure 4.1: Body and Hair region

A similar method of defining peripheral regions for similarities between persons was used in [119] to associate names to the persons in family albums in order to query the albums for persons using their names. They are using as body region a much larger portion from the image that includes the face region.

4.2 Literature Review on Extracting Descriptors

There are many methods to extract information from these peripheral regions that can be used for similarity search. Three main classes of descriptors can be defined: (i) colour based descriptors (ii) shape based descriptors and (iii) texture based descriptors. For a very useful and comprehensive review of image descriptors that can be used for similarity searching please see [41].

Colour based descriptors are widely used in image retrieval systems. Usually these features describe the colour distributions in the image in the form of the image colour histogram. Other colour based descriptors are the colour constancy, which represents the capability of perceiving the same colour in different illumination conditions, and colour ratios as relations between neighboring pixels.

A very important aspect of these descriptors is the colour system used to compute them. For instance the RGB colour space is very commonly used to describe images. When large intensity variations are present in the image, the normalized RGB space [104] has superior retrieval capabilities. This normalized representation is invariant to changes in illumination and object geometry.

The Lab colour space [23] is derived from models of human vision. It was designed to provide relative perceptual uniformity which means that colours which, in terms of human perception, are close to each other will be close also in the sense of the distance between their representations in this colour space.

The HSV colour space is know for its invariant properties, where each component is assigned to a different property of the pixel. In [99] is reported that transformation and quantization of the RGB space into the HSV colour space satisfy the properties of uniformity, completeness, compactness, and naturalness which can prove efficiently in some applications.

Shape based descriptors are usually based on detecting edges in the image. There are many edge detectors reported in the literature like the Canny detector [20] or the technique of automatic gradient thresholding [42].

Texture descriptors are also widely used in image retrieval systems. Texture information represents any information from the image that is not related to the colour or the shape information. A very common technique for quantifying the texture of an image region is based on the wavelet transform as described in [64] and [100]. But the wavelet transform also gives information about the local shape of the image so this is a combined method of classification.

Other texture descriptors are based on fractal analysis in [58] and on gradient analysis in [75].

Combined approaches such as wavelet based techniques are biologically inspired and are used where the image components cannot be accurately classified using only colour, shape or texture information. Only classifiers which provide a combinational analysis of these qualities of an image region can achieve good results and accurately distinguish between subtle regional variations.

In the next section the colour based descriptors and a combined colour and texture method known as the colour correlogram are described in detail.

4.3 Colour Based Features

The most common initial approach in any image processing system is to try to use the colour information of the image.

The advantage of colour based approaches compared to shape or textured approaches lies in their relative simplicity. Their principle disadvantage, especially in classification systems, is that colours are influenced by the luminance, which makes the same object under different illuminations to appear as different objects.

4.3.1 Histogram Techniques

Given an image with L colours, the histogram is a discrete function $h(k) = n_k$ where k is a colour from $1 \ldots L$ and n_k represents the number of pixels in the image of colour k. In practice, the normalized histogram is used:

$$h_n(k) = \frac{n_k}{N}$$

where N represents the total number of pixels in the image. The values of the normalized histogram range between 0 and 1 and the sum,

$$\sum_{k=1}^{N} h_n(k)$$

is equal to 1.

Those properties confirm the fact that the normalized histogram is a probability distribution that gives the probability of a pixel to be of a certain colour.

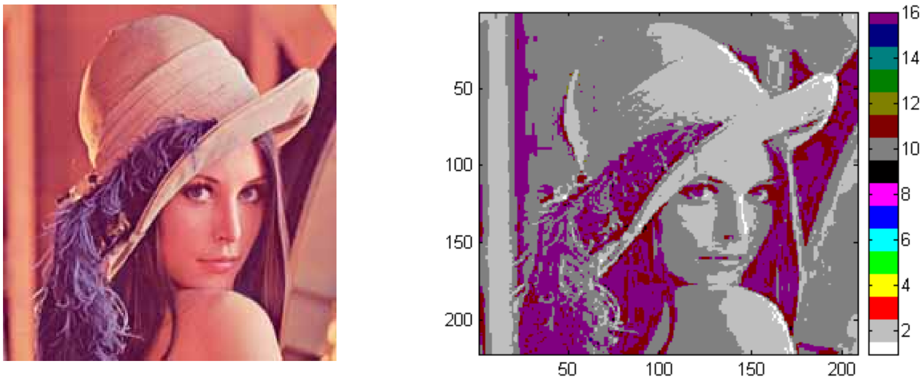
An important aspect in calculating the histogram of an image is the colour space used. The simplest way is to use the greyscale colormap but a lot of information is lost when passing from colour images to grey images. There are many colour spaces available and therefore there are lots of opinions about which colour space is optimal for a particular application [24].

The Red-Green-Blue (RGB) colour space is the most common one because the image pixels are usually stored as RGB triplets. But the RGB space is not perceptually uniform so usually this space is transformed into another colour space. Simple transformation includes bits manipulation of the RGB values as in [37] and [37]. Another transformation is described in [46], where RGB is transformed into the quantized CIE-LUV space with 512 colours.

In this context we remark that many such transformations are "one-way" and it will not be possible to recover the original image at the same level of quality. On the other hand, certain computations can be achieved both more accurately and more efficiently if the right colour space is chosen.

Given that there are benefits to performing certain analysis in different colour spaces, the question we must now pose is: having an image in our database which uses colour space C_1 how do we convert it to another color space C_2?

The simplest transformation is to calculate for each pixel in the image described by the colours in C1 the closest colour in C_2 by evaluating the Euclidean distance between colour triplets. When C_2 has fewer colours than C_1, this method will give a maximum error between the two images. An example of the resulting image using this method to reduce the number of colours from RGB (232 colours) to a 16 colour VGA colormap is given in Figure 4.2.

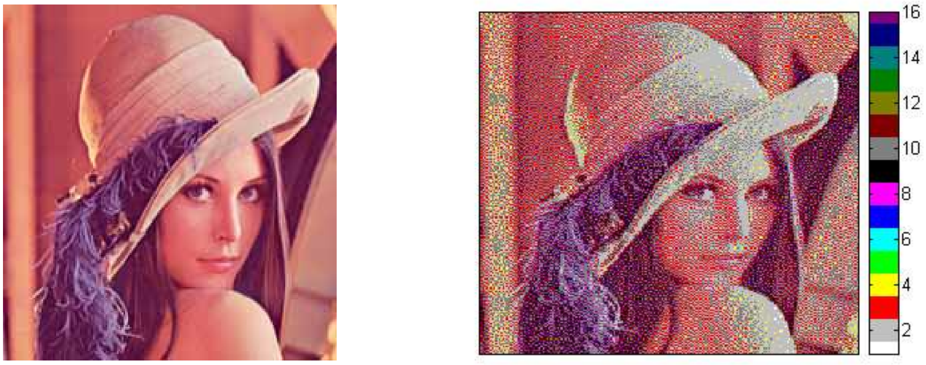


(a) Original Image

(b) Transformed Image

Figure 4.2: RGB to VGA colour transformation

Another method for reducing the number of colours or transforming from one color space to another is a dithering technique which will distribute the error (the distance from the new value of the pixel from the new colormap to the old value) amongst the neighbouring pixels. A classical method is the Floyd-Steinberg [38] error diffusion dithering algorithm where the error for pixel (x, y) is diffused to the neighbouring pixels $(x + 1, y)$, $(x - 1, y + 1)$, $(x, y + 1)$, and $(x + 1, y + 1)$, with weights 7/16, 3/16, 5/16, and 1/16, respectively. Figure 4.3 presents the results when using the Floyd-Steinberg dithering method.



(a) Original Image

(b) Transformed Image

Figure 4.3: RGB to VGA colour transformation using dither

A well-known application of histogram is using the histogram equalization as an image enhancement technique. The idea is that if the colours of an image are distributed across the entire colour domain, the contrast should be optimal. In Figure 4.4 we show an image with 3 different histograms to illustrate how the quality of a grey scale image can depend strongly on the histogram distribution.

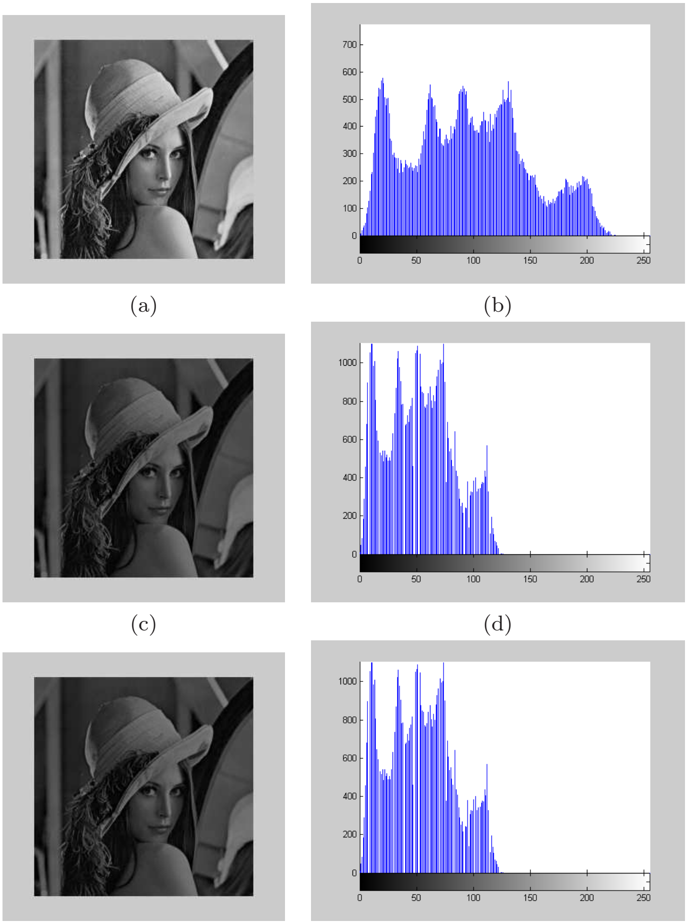
The first system that used histogram for image retrieval was reported by Swain and Ballard in [104], after that the colour histogram became an important tool for characterizing the content of an image.

The advantages of histograms for classification are the simplicity and speed of the algorithm, and the robustness it exhibits to small changes in the camera view. The disadvantage is that it does not matter how the colours are distributed spatially. So different kind of pictures with similar colours may be reported as having very similar histograms, whereas the same objects with slightly different colours due to different illuminations could be reported to have no similarity.

Other techniques were subsequently developed which overcame some of these disadvantages by also taking into consideration the spatial distribution of colours in an image.

4.3.2 Colour Correlogram

The techniques that employ information about the spatial distribution of colour in an image can be divided into two principle classes: (i) image partitioning methods, divide the images into sub-regions and compare histograms on a regional basis using position constrains for the entire image; (ii) histogram refinement methods, involve the addition of local spatial information to the histogram.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4.4: Images and associate histograms

As an example of the first of these classes, the authors of [102], divided an image into five fixed overlapping windows, and for each window the first three moments are calculated. Each picture will be represented by a vector formed by the moments calculated for each of the five regions. The storage requirement is small, and the technique performs better than the full image histogram and is insensitive to small rotations due to the overlapping windows. Another example of an image partitioning technique is described in [98]. Here the image is divided in regions based on binary colour sets calculated using histogram back-projection.

A well known technique for histogram refinement is described in [98] and uses the notion of a coherent pixel. A pixel is coherent if is a part of a large group of pixels with the same colour, otherwise it is not coherent. The colour coherence vector (CCV) [83] represents the classification of each pixel in the image. An enhanced version of CCV is CCV with successive refinement which uses additional special features. The advantage of CCV is that is fast to compute and it performs better than histogram.

The colour correlogram is a method between the two classes taking into consideration both local properties of the colour distribution and global properties of the colours in the image. By doing that a balance is realized between false positives and robustness to small changes in the image.

Let I be an image with dimension n x m. The colours in image I are quantized in L colours. A pixel in the image is noted $p(i, j)$.

The histogram of image I is:

$$h_I(k) = countp(i, j) = C_k | 1 < i < n, 1 < j < m \tag{4.1}$$

where $1 < k < L$ and C_k is the k_{th} color in the colormap.

In order to define the correlogram, we have to define a distance measure in 2-D space. The L_∞ norm is used: for two pixels in the image $p_1(i_1, j_1)$ and $p_2(i_2, j_2)$ the distance between them is:

$$D_{L_\infty}(p_1, p_2) = |p_1 - p_2| = max|i_1 - i_2|, |j_1 - j_2| \tag{4.2}$$

Definition. The colour correlogram represents the probability for a given pixel of colour C_k to be at a distance d of another pixel of colour C_q:

$$Corr_I(k, q, d) = P\{p_b(i, j) = C_q | p_a(i, j) = C_k, |p_a - p_b| = d, 1 < i < n, 1 < j < m\} \tag{4.3}$$

where $1 < k, q > L$

The size of the colour correlogram is equal to $L^2 * d_{max}$.

The colour autocorrelogram represents the probability for a given pixel of colour C_k to be at a distance d of another pixel of the same colour C_k :

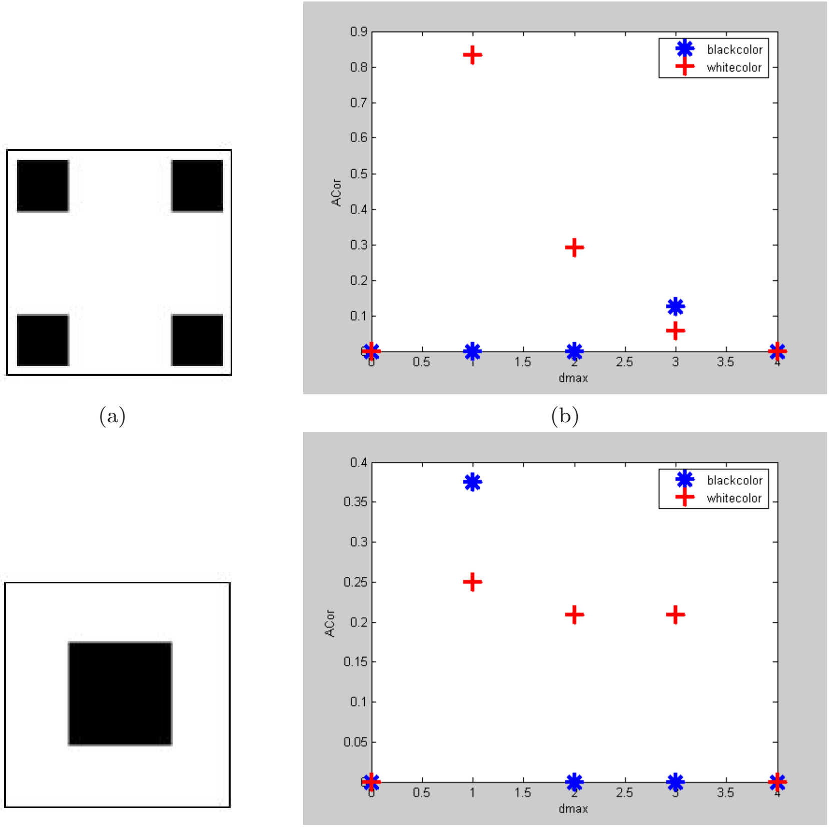
$$ACorr_I(k,d) = P\{p_b(i,j) = C_k | p_a(i,j) = C_k, |p_a - p_b| = d, 1 < i < n, 1 < j < m\} \tag{4.4}$$

The relation between correlogram and autocorrelogram is of course:

$$ACorr_I(k,d) = Corr_I(k,k,d) \tag{4.5}$$

The dimension of the autocorrelogram is equal to $L * d_{max}$.

An example of two simple images and their colour autocorrelograms is given in the next figure. It is obvious that the histograms for the two pictures are identical therefore you can not distinguish between them using histogram method.



(a)

(b)

(c)

(d)

Figure 4.5: Images and associate histograms

Another image descriptor derived from the colour auto-correlogram is Banded colour auto-correlogram introduced in [52] by Huang et al., and defined by the formula:

$$BACorr_I(k) = \sum_{d<d_min} ACorr_I(k, d) \tag{4.6}$$

The main advantage of the banded correlogram is that it has the same dimension as the histogram, i.e. it is a vector with the number of elements equal to the number of colours in the image) but it also incorporates spatial information. The results reported in the literature prove that the banded correlogram descriptors give better results than the histogram descriptors [52]. In the next figure there are represented several pictures with their histograms and banded correlogram obtained using a small colormap with 16 colours. This will give the reader a practical feel for the distinction between the two methods.

An important aspect of the algorithm is its speed. In the next section a technique originally described in [52] is discussed in some detail. This algorithm is very effective at speeding up a practical implementation of the algorithm when compared with the standard, and more intuitive algorithm.

4.3.3 Fast Algorithm for Calculating the Colour Correlogram

A good starting point for calculating the correlogram is to first compute the following component:

$$\Gamma_I(q, k, d) = count\{p_b(i, j) = C_q | p_a(i, j) = C_k, |p_a - p_b| = d, 1 < i < n, 1 < j < m\} \tag{4.7}$$

which represents the total number of pixels of colour C_j at distance d from any pixel of colour C_k.

The correlogram can be computed as:

$$Corr_I(q, k, d) = \frac{\Gamma 1 < j < m}{h_I(k) * 8d} \tag{4.8}$$

the factor $8d$ is due to the properties of L_∞ norm.

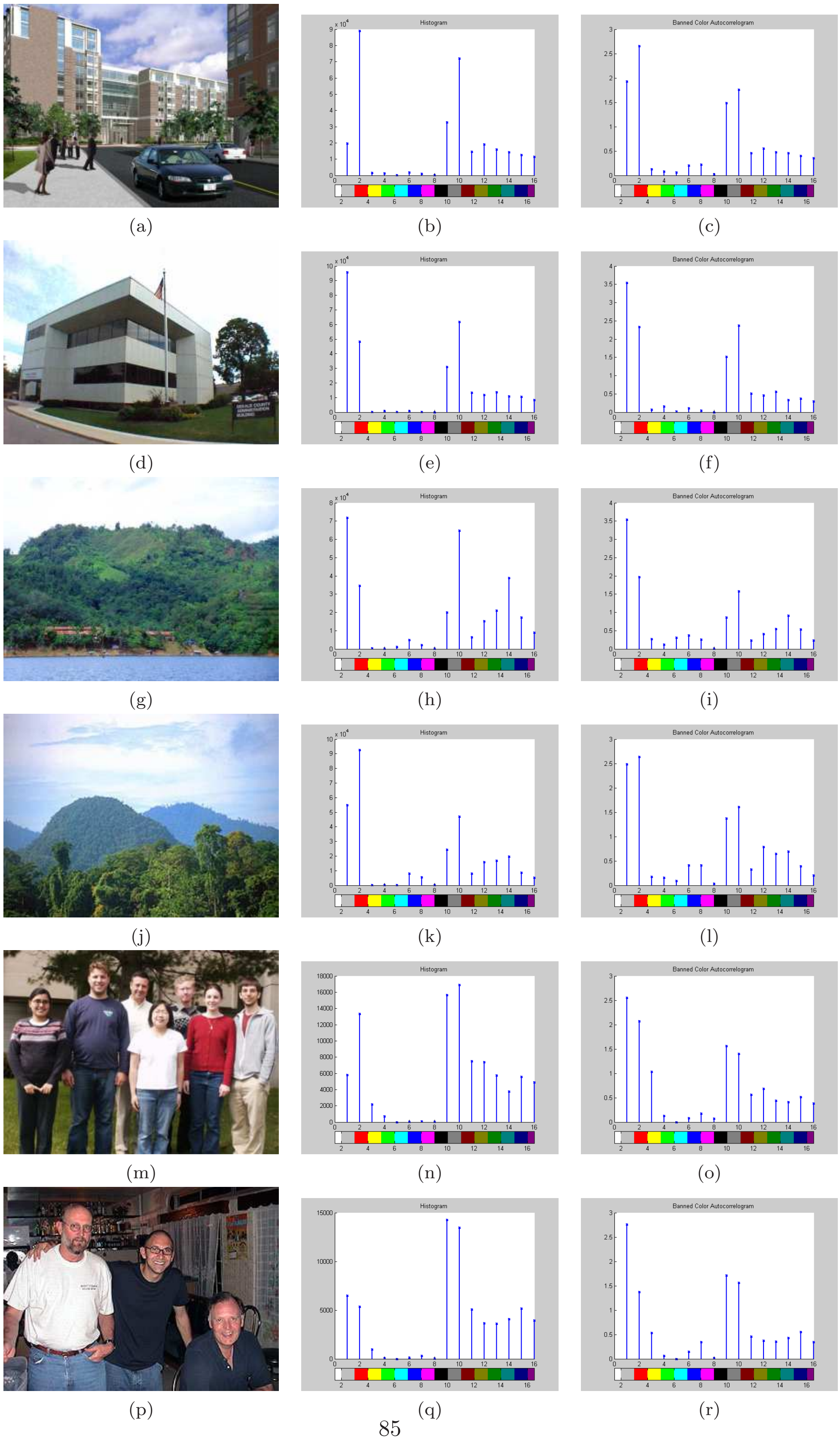
The intuitive algorithm will be for each pixel of colour C_k in the image to count at each distance d how many pixels of colour C_q are. This requires $O(nmd^2)$time.

Two quantities are defined:

$$\lambda^h_{i,j}(k, d) = count\{p(i + x, j) = C_k | 0 \leq x \leq d\} \tag{4.9}$$

$$\lambda^v_{i,j}(k, d) = count\{p(i, j + y) = C_k | 0 \leq y \leq d\} \tag{4.10}$$

which represent for each pixel (i, j) the number of pixels of a given colour at



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

(p) (q) (r)

Figure 4.6: Histograms and Banded Colour Autocorrelogram for different images

distance d in the positive horizontal direction (eq. 4.9) and positive vertical direction (eq. 4.10).

The two equations (4.9) and (4.10) can be computed incrementally using the next formula:

$$\lambda^h_{i,j}(k,d) = \lambda^h_{i,j}(k,d-1) + \lambda^h_{i+d,j}(k,0) \tag{4.11}$$

with the initial condition:

$$\lambda^h_{i,j}(k,0) = \begin{cases} 1 & ifp(i,j) = C_k \\ 0 & otherwise \end{cases} \tag{4.12}$$

Equation (4.11) is calculated for all the pixels in the image and for all distances from 1 to d_{max}.

Similar we can calculate for vertical direction:

$$\lambda^v_{i,j}(k,d) = \lambda^v_{i,j}(k,d-1) + \lambda^v_{i,j+d}(k,0) \tag{4.13}$$

Ignoring the margins, we can calculate (4.6) using:

$$\begin{aligned} \Gamma_I(q,k,d) = \sum_{p(i,j)=C_i} \left(\lambda^h_{i-d,j+d}(q,2d) + \lambda^h_{i-d,j-d}(q,2d) + \right) \\ + \sum_{p(i,j)=C_i} \left(\lambda^v_{i-d,j-d+1}(q,2d-2) + \lambda^h_{i+d,j-d+1}(q,2d-2)\right) \end{aligned} \tag{4.14}$$

This algorithm based on dynamic programming takes $O(nmd)$ time and it is very efficient when d is small.

4.4 Localized Description of Peripheral Regions

As discussed in section 4.1, we can define peripheral regions of the face containing useful information that can be used in the retrieval (classification) stage. We defined the body region as the region from the person which gives information about the person's clothes, information which is more robust to different confounding factors than the face is. The second region is the hair region which gives information about people's hair style and hair colour. We mentioned the compromise that need to be done concerning the size of the regions. Larger regions can provide more information but increase the likelihood that the region may be out of the picture or occluded by objects in front of them. In both of these cases we lose the benefits of these additional regions and the recognition process could be negatively affected if our algorithm does not detect these cases.

After a series of manual observations of a diverse set of images, we reached empirical definitions for the body and hair regions using the face region (701) as illustrated in figure 4.6. The body region (703) has the same height as the face and three times the width and is situated below the face starting at half of the face height distance from the lower limit of the face. This captures the most important information about the clothing colour, patterns texture, and neckline, as well as any body jewelery. Further, as the region does not extend beyond the shoulders it is less likely to introduce confounding information from the background regions, or to extend beyond the outer perimeter of the main image.

The hair region (702) is situated just above the face region and has the same width as the face and the height is equal to a third part of the height of the face.

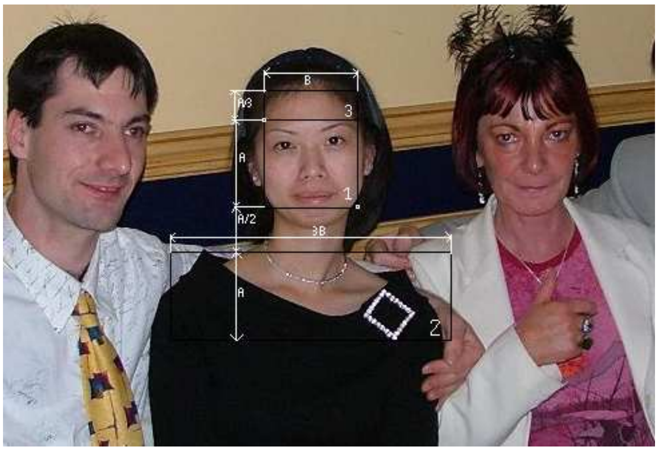


Figure 4.7: Body and Hair region

After we defined and localized these peripheral regions we extracted features that can be used in the classification stage. We decided to use features that describe the colour and texture distribution in these regions because this can define person characteristics when comparing with others.

We have to decide first which features will give better results in the retrieval process: histogram, autocorrelogram or banded autocorrelogram features. The second task is to determine the optimal parameters to use in the feature extraction method: number of colours (colormap) for histogram features and maximum distance and number of colours (colormap) for the autocorrelogram and banded autocorrelogram features.

An extensive series of tests were performed in order to extract the optimal configuration for feature extraction and also to see what is the best combination of regions for optimal results. The results and conclusions of the tests are presented shortly in **Chapter 6**.

4.5 Conclusions

Using histograms to compare images is a very simple and fast technique which has been used widely in content based image retrieval systems, often with good results. The advantages are (i) the simplicity of the algorithm, (ii) the speed, (iii) the fact that the size of the features vector set is small and depends only on the number of colours and not on the image dimensions and (iv) the robustness of the method for small changes of the view point. The disadvantage is the fact that histogram characterize the global colour distribution in the image so images with different contents may have identical histograms.

New algorithms were developed to take into consideration local distribution of colours within the image. Colour correlogram is one of these algorithms and can give a good approximation of the global distribution of colours in the image, in addition to determining local colour structures. This provides a good balance between the occurrence of false similarities in a retrieval system and the robustness of the algorithm with respect to changes of the view point in an image. The dimension of the colour correlogram vector is equal to the square of the number of colours multiplied by the maximum distance. The colour autocorrelogram yields an alternative feature vector with a smaller dimension. Also using the fast algorithm for calculating the correlogram, the time needed for feature extraction in the retrieval system is kept low - an aspect of this technique which is critical when matches must be made across large image sets.

In order to improve our person retrieval system we decided to complement the extracted face recognition information with additional information extracted from peripheral regions defined by the detected face region. Thus the retrieval system will be more robust to changes of the image acquisition conditions such as illumination, face pose and size, image focus and quality.

In our final application the use of additional information from the peripheral regions can be selected or deselected by the user. When this extra information is not used the grouping of the images is based only on the face similarities. The method of combining feature sets is further described in **Chapter 5**.

By selecting to use the extra information the user will be able to group more accurately the pictures taken during the same event or occasion. This is closer to the intuitive method of browsing photo collections used by people who sort through collections of printed images.

The peripheral regions were defined after a series of observations and their size reflects a balanced equilibrium between the usefulness of the information extracted and the probability for the region to appear correctly in the image.

Chapter

5 Combining Multiple Classifiers - Multimodal Systems

Now we have outlined the core elements of our *Person Recognizer*, via the *Face Detector*, the *Face Recognizers* and the *Peripheral Region Analyzer*, it is time to consider how we can link together these components to achieve an improved recognition engine. In this chapter we present some multimodal systems approaches as potential solutions for increasing the robustness of our recognition system.

First Section 5.1 presents the rational of using the multimodality approach in out system and its advantages.

Section 5.2 gives a general overview of the multimodal systems and their advantages in pattern recognition. A classification of multimodal systems is presented as well, starting with feature fusion systems in section 5.2.1 and decisions fusion systems in the next section 5.2.2.

Next, section 5.3 describes how we can implement multimodality in a person recognition system.

Section 5.4 provides details of the actual implementation that was chosen to enable multimodal similarity measures in our Person Recognizer system. In particular we present the statistical approach used to analyze and combine results from multiple classifiers across an image collection. We remark that this basic technique scales across groups of images, allowing us to build super-collections from multiple, smaller collections.

5.1 Using the Multimodality in the Person Recognizer

When designing any pattern recognition system we have the option to choose between different types of algorithms for analyzing and classifying the given pattern. The classical approach is to use a single algorithm for pattern analysis and feature extraction and a single algorithm in the classification stage. As every algorithm has its own particular advantages and disadvantages, we remark that the overall system will be affected by the combined disadvantages of the pattern recognition and pattern classification stages.

This statement is valid for any research domain where digital signal processing techniques are applied, but it has particular implications in biometrics where accuracy and robustness are essential due to the variability of the underlying biometric patterns being analyzed.

Different methods for pattern analysis extract features that contain different information from the patterns. Using a single category of features in the classification stage will make the system more vulnerable to confounding factors that affect that particular type of feature. For instance when trying to recognize an object in an image we can choose to use information about its color, its shape or its texture or for a better accuracy a combination between them. When using multiple types of features that contain different information from the pattern, it is unlikely that all features will be affected equally by the same factor, so the classification system should be more robust. This is particularly the case if we choose pattern analysis techniques that are complimentary.

We consider, as examples of analysis techniques that extract features representing different information from the field of speech recognition: (i) cepstral analysis, which describes the frequency distribution of the signal, and (ii) linear prediction analysis, which tries to get rid off redundant information in the speech signal by computing the optimal parameters of the filter used to compute the current speech sample as a linear combination of previous samples. Multimodal classification techniques involving combinations of these features and others, such as speech energy, are already used widely in speech recognition systems. Our goal here is to investigate some equivalent approaches for the field of face recognition.

Some relevant examples of different image processing techniques are: (i) spectrum analysis approaches, such as the Fourier transforms (FFT) and discrete cosine transform (DCT), which describe the 2D frequency distributions in images; (ii) principal component analysis (PCA) approaches, used for dimensionality reduction for the collections of patterns, and based on describing the patterns (images) as a linear combination of the eigenvectors computed from the covariance matrix of the collection. Inspired by equivalent research in speech processing and recognition systems [67], we decided to investigate the potential for combining these techniques as component parts of a multimodal *Person Recognizer*. Other descriptors such as colour distribution (histogram, colour correlogram) or texture may also be combined in order to obtain a further degree of robustness in our image classification system.

5.2 Overview of Multimodal Systems

We now recall the basic approaches of pattern recognition: the first step is to analyze the pattern in order to extract only the important information useful for classification and to ignore the redundant information. The extraction returns a complex multi-dimensional feature vector (for instance the spectral or prediction components in the case of speech/audio processing and the frequency, color, shape or texture components in the case of image analysis); the second step is pattern classification, which typically simplifies these complex patterns by determining an overall distance between extracted features.

Simple classification algorithms involve computing distances (L1 norm, L2 norm-Euclidean, Mahalanobis etc) between feature vectors or groups of feature vectors obtained from analyzing the patterns. These have the advantage of being simple and straightforward to implement but they cannot incorporate the variations between multiple samples of the same pattern when these are available for training (except Mahalanobis which takes into consideration also the variance between samples from the same pattern). These distances between to high dimensional feature vectors only give us an idea of how close/far away are the two distinct vectors in the high dimensional space.

On the other hand, more complex classification algorithms, for example Artificial Neural Networks (ANN)[49] and similar statistical methods, which are more complex to implement and consume more computational time and resources, have the advantage that they build models for each pattern to be classified by using several variations on the same pattern. They also allow for these models to be retrained by incorporating new samples of the same pattern when these become available. Support Vector Machines (SVM) [111] represent another classification technique with very good generalisation capabilities from a small number of data samples of the same pattern. All these algorithms require, for optimal classification, that the feature vectors obtained from the pattern to have the same length. This can be a disadvantage in certain applications, especially in speech recognition where feature vector length can be quite variable. Note that this disadvantage disappears when using Hidden Markov Models (HMM) [90] for classification, because they can cope with different lengths of the same pattern, but they are more difficult to train and also could be more influenced by other variations.

Inspired by the way that humans tend to classify complex patterns, such as human faces - by looking more closely at the face if they are uncertain; or speech - by listening more carefully or trying to read the lips, multimodal systems were developed. These systems seek to combine the advantages of multiple methods of feature extraction and classification to arrive at a single decision with an improved level of confidence [54].

A multimodal system can be defined as a pattern recognition system which uses more than one type of features to describe the pattern and/or more than one algorithm to classify the extracted pattern features. A generalized architecture of such a multimodal recognition system is presented in Figure 5.1 below.

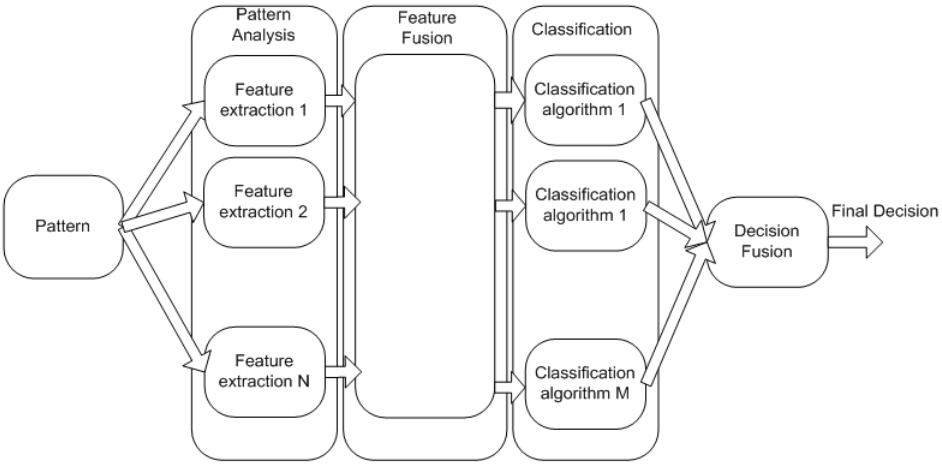


Figure 5.1: Multimodal system general architecture

It can be observed from the general architecture that there are two principle approaches to constructing a multimodal system [22].

The first approach involves using more than one category of features to describe the pattern to be classified and combines these heterogeneous features before presenting them to the input of the classification system. In this case there is a fusion of the features, prior to classification, but a single decision is determined from this combined set of classifiers. This category of system is termed a feature fusion system because such systems implement combination of the feature vectors presenting a single feature vector to the classification algorithm.

The second approach, may either apply a single classification algorithm to multiple heterogeneous features sets, or alternatively apply multiple classification algorithms to the same feature set. It is distinguished from the first approach in that there are multiple classification outcomes, each generating an independent evaluation of the extracted feature set, and these must be combined at the decision stage to arrive at a single outcome. This second approach is known as decision fusion because multiple decisions are combined in order to arrive at a final decision on the classification of an input biometric pattern.

We remark that it is also possible to combine the two categories into a hybrid system where heterogeneous feature sets are combined in one or more ways and these combined feature sets are submitted to more than one classification algorithms.

5.2.1 Feature Fusion Type Multimodal Systems

This type of systems uses a combination of different algorithms for analyzing the detected patterns. Note that the algorithms may analyze the same pattern or different aspects of that pattern in order to characterize it more accurately. As an example of multimodal system drawn from the field of speech recognition, and based on feature fusion, consider a system designed to recognize spoken words [107]. In order to recognize spoken words, this system analyzes both the speech signal (as in any classical system) and the image sequence of the speaker when pronouncing the words. The advantage of this approach as demonstrated in [107] is an increasing robustness of the system to the presence of background noise over the speech signal.

A general architecture for the feature fusion category of multimodal systems is given in Figure 5.2. After pattern analysis the feature vectors are combined in order to create multimodal models of the patterns which form the inputs to the classification stage.

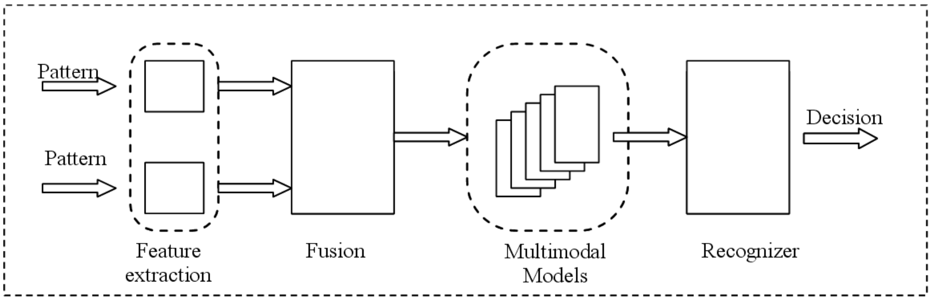


Figure 5.2: Feature fusion multimodal system

The most common approach for combining the feature vectors into a "super-vector", which will be presented as an input to the classification system, is that of simple feature vector concatenation [40].

Other methods involve statistical analysis of the feature vectors and use their properties in the fusion stage.

The advantage of this type of systems is that they are simpler to implement. In addition, when extracting different aspects for analysis, from the same original pattern the drawbacks of one particular aspect or type of features associated with the original pattern will be minimized by using the other features, given that the information contained by these different features/pattern aspects are substantially independent of each other.

The main disadvantages are the additional sensor hardware, computational resources and time needed to gather and process this extra information when comparing with an unimodal system. A further disadvantage of this type of system is that in order to combine the feature vectors, the multiple pattern aspects or multiple types of features may need to be spatially or temporally synchronized in order to represent the same part of the pattern and avoid complete confusion of the classification algorithm [40].

5.2.2 Decision Fusion Type Multimodal Systems

Decision fusion systems generate more than one type of classification output. These outputs are derived by either (i) applying multiple classification algorithms to the same feature set, (ii) applying the same classification algorithm more than once to different feature set extracted form the main pattern, (iii) applying multiple classification algorithms across heterogeneous feature sets, or (iv) a combination of all three approaches. We remark that decision fusion systems are a superset of feature fusion systems. In other words a feature fusion system may provide one, or more of the classification inputs to a decision fusion system.

After the classification stage the key aspect of a decision fusion system is that it will generate at least two independent classification outputs for the main input pattern. The challenge is to find some means of sensibly combining these classification outputs so that the resulting classification decision will have a greater success rate than either classification output used on its own.

This approach can prove very useful in improving the success rate of complex pattern analysis system. It was adopted for application to biometric based authentication/classification [93], a field where a very high accuracy is desired and where more than one type of biometric signal is available. Each of the biometric signals is analyzed and classified independently and an algorithm for obtaining the final decision is used. This algorithm is the simplest ”decision fusion” approach and it is based on Boolean verification of multiple biometrics in order to authorize a person as trusted.

By using such a system it will be more difficult for an impostor to foll the system because at least one of the biometrics will detect a false intruder. Now there are some researchers who opine that using more than one biometrics may not always be helpful, see [10] for example. Nevertheless it is sensible to assume that in most cases it will lead to some improvement as practically all the evidence in the literature demonstrates and proves this through successful experimental outcomes. A key goal of this thesis must be to demonstrate that such a combined approach is justified for our Person Recognizer system.

A general architecture of the decision fusion based pattern recognition system is given in Figure 5.3. After the pattern signal is analyzed and the features are extracted we can build different models using these features. Using the separate models, decisions are taken using one or more classification models and combining the classification scores a final decision is obtained.

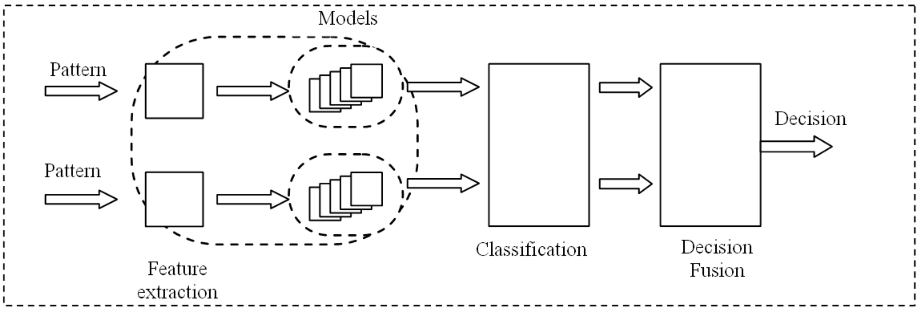


Figure 5.3: Decision fusion multimodal system

There are different algorithms that can be used for combining different scores to obtain the final classification score. These can be simple voting algorithms, which take into account only the ranking scores of each classifier and apply the same weighting, or importance to each classifier. There are also many more complex algorithms that analyze the score distributions for each classifier and take into account also their statistical properties in order to give greater weighting to those classifiers that have superior discriminative properties. This will be more thoroughly explained in the next section.

The main advantage of the decision fusion approach is that it can increase the robustness of the overall recognition system by taking advantage of the properties of the different classification algorithms and also by incorporating the discriminative properties of different types of features and different types of classification algorithms.

The main disadvantage, which is valid for all multimodal systems, is that it needs extra computing time and resources to process the extra information and to train the additional classifiers.

If the requirements for the classification system regarding complexity, time and storage allow the use of more than one type of feature or more than one classifier and the experimental results show that greater accuracy and robustness can be achieved over an unimodal system, then is better to implement a suitable multimodal approach.

5.3 Multimodal Systems for Face Recognition

In biometric systems the most common approach to multimodality is to use more than one biometric signal in order to obtain higher accuracy and robustness. There are many systems that use combinations of biometrics for person identification/verification, such as facial image and speech signal [16], facial image and fingerprint scanning [51], facial image and retinal scanning [115]. Many other such combinational approaches can be found in the literature.

These systems assume that if one biometric signal is affected by noise, the other(s) will not be. Thus the combined system will make the correct decision in the end.

For verification systems a simple logical AND system can be employed at the decision fusion stage, in order to ensure that ALL of the biometric classification techniques were successful. In this case, if a false intruder will try to fool the system by altering some of his biometric characteristics, multimodality will make his job more difficult because he would have to modify more than one biometric pattern.

This is a valid approach and it has proved to be very helpful in many systems [93], but it can only be used when it is possible to concurrently capture and process more than one biometric input. When this is not possible, we can still use a multimodal approach in order to make the system more reliable, by trying to extract more than one type of information, or feature set, from the given pattern and use all the extracted information in the classification stage. By doing this it is likely that at least one type of features will be robust to intrusions attacks or presence of noise.

In face recognition there are many proposed methods which are reported to perform very well in certain conditions. Many of these depreciate their accuracy when the testing conditions are changed and different factors affect the appearance of the face image. A comprehensive review of face recognition techniques was already provided in **Chapter 3** so we will confine ourselves here to providing a brief summary.

Classical algorithms for face recognition based on template matching as described in **Chapter 3** include Discrete Cosine Transform (DCT), Principal Component Analysis (PCA) [109], Linear Discriminant Anaysis (LDA) [15], Elastic Graph Matching (EGM) [62], Hidden Markov Models (HMM) [74]. It is well known that these methods are affected by various factors, in particular by differences in illumination conditions and differences in face pose and face orientation.

Other approaches for face recognition use geometrical features for classifying people. These geometrical features are more robust to changes in illumination and pose but there are cases when it is more difficult to detect certain key points which define the geometrical features - e.g. eye or mouth regions may be partly occluded.

A combination of these two approaches (geometrical features and template matching) could prove to perform very well in different environments. But also combinations of template matching approaches could prove very useful when the extracted templates from the face patterns represent different independent information.

As previous discussed in **Chapter 3** we decided to combine two classical recognition approaches in order to compute similarities between face regions. As proven by our initial tests, the two methods seem to behave differently depending on the properties of the image collection.

As discussed in **Chapter 4** by using additional information extracted from peripheral region we could increase the robustness and accuracy of the Person Recognizer especially considering the nature of the pictures that we intend to work with. This additional region contains information that is not correlated with the information from the face region and also this information is more robust to different factors that affect the result of face classification.

Now that we decided from where to extract the information to be used in the classification stage, we have to address the problem of how can we combine the similarity results from each region and type of features in order to compute the final similarity score between persons. Next section is dedicated to this issue.

5.4 Implementation of Multimodality in a Person Recognition System

In the system proposed for organizing and browsing through digital image collections, the multimodality is present in two cases: first, when using two types of features for face region descriptors, namely discrete cosine transform DCT features and principal component analysis (PCA) applied on a wavelet sub-band features, and second, when features extracted from multiple regions are used for person classification and the results are combined for the final ranking.

In both cases the same method is used for combining the different scores, method that will be presented next, along with the reasons that recommend the use of this method.

We start by formulating the problem of combining multiple score in order to obtain the final similarity measure. Initially we tried to sum the distances from each classifier and use the result as the final score. As we will show this is not a good way of combing the score because of different domains of variations of the distances. Next we decide to normalize the scores to be in a given domain and we tested the intuitive approach of dividing each score by the maximum distances obtained with each of the classifier. Even if this method showed an improvement we will demonstrate that this also is not an optimal way of combining multiple scores. Finally we will present the statistical approach used in the final implementation which proved to be superior to the other normalizing method.

Let us assume we have two classifiers: c_1 and c_2. We have a collection of N patterns $P_1, P_2 \ldots P_N$ and we want to rank the collection of patterns according to their similarity with test pattern p_T, using the two classifiers.

Applying c_1 and c_2 over the pattern collection we will obtain two sets of distances $d_{\gamma 1}$ and $d_{\gamma 2}$ both vectors having the dimension N.

Simple voting techniques take into consideration the ranks of these distances in order to obtain the final ranking.

We can assume that normally, when something (noise) affects one of the classifiers, the distance to the correct class is increased and a distance to a different class decreases. In the discussions that follow we will use two dimensional graphics to represent the classes in order to present these concepts more clearly. Typically the graphics in such cases are highly multidimensional (20+ elements in a feature vector) but this becomes very difficult to represent, describe and visualize. Thus we confine our descriptions to the 2D equivalent case.

In the next case we assume that one of the classifier is affected by noise, which makes the distances resulting from it fuzzier. On the other hand the second classifier, which is not affected by noise, works very well and its distances are correctly returned with a very small distance corresponding to a correct result. Using the voting approach, the correct result (with a small distance) will be ranked by the noise-free classifier as the highest (first place), while the second classifier will return an incorrect result (because of the greater distance due to noise presence there may be other patterns closer to the correct pattern) with a lower ranking for the test pattern, as shown in Figure 5.4. When combining the two classifiers, the system could make the wrong decision if the correct result is ranked low by the second classifier and another pattern is ranked high enough by both classifier.

Let us analyze this case more deeply from Figure 5.4. The classifier in the left side is the noise free classifier. It works well and returns the right pattern as being the first one with a very small distance (high similarity) and all the other patterns are returned with high distances (low similarity). The second classifier is affected by noise so its results are fuzzier and returns the right answer, let's say, on the third position having the distances very close to the first two returned patterns.

When using the voting approach if the first classifier returns on the second position (with a very high distance) one of the patterns that the first classifier returned on the first two positions than the final decision could be wrong or at least could be a tie. This, even tough the first classifier was really sure about its decision and the second one had some fuzzy (patterns really close one to another) results but it wasn't to far from the right decision either.

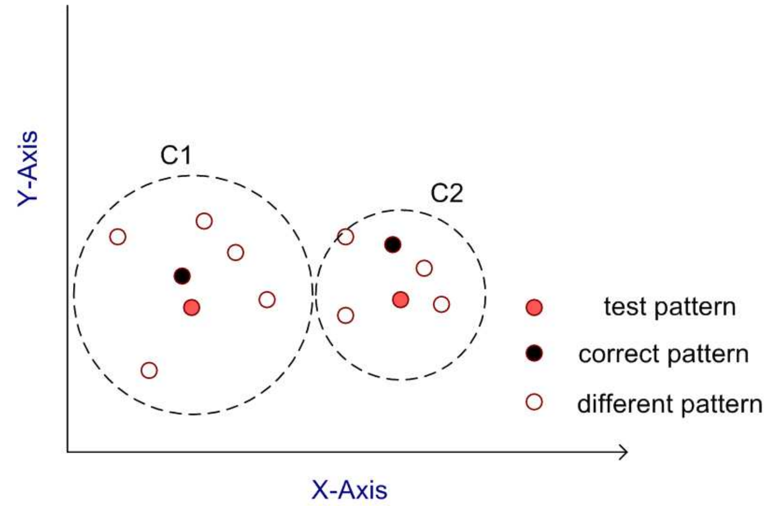


Figure 5.4: Example of two classifiers that will fail for voting approach

We have shown that the voting methods that only consider the rankings, and does not take into consideration the statistics of the distances returned by each classifier could return false decision in fairly common cases.

The simplest method of combining distances in order to obtain a final score is to sum the corresponding distances from each classifier. The problem with this approach is that if the classifiers return the distances in a different absolute range of values (because the features may have different values), then the classifier which returns the distances in the higher absolute range will have a bigger influence in the final score independently of its discriminative properties. In other words, if the range of one set of classifiers lies between 0 and 1000, whereas the range of another set of classifiers lies between 0 and 100 it is likely that classifiers from the first set will dominate.

A straightforward approach to combining distances from different domains is to normalize them using the highest value in order to have all the values in the domain lie within a normalized range of [0,1].

$$d_F(P_i, p_T) = \sum_{j=1}^{2} \frac{d_{c_j}(P_i, p_T)}{max_i(d_{c_j}(P_i, p_T))} \tag{5.1}$$

Where $d_F(P_i, p_T)$ represents the final distance between the pattern P_i and test pattern p_T and $d_{c_j}(P_i, p_T)$ represents the distance between the same patterns using classifier c_j.

Let us consider a very common case where due to some factors we have in one classifier one or more outliers which will give a very high maximum like in Figure 5.5. This will make the normalized distances from this classifier to be very small, giving a higher importance to the other classifier, which if affected by noise will affect the entire classification no matter what the first classifier (which may have a better discriminative property) decides.

Thus we need to find a method of normalizing between different classification algorithms. This problem is, however, somewhat non-trivial because the range of values obtained from any particular classifier will depend on the data set to which that classifier set is applied. Consider, for example, an image collection where all of the images are frontal and taken in consistent lighting conditions. We may find that a particular classifier, when applied to facial regions extracted from this dataset will show a variation in distance ranging from near to zero, being a very close match, up to values of around 100 which represents the extremes of variability between the extracted facial regions. Now consider a different image collection where images are taken over a much wider range of conditions, including outdoor lighting in full summer sun and in cloudy winter conditions and indoor lighting in both close-up portraits and in a variety of poorer conditions, including images captured indoors without flash, or captured at the extreme range of the flash. Clearly this second image set will exhibit a much wider variability in classifier outputs, ranging again from distances close to zero, representing very close matches, to values of 1000 or more due to the much wider variability across this image collection.

Now the reader can understand that it is not a simple matter of normalizing a particular classifier according to a predetermined set of criteria, but rather that the required normalizations are themselves a function of the data set across which the classifier is to be applied.

If we decide to statistically analyze the distance distribution from each classifier it will be possible that the distributions can be modeled as a Gaussian

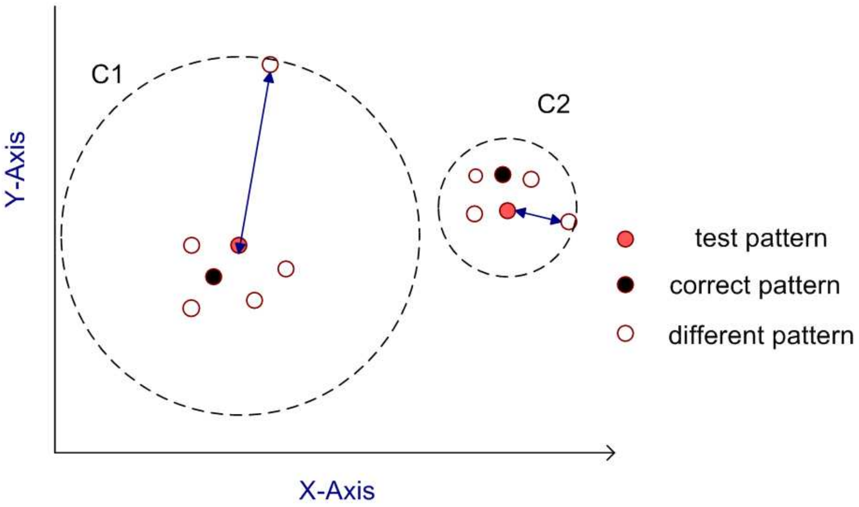


Figure 5.5: Example of two classifiers that will fail using normalization by maximum distance

distribution described by the mean and variance shown in Figure 5.6.

A good classifier with a good discriminative property should have high variance which means that the mean distance should be bigger than the correct distances.

If we want to normalize the distances using the variances from each classifier we should divide each distance by the variance. If we do this for a strong classifier where the variance is high, we can conclude that this classifier will have a smaller weight in the final combination comparing with a weak classifier which exhibits a small variance because the normalized distance is "inversely" proportional to the variance of the classifier.

Another logical approach is to use the mean of the distributions to normalize the distance of each classifier [106]. This makes sense if the classifier has a good discriminative property, in which case the result of the division will be even smaller for the small distances (which correspond to the correct distances) and bigger for higher distances. This approach will also solve the possible different value domains problem of the classifier

$$d_F(P_i, p_T) = \sum_{j=1}^{2} \frac{d_{c_j}(P_i, p_T)}{mean_{i<N}(d_{c_j}(P_i, p_T))} \tag{5.2}$$

Along with the logical explanation above this method also proved helpful in our experiments and comparing with the normalization using the maximum values we obtained significant improvements.

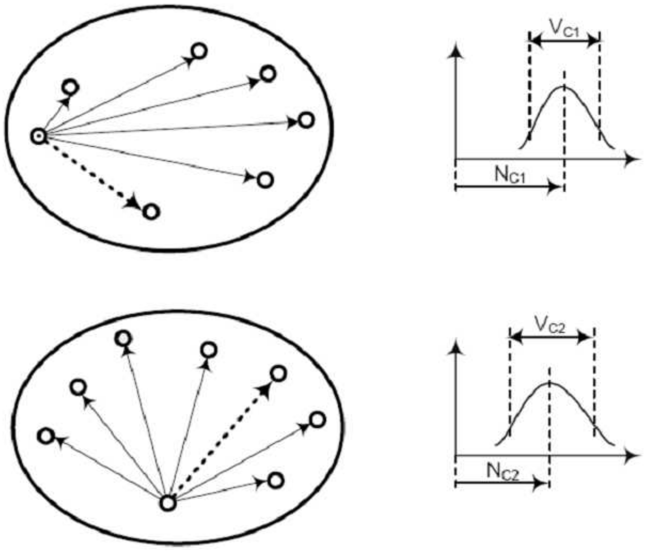


Figure 5.6: Two distance distributions modelled as Gaussian distributions

More complex algorithms can be developed which take into consideration more statistical properties of the distance distributions, but such a study is beyond the scope of this thesis.

The proposed method could be extended to more than two classifiers in the same way using the next formula for n classifiers:

$$d_F(P_i, p_T) = \sum_{j=1}^{n} \frac{d_{c_j}(P_i, p_T)}{mean_{i<N}(d_{c_j}(P_i, p_T))} \tag{5.3}$$

This method was used in the proposed system first for combining the distances resulting from PCA and DCT features inside the face region and also for combining the resulted distances from the face region with the distances obtained comparing the body and hair regions using the colour information features.

Multiple tests were performed in order to validate the applicability of this approach. Section 6.1.1 from **Chapter 6** shows the recognition rates obtained using the two approaches. We performed also other tests that are not presented in **Chapter 6** and in every case the proposed method performed better which lead us to decide to use it in the final implementations.

Chapter

6 Testing Procedures and Preliminary Experimental Results

This chapter presents the testing procedures and the results of various preliminary tests that were performed in order to refine and analyze the performances of different components of the final Person Recognizer system. Section 6.1 will present the initial tests of the system: it begins by describing the collection of images used in the testing phase and continues in section 6.1.1 with a presentation of the results from the automatic retrieval system where only the facial region is used; section 6.1.2 presents the retrieval results where only the peripheral regions are used; and section 6.1.3 presents the results where all three regions are used to provide information to the recognition engine. These results help us to figure out the optimal configuration for implementing in the final image retrieval application.

The second part of the chapter, section 6.2, focusses on the face recognition module of the system. The performance of this module is tested as a classical face recognition system using four standard databases for face recognition: BioID, Acherman, Yale and UMIST. Improvements and conclusions are presented as well. These results will give an idea of how well the face recognition module performs on standard face recognition databases as well as suggesting some ways in which the performance of this module might be improved.

The last section of the chapter presents our conclusions with regard to these initial test results.

6.1 Database Description and Initial Test Procedures

It was stated in the previous section that the proposed retrieval application is intended for use by everyday users and is designed for application to consumer collections of images. In order to test the application in conditions close to this goal, the test database has to incorporate some of the properties of such consumer image collections.

No restrictions are to be applied to the images, e.g. controlled illumination, face size, image quality, face orientation, or number of faces in the images and so on.

The database used in these tests was built by combining collections of images gathered from different users randomly. The entire set of 170 manually extracted face regions from the images from this database is given in Figure 6.1. Every face has associated corresponding peripheral regions. It can be noticed that many types of variations are present in the collection.

One requirement for the retrieval system is to be fully automated so the faces represented in the collection are automatically extracted using the face detector described in **Chapter 2** and each detected person has the corresponding body and hair regions automatically extracted and associated with the original face region. No other manual method was used to align or correct the face, body or hair regions thus the regions shown in this representative set are those extracted automatically using the algorithm described in **Chapter 2**.

The testing procedure consists in taking each person in the collection, computing the similarities against the other persons in the collection, and returning in the end the ranked images/faces according to this similarity measure (distance) which is analyzed.

The classical method for testing the recognition performance is to represent the precision/recall curve [91]. This implies defining a threshold for the similarity measure, or defining a threshold for the maximum rank of the resulted similarity list and counting how many correct results are inside the region defined by the threshold and how many correct results are in the entire collection.

For our database this is more difficult, because it is not a symmetrical database. Some of the persons are present in the collection only two or three times whereas some people occur ten or more times.

Therefore, in order to compute a performance measure for the system applied on our database, instead of defining a threshold, we need to consider the entire ranked list that is calculated and see where the correct face regions are located in that list. In the end we sum the ranks of the correct face regions to obtain the final recognition score for that person. The score returned by the system

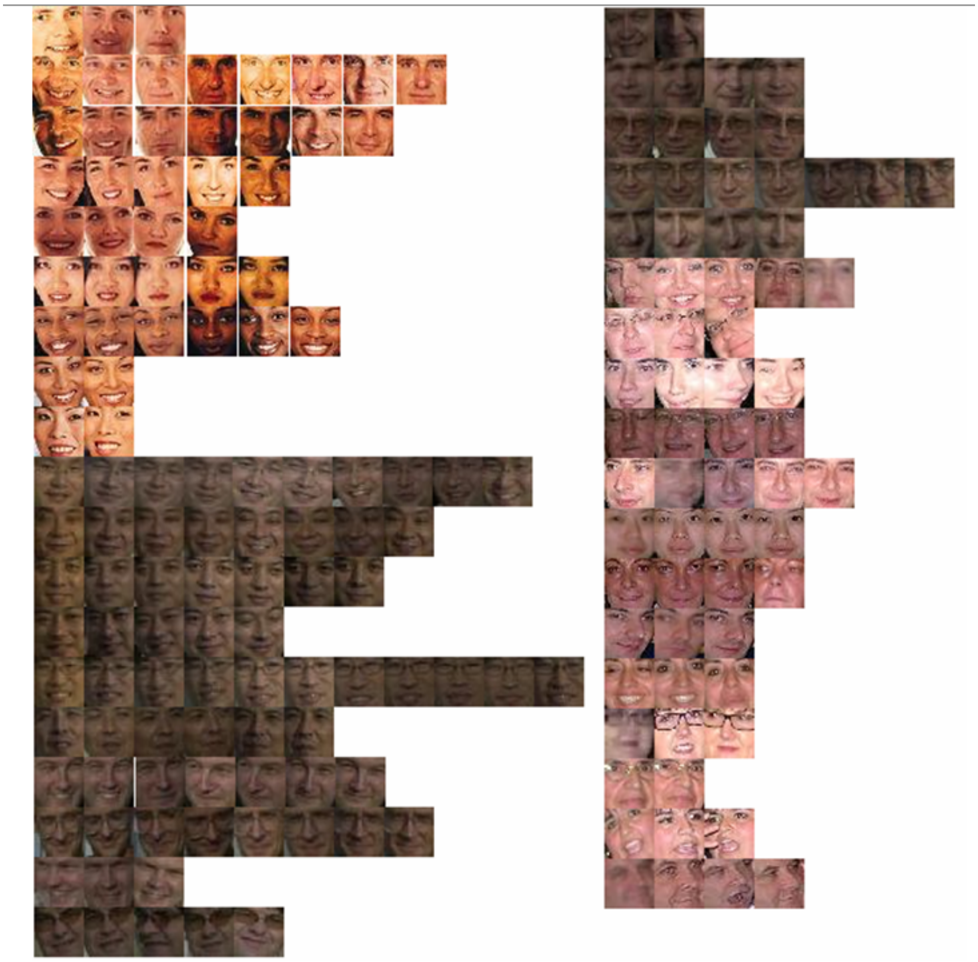


Figure 6.1: Database used to test the retrieval application

is calculated using the following algorithm: if the i_{th} person has n correct faces in the database, then after classification we find the rank (after ordering the similarity distances) for each of the n correct pictures and we sum these ranks.

Thus we can evaluate the system using the following formula:

$$result = \sum_{i=1}^{N} \sum_{j=1}^{n_i} rank(Q_j) \tag{6.1}$$

where: Q is the rank of the correct answer Q_j after classification of the query face Q_i, and n_i is the number of the correct answers for i^{th} query.

Using this algorithm, a perfect retrieval system for this database (which will return the correct answers as being the n_i first best matches) will return a score equal to 4136 which corresponds to 100% and the worst retrieval system (which will return the correct answers as being the $N - n_j$ last best matches) will return a score of 175438 which corresponds to 0%.

The score of 4136 was obtained by retrieving each sample in the collection and the system should return the correct samples with the lowest ranks. By summing all these ranks for each sample we obtain this number. For instance if we search the first image in the database, the perfect retrieval will be when the other two images of the same individual will be in the first two images so the score will be 1+2=3. If we do the same for each image and sum all scores in the end we will get 4136. The worse score 175438 was obtained in the same way but when after classification the correct answer will have the highest ranks. In the same example for the first image the worse retrieval will return the images on the last two positions 169 and 170 and the result is 339. If we repeat this for each image we will obtain 175438.

For each score applying a simple algorithm we can calculate a retrieval rate in percents (%) which is easier to understand.

6.1.1 Retrieval Results for Face Region

Two measures are defined to test the system:

(i) the *Image Retrieval rate (IR)*:

$$IR_{Q_i} = \sum_{j=1}^{N_i} rank(I_{Q_j}) \tag{6.2}$$

where IR_{Q_i} represents the Image Retrieval rate for person query Q_i, N_i represents the correct number of images with person Q_i in them and the rank represents the position of the image I_{Q_j} after classification.

(ii) the *Face Retrieval rate (FR)*:

$$FR_{Q_i} = \sum_{j=1}^{N_i'} rank(Q_j) \tag{6.3}$$

where FR_{Q_i} represents the Face Retrieval rate for person query Q_i, N_i' represents the number of presences of person Q_i in the collections and the rank represents the position of the face Q_j after classification.

The FR is more accurate concerning the system performance whereas the IR rate gives a better "overview" of how the system visually performs.

Using the FR rate we can tune the system in order to get the best results from each component. This measurement is objective and shows exactly how each component classifies the regions. The IR rate is a more subjective measurement.

Let us consider the case when a search sample is given to the system and the system returns in the first images, an image which contains the correct person in it but it also has some other persons in it: if the recognition system correctly identified the right person in the image than the two measurements will be the same but if the recognition system wrongly identified another person in that image as being the most similar with the given example than the FR will count this as a mistake and will drop, whereas but the IR will not consider this as a mistake because the image correctly contains the right person. Also the user will count this as a hit when the system returns the result to him.

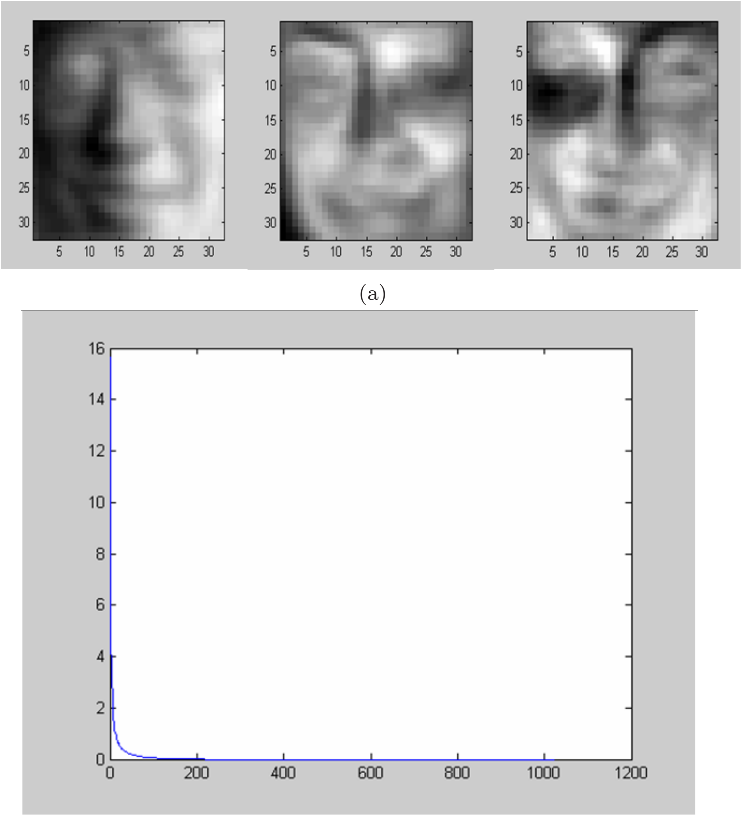
We have to take this into consideration because often due to very similar conditions of illumination different people will have high similarity or when using body features people will be in front on another thus influencing the result of the classification.

This initial database was used to design the face recognition module and to tune its parameters to optimize the overall rate of recognition for each face candidate region. In order to achieve that, we have used only the Face retrieval rate (FR) score when testing and performing face recognition optimizations.

Our first set of initial tests was based on the use of the classical PCA method described in **Chapter 3**. Prior to PCA analysis each region was resized to a standard size (32x32, 64x64 and 128x128) and transformed into a greyscale image. The similarity score between faces was computed as the Euclidean distance in the feature space.

The parameters that we can tune in order to see what is the best configuration for PCA feature extraction are (i) the size of the normalized face region and (ii) the number of PCA coefficients that will be used for classification. In Figure 6.2 we represented the first three eigenvectors (eigenfaces) and also the evolution of the eigenvalues with their rank when resizing the face regions to

32x32. We can notice that the eigenvalues are mostly concentrated in the first 50 ranks which gives us information about how many eigenvectors to use for projection.



(a)

(b)

Figure 6.2: Eigenvectors and Eigenvalues representation

The best retrieval rate obtained using this method for different configurations was 78.52%. We observed that variations in the face regions, in particular with respect to illumination, pose and facial expressions can influence the results so we decided to filter the original image in order to minimize these effects.

We started by applying the PCA method on the wavelet decomposition on LL sub-band which corresponds to performing a low pass filtering on both vertical and horizontal directions of the original image. If the original image has the dimension NxN each of the sub-bands will have the dimension $N/2$x$N/2$.

Initially we used the first 50 PCA coefficients. The rates obtained for different sizes of the original images were: 82.41% for 64x64, 82.69% for 128x128 and 83.1% for 32x32.

Table 6.1: Recognition rates for different normalization size

Size	32x32	64x64	128x128
Recognition Rate	83.1%	82.41%	82.69%

We also tried to apply the PCA method on the 2nd level LL sub-band of the wavelet decomposition for images of 128x128 pixels but the retrieval rate dropped to 81.69%.

It is a known fact that the first eigenvectors correspond to the illumination component in the image. It is better not to use the PCA coefficients corresponding to these eigenvectors in the classification stage in order to obtain an improved robustness to illumination variations.

We choused to resize the images to 32x32 pixels. A possible explanation for better results using this size is the fact that the average size in the database is smaller than 64x64 which means that some faces will be up-sized and some down-sized. Up-sizing the images is known to introduce artifacts in the image which can reduce recognition rates. When resizing to 32x32 pixels practically all face regions will be down-sized and if this is properly implemented there will be negligible distortion of the face region. A further point worth mentioning is that sub-sampling a face region corresponds to an averaging of the region and may also help reduce image noise and remove some higher frequency artifacts which are detrimental to the recognition algorithm.

When the eigenvectors corresponding to the eigenvalues ranked between 2 and 40 are used in the classification stage, the retrieval rate is 84.71%. When the eigenvectors from 2 to 16 are used the retrieval rate actually increased to 85.21% and the best score of 85.27% was obtained when using the eigenvectors from 2 to 18.

Table 6.2: Recognition rates for different number of eigenvectors

no. of eigevectors	2:40	2:18	2:16
Recognition Rate	84.71%	85.27%	85.21%

Pose variations still significantly influence the results of the retrieval significantly so we decided to adopt a well known idea - see for example [17] - which is to use the mirror image of a face region in the classification process. For each test image the mirror image was computed and then the wavelet transform was applied and the mean was subtracted. The mirror image was then projected into the eigenspace.

In the test phase, when the mirror image is used, the similarity distance between a detected test face and a trained face from the database is measured for both the original test face and for its mirror image. The actual distance for that test face is then recorded as the minimum of these two distances.

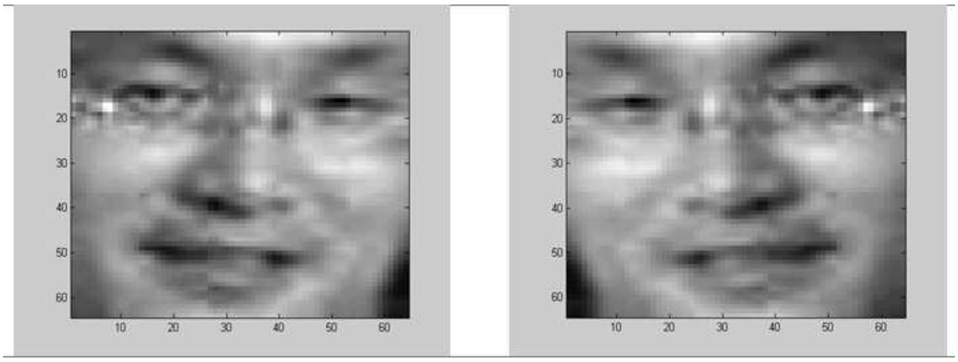


Figure 6.3: The original image along with its mirror representation

Using this technique it will not affect the speed, complexity and storage requirements of the overall retrieval system because it won't be used in the training stage, it will only be used in the testing/retrieval stage. Also the mirror image doesn't have to be stored in the memory, as it can be generated from the original image by loading horizontal lines of pixels in reverse order. Furthermore, the PCA projection is fast to compute as well. The improvement obtained using this method was a 88.29% retrieval rate for 32x32 pixels image size and 1 level wavelet decomposition using the eigenvectors from 2 to 16. That represents more than a 5% boost in recognition rates which is quite significant.

For our second set of preliminary testing the DCT-based face recognition algorithm described in **Chapter 3** was our focus. This algorithm involves computing the 2D DCT spectrum of the grayscale image. The spectrum will have the same dimension as the original image. The similarity measure used is, again, the Euclidean distance between feature vectors, computed in the DCT space. The possible parameters which could be varied in order to optimize DCT face retrieval rates are again, (i) the normalized size for the face regions and (ii) the number of DCT coefficients used in the classification stage.

The best results were obtained when the first 10% of the coefficients from the spectrum were used, corresponding to low frequencies. The first DCT coefficient corresponding to the continuous, or DC component, was discarded.

The retrieval rates for different sizes of face region were: 83.05% for 128x128, 83.07% for 64x64 and 84.39% for 32x32 which concur with the best results obtained using PCA features. Note that applying the mirror image technique won't help the performance of the DCT algorithm because the spectrum of the two faces, original and its mirror, is identical. Thus its use is restricted to the PCA based algorithm.

When combining the best scores from the two algorithms using the maximum normalization described in **Chapter 5**, the resulting recognition rate was 88.30%. Using our proposed method, also described in **Chapter 5**, for combining multiple scores using the PCA and DCT distances computed on 32x32 face regions the best score obtained on the test database was 89.77%.

The recognition rates obtained when combining the face recognition rate with the body & hair similarity measures using the same method for combination of individual classifier decisions was 95.41% for this initial test dataset.

Encouraged by the results of these first tests we increased the size of the database to 520 images containing around 800 manually marked faces. The face detector only reported around 700 faces and thus only those faces which were automatically detected were taken into consideration when calculating the performance statistics.

The retrieval rates obtained using only the face regions are reported in Table 6.3 and represented in Figure 6.4.

Table 6.3: Retrieval Rates for Face region

Features	DCT	PCA	DCT+PCA
Face Retrieval Rate (FR)	70.74%	72.85%	73.5%
Image Retrieval Rate (IR)	80.32%	82.23%	82.28%

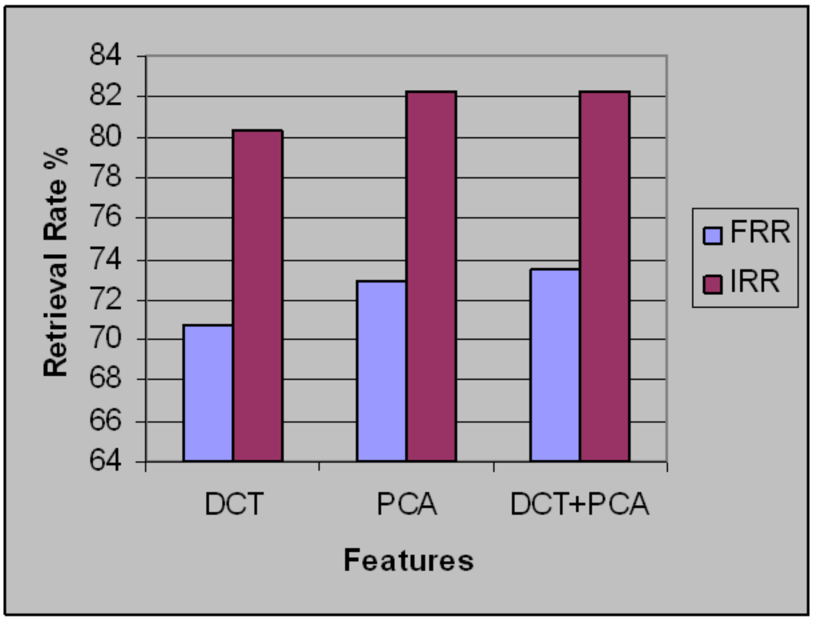


Figure 6.4: Retrieval Rates for Face region

It can be noted from the figure that when combining the results from PCA and DCT methods both retrieval rates are higher. Also the Image Retrieval rate has a higher value than the Face Retrieval rate which may suggest that there is a correlation between face regions in different images in the collection.

6.1.2 Retrieval Results for Peripheral Regions

Tests were also performed to find the optimal configuration for extracting information from the peripheral regions as described in **Chapter 4**. In this case configuration refers to a determination of (i) what colour space to use for computing the histogram and colour correlogram, and (ii) the maximum distance over which to compute the colour autocorrelogram and banded correlogram as described in **Chapter 4**.

For classification we used the similarity measure between correlogram feature vectors (Q_i, Q_j) given by the formula:

$$S(Q_i, Q_j) = \sum_{m,n<N,k<d} \frac{|f_{Q_i}(m,n,k) - f_{Q_j}(m,n,k)|}{1 + f_{Q_i}(m,n,k) + f_{Q_j}(m,n,k)} \tag{6.4}$$

where S represents the similarity between region Q_i and Q_j, N represents the number of colors, $f_{Q_i}()$ is the feature vector extracted from region Q_i and d is the maximum distance for calculating the features.

For these tests the Euclidean distance was used as a measure of similarity between histogram feature vectors.

When combining both regions, the distance between two persons is determined as the sum of distances between the body regions and the hair regions of those persons. In other words both peripheral regions were accorded equal priority weightings for these tests.

We used 3 colormaps for comparisons: first the vga colourmap with 16 colours, second the GIF colormap with 256 colours and finally the reduced GIF colourmap with 64 colors. These colourmaps are illustrated in the next figure.

The Floyd-Steinberg error diffusion dithering algorithm was used for changing between image colourmap representation. The maximum distances used to calculate the colour autocorrelogram and banded colour autocorrelogram features were set to values of 2, 4, 7 and 9 pixels.

The results obtained for each of the two peripheral regions and also for the case when they are combined, with equal weightings, are reported in Table 6.4 below.

It can be noted from the table that the vga colourmaps with only 16 colours gives better results than the GIF colourmaps with 256 or 64 colours. Also the Banded AutoCorrelogram performs better than the AutoCorrelogram features.

Table 6.4: Experimental results

Features(color,d)	Body	Hair	Body+Hair
Hist (16)	21897 - 89.63%		16280 - 92.91%
Hist (64)	20454 - 90.47%		15333 - 93.46%
Hist (256)	20861 - 90.23%		15239-93.51%
Bacor (16,2)	22194 - 89.45%	2219 - 89.45%	15441- 93.40%
Bacor (64,2)	20594 - 90.39%	24369 - 88.18%	16546 - 92.75%
Bacor (256,2)	22650 - 89.19%	22513 - 89.27%	17348 - 92.28%
Bacor (16,4)	19918 - 90.78%	22848 - 89.07%	14559 - 93.91% best
Bacor (64,4)	19762 - 90.87%	25063 - 87.78%	16095 - 93.01%
Bacor (256,4)	22676 - 89.17%	23155 - 88.89%	17708 - 92.07%
Bacor (16,7)	18925 - 91.36%	24359 - 88.19%	14602 - 93.89%
Bacor (64,7)	19381 - 91.10%	26129 - 87.16%	16234 - 92.93%
Bacor (256,7)	22691 - 89.16%	24127 - 88.33%	18313 - 91.72%
Bacor (16,9)	18785 - 91.44%	26794 - 86.79%	14769 - 93.79%
Bacor (64,9)	19290 - 91.15%	26798 - 86.77%	16365 - 92.86%
Bacor (256,9)	22700 - 89.16%	24707 - 87.99%	18568 - 91.57%
Acor (16,2)	21765 - 89.70%	22196 - 89.45%	15116 - 93.59%
Acor (64,2)	20667 - 90.34%	23979 - 88.41%	16549 - 92.75%
Acor (256,2)	22709 - 89.15%	22758 - 89.12%	17727 - 92.06%
Acor (16,4)	20158 - 90.64%	22774 - 89.11%	14877 - 93.72%
Acor (64,4)	20104 - 90.67%	24148 - 88.31%	16296 - 92.9%
Acor (256,4)	22804 - 89.10%	23757 - 88.54%	18437 - 91.65%
Acor (16,7)	19706 - 90.91%	24905 - 87.87%	15675 - 93.26%
Acor (64,7)	19926 - 90.78%	25905 - 87.29%	16799 - 92.6%
Acor (256,7)	23051 - 88.95%	25613 - 87.46%	19471 - 91.04%
Acor (16,9)	19657 - 90.93%	26310 - 87.05%	16319 - 92.88%
Acor (64,9)	20014 - 90.73%	26964 - 86.67%	17327 - 92.29%
Acor (256,9)	23141 - 88.9%	26886 - 86.71%	20159 - 90.64%

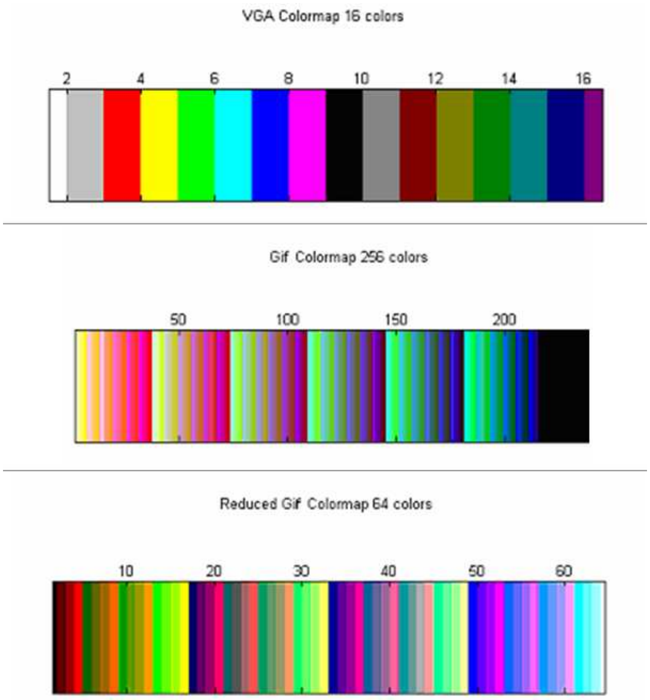


Figure 6.5: Colormaps used for testing

Thus better results were obtained by using a smaller number of colours.

One possible explanation for this observation is that when the number of colours is smaller we project more similar colours in the same clusters. Thus we eliminate the effects of small variations of illumination against colour features. As expected when combing the features from both regions, the results improved compared to the results obtained when each region is analyzed on its own.

For a better understanding Figure 6.6 contains graphical representations of the retrieval rates for different combinations of parameters.

As expected when combing the features from both regions, the results improved compared to the results obtained when each region is analyzed on its own. The best result on this database (93.91%) was obtained using Banded AutoCorrelogram features computed using the vga colourmap with 16 colors and with a maximum distance set to 4 pixels.

The peripheral regions were not resized to a standard size as we did with the face regions. We don't need to resize them because the size of the feature vectors does not depend on the size of the regions. The dimension of the feature vectors depends only on the color space used to compute them.

Also by resizing them to a smaller dimension we risk loosing important information for classification. The only reason that could force us to down-size the regions is if the feature extraction algorithm will be to slow to apply on the full size regions. This is not the case in our system.

Again it can be noted from Figure 6.6 that when using only 16 colours for computing the features, our results are significantly better. We can also note that the combination of the two regions gives significantly better results than those obtained when either region is used on its own to calculate a similarity measure.

6.1.3 Retrieval Results using Combined Face & Peripheral Regions

When combining the scores obtained using face similarity measures with the ones obtained from body & hair similarity measures for the same database described in the previous section, the following retrieval rates were obtained:

Table 6.5: Retrieval Rates for Combination of Regions

Regions	Face	Surrounding	Combination
Face Retrieval Rate (FR)	73.5%	79.14%	79.18%
Image Retrieval Rate (IR)	82.28%	83.65%	85.9%

It can be observed that the final retrieval score obtained through a combining of the similarity measures from face retrieval and surrounding regions retrieval increases, although not as significantly as in our initial tests.

These rates were obtained using the optimal configuration for feature vector extraction, as follows:

- face regions resized to 32x32 pixels, converted to gray scale images.

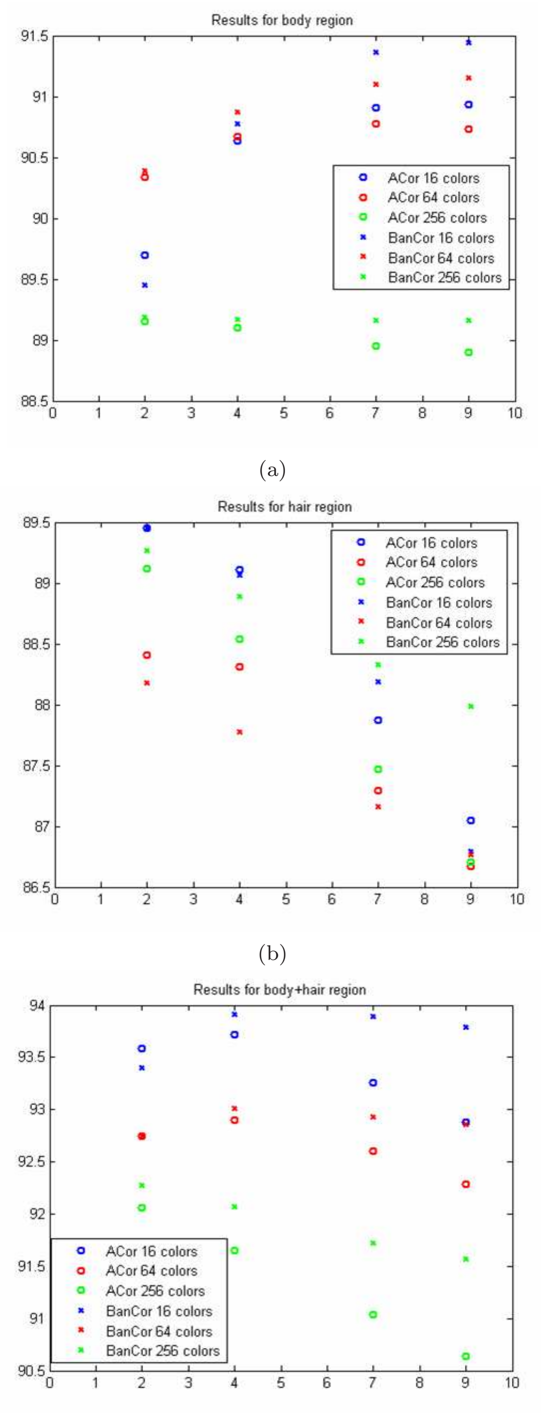
- DCT features extracted using only first 100 coefficients (except the first one).

- for PCA: use 1 level wavelet decomposition LL sub-band, and coefficients corresponding to 2-18 eigenvalues and use mirror face approach.

- - for body& hair features, computing banded autocorrelogram (with maximum distance 4 and the VGA 16 colors colormap).

Other important factors have to be analyzed in order to build a viable retrieval system, such as the time required to train the collection of images, the time required to search for a particular person, the size of the resulting feature vector database.

All these factors depend on the size of the images in the database. If the pictures are large, this implies that more information must be processed, analyzed and stored. When training the described database with 520 images with



(a)

(b)

(c)

Figure 6.6: Experimental results

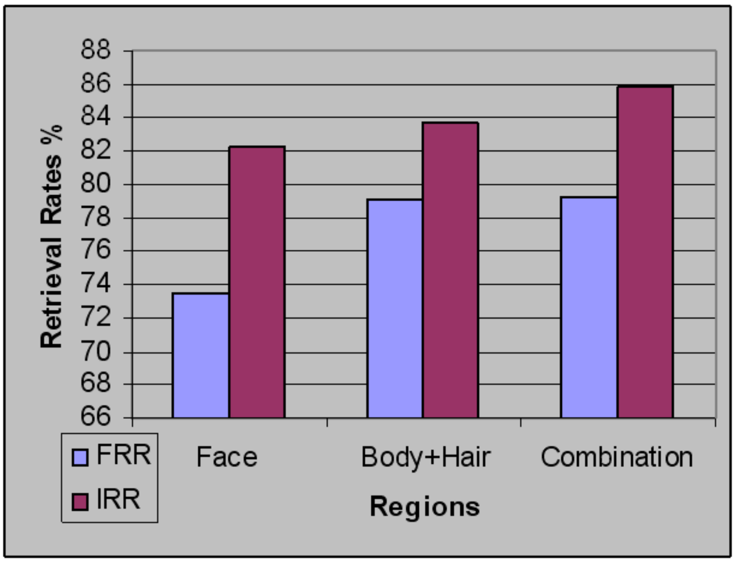


Figure 6.7: Retrieval rates using face and additional features

an average picture size of 1200 x 800 pixels, the time needed for training is in the region of 5 minutes. A larger database was tested (1000 images with an average size of 1500x1200 pixels) and the training time increases to around 20 minutes.

The training time for each image in the database requires the following process steps to be executed:

- detecting the persons in each image and then, for each person:
- extracting the face region
- transform them into grayscale
- extract DCT features and store the features
- apply wavelet decomposition
- extract PCA coefficients from LL subband and store the features
- extract body & hair regions if possible
- compute color correlogram and store the features

It should be mentioned that of the gross training time, more than 90% is required by the face detector module, which even though it is one of the fastest face detectors reported in the literature, is by far the slowest component in the system. This suggests that if face regions were marked in the image metadata the system performance would be an order of magnitude faster. We remark that the latest consumer digital cameras have begun to incorporate face tracking technology and thus the availability of such metadata within the image EXIF header is very likely in the near future

The time needed to search for a person in the database is significantly faster than the training process. Typically it is less than one second even for the largest database we used in these tests which had more than 1000 images.

Regarding the storage size of the feature database, it typically requires less than 1% of the storage size for the image collection, so this will not be an impediment to practical implementations of our system.

6.2 Comprehensive Testing of the Face Recognition Module

In order to analyze more closely the behavior of the face recognition module, we tested its accuracy on four standard face recognition databases. By using standard research databases where particular image variations are controlled, we will be able to quantify, and qualitatively analyze the variation factors that influence the results of the recognition process. Furthermore it should be possible to determine how the influence of these factors can be reduced or eliminated.

Each database used in these tests has different types of variations between images. For instance the UMIST database [45] has very limited variations; the YALE database [110] presents the faces under a range of different illuminations and different facial expressions; the Acherman database [12] has more variations regarding the orientation of the faces and the last database, BioID [11], has limited variations of pose and illumination but also contains more images of the same persons which permits us to study the influence of the number of images per person, used for training, against the retrieval accuracy of the system.

By analyzing the results of these tests we can learn how to configure best the recognition module in order to minimize the influence of these variation factors over the accuracy of the module.

6.2.1 Testing Procedure

The idea is to test only the face recognition module on standard face recognition databases, in order to analyze its performances as a classic face recognition system.

For each database, different numbers of faces were used in the training stage and the rest of them were used for testing.

The method used for classification is a "nearest neighborhood" algorithm using the combined distances from DCT and PCA features. This implies that the test face belongs to the cluster of the closest face in the feature space.

As an example of the testing procedure let us assume that the collection of images has 10 individuals with 10 images per person. The first test consists in choosing as training gallery the first image for each individual.

Using these training images the PCA eigenvectors, or basis vectors, are computed and stored. Then DCT and PCA coefficients are extracted and stored for each face.

The retrieval test set consists of the remaining 9 faces of each individual. Each face in the training set will be matched against all the faces in the test set. If the closest face belongs to the same person, then the number of correct recognition hits is increased by one. In the end the number of hits is compared with the number of faces tested and the recognition rate is computed.

The next test consists in adding a second image from the test set to the training set and performing the same procedure: re-compute the PCA eigenvectors, extract DCT and PCA coefficients for the new training set (based on 2 images per person) and test the retrieval of each face in the remaining test collection of 8 images per person.

Additional tests are performed for three, four and more face regions used for training. These tests are ceased when the number of faces per person in the training set becomes higher than the number of faces per person remaining in the retrieval test set.

The preprocessing consists of the same stages as presented in the previous section for testing the retrieval application. The faces are cropped using the face detector, transformed in greyscale, histogram equalization is performed, and then the DCT and PCA features are calculated.

6.2.2 BioID Database Testing

The BioID database contains 400 pictures of 20 different persons. The faces presented are frontal views with variations in facial expression and illumination. The database is represented in Figure 6.8.

Several tests were performed using in the training stage different numbers of faces (from 1 to 10 faces) to see how this affects the recognition rate. Faces were cropped to 32x32 pixels size, the number of DCT coefficients used was 100 and the PCA coefficients used were eigenfaces 2 to 18 corresponding to the ranked eigenvalues.

The recognition rates obtained using only DCT and only PCA features along with the recognition rates using their combination are presented in next table:

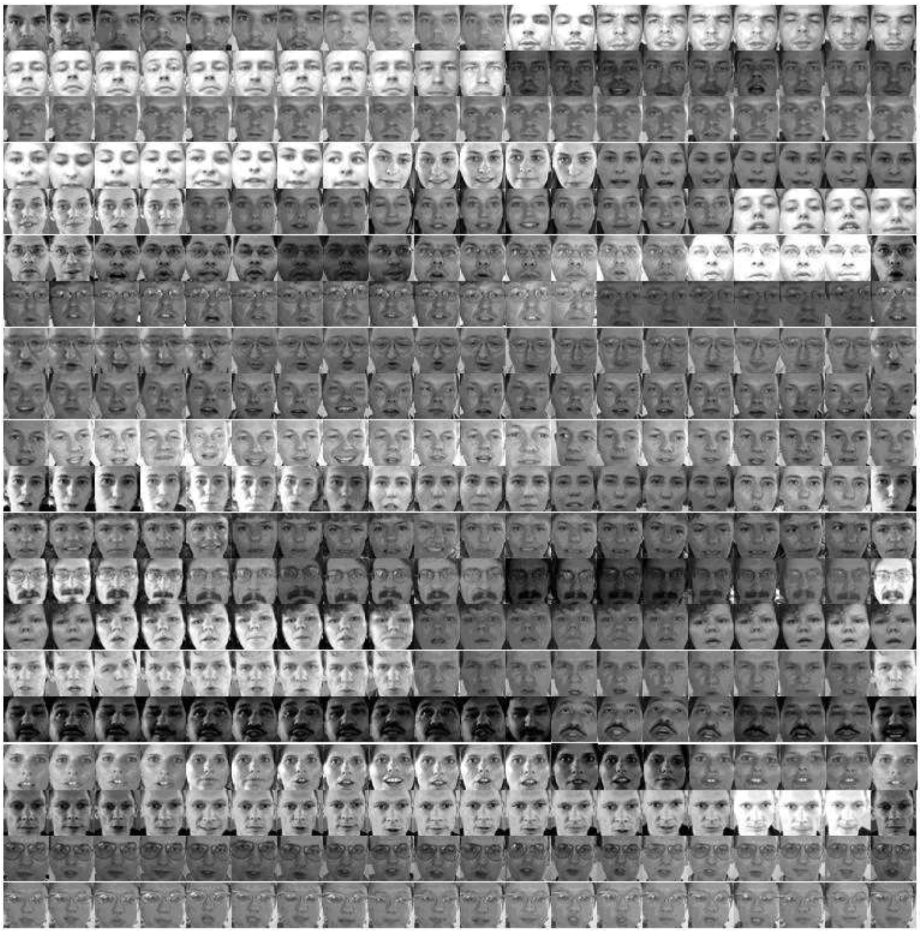


Figure 6.8: BioID database

Table 6.6: Recognition Rates for BioID database

DCT (%)	PCA(%)	Combination (%)	No faces train/test
50,51	47,9	59,7	1/19
70	71.6	74.7	2/18
76.7	77.6	82.35	3/17
79.68	83.44	83.43	4/16
84	86	87.6	5/15
84.64	87.14	88.21	6/14
86.54	88	90	7/13
90	92	91.67	8/12
95.45	92.27	94.55	9/11
98	94	97	10/10

A graphical representation of the evolution of the recognition rate against the number of faces used for training is given in Figure 6.9.

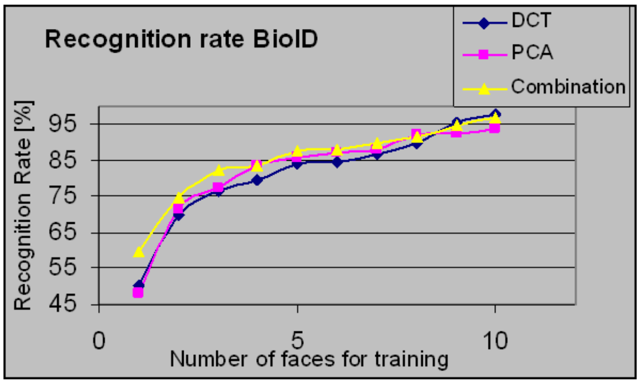


Figure 6.9: Results for BioID database

Other tests were performed using different combinations for the number of PCA and DCT coefficients but the results did not improve significantly.

In order to see how the classification algorithm discriminates between classes, we analyzed the intra and inter class distance distributions for DCT and PCA approaches. The red line represents the distribution intra-class, that is all distances between faces from the same individual, summed across the entire collection. The blue line represents the distribution inter-class, that is distances between faces from different individuals.

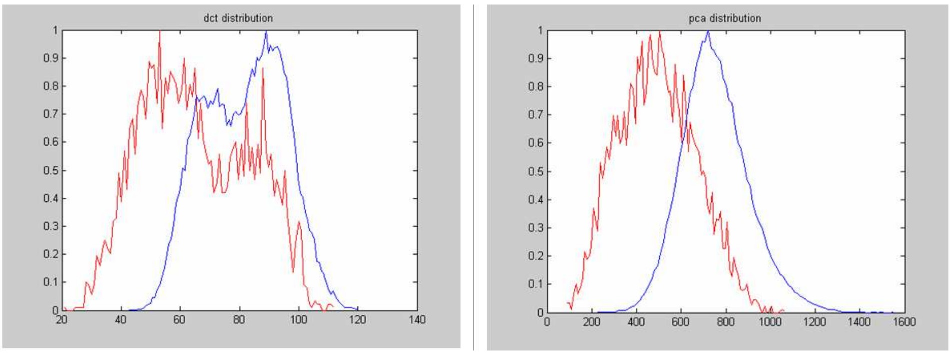


Figure 6.10: Distance distributions for BioID database

Variations over the same individual in the database are determined by different illuminations and face expressions, but these variations are limited. As would be expected adding new faces to the training set will improve the recognition rates. Both methods give similar results and the combination is also close to their rates when multiple images are available for the same individual. However we note that when a small number of images are used in the training set the combination of two methods performs noticeably better than either method used separately.

The PCA distance distributions can be easily modeled using Gaussian distributions and are easily separated. The DCT distributions are more complex and can be modeled using a mixture of two Gaussian distributions.

A possible explanation of the second minor peak in the DCT distributions can be that clusters having the same illumination can be projected closer in the face space between different people than clusters with different illumination of the same individuals. For instance two different individuals having similar illumination will have a higher similarity, in DCT space, than two instances of the same individual with very different illuminations. And in this particular database there are some groups of faces across different individuals with very similar illumination.

6.2.3 Achermann Database Testing

The Achermann database contains 260 images of 26 people, each with 10 images. The main variations in this collection are pose variations of the faces. The preprocessing stages are the same as the ones used for testing the BioID database. The faces in the database are represented in the next figure.

The testing procedure is the same as for BioID. Firstly we train the database with different numbers of faces (from 1 to 5 faces) and test retrieval rates with the remaining faces (from 9 down to 5 faces depending on the number used for training).

The obtained results are given in the next table and are represented in Figure 6.12.

Table 6.7: Recognition Rates for Achermann database

DCT (%)	PCA(%)	Combination (%)	No faces train/test
56.41	52.99	59.82	1/9
64.9	56.25	62.5	2/8
75.8	65.9	73	3/7
82.6	77.5	82	4/6
92.8	82.3	82.93	5/5

The distributions of the distances are represented in Fig. 6.13 where the red and blue lines have the same significance as previously described for the BioID image set.

Again we can notice that using more faces in the training stage improves the recognition rates. The DCT approach performs slightly better than the PCA

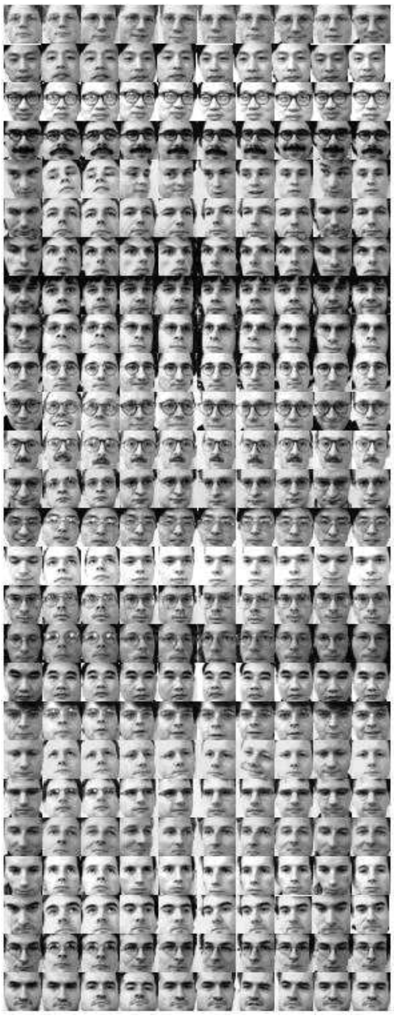


Figure 6.11: Achermann database

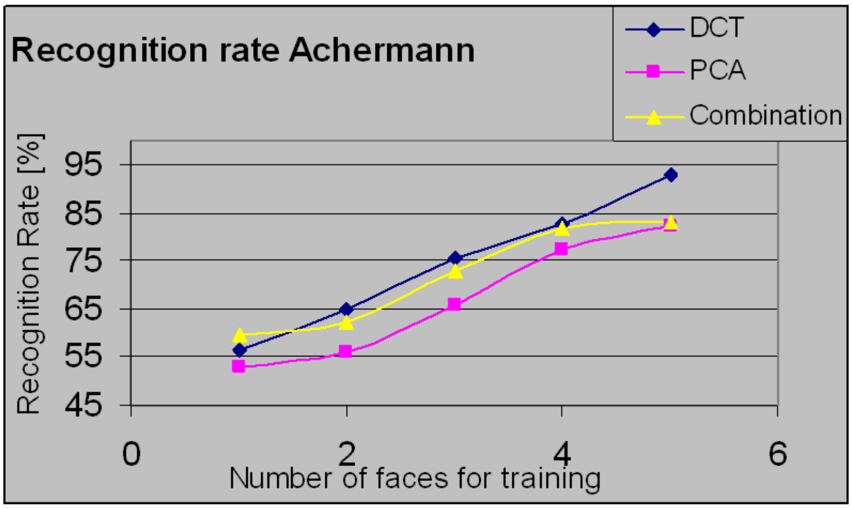


Figure 6.12: Results for Achermann database

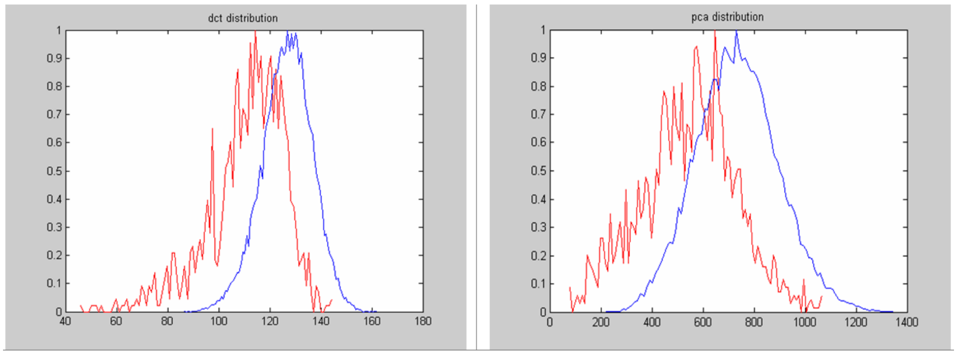


Figure 6.13: Distance distributions for Achermann database

approach and the combination between them. This is probably because the variations in this particular database are more limited than some of the other test databases, especially with respect to illumination. Again the combinational method performs better when a low number of faces are used for training.

Distance distributions between the two approaches are quite similar in shape but with different domains. All distributions have the appearance of typical Gaussian distributions and thus can be modeled this way.

6.2.4 UMIST Database Testing

The UMIST database is the simplest collection that we used for testing because there are relatively small variations across the face regions with respect to illumination and pose. It contains 100 images of 10 individuals. This database is represented in Figure 6.14.

The preprocessing steps are again identical to the ones used for the other databases.

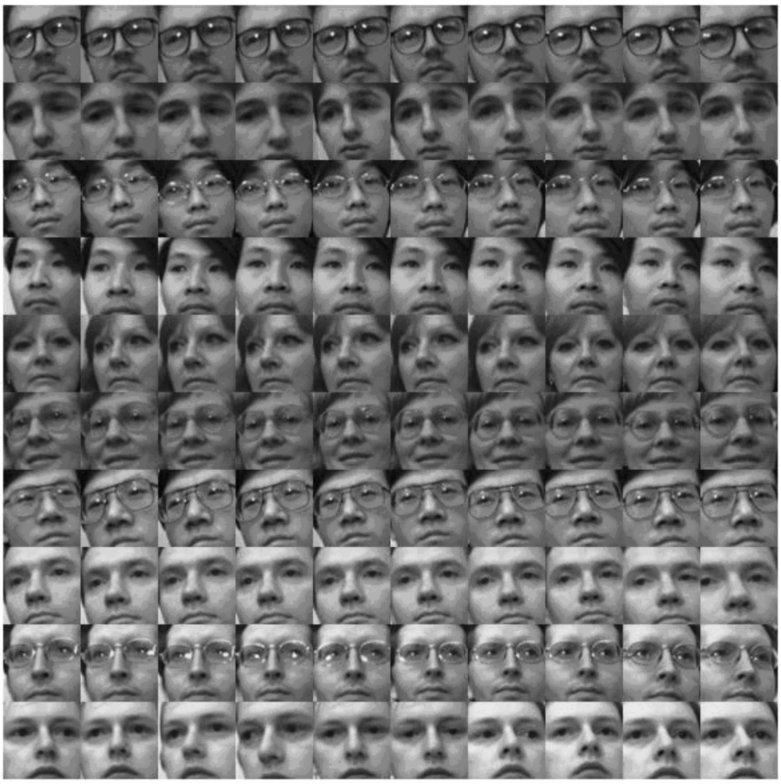


Figure 6.14: UMIST database

The recognition rates are very high especially when more faces are available for training. This was expected due to the relatively small variations in face illumination and pose.

Table 6.8: Recognition Rates for UMIST database

DCT (%)	PCA(%)	Combination (%)	No faces train/test
77	53	58.8	1/9
81.25	48.75	63.75	2/8
87.1	68.5	85.7	3/7
91.6	83.33	91.6	4/6
98	98	98	5/5

The graphical representation is given in the next figure.

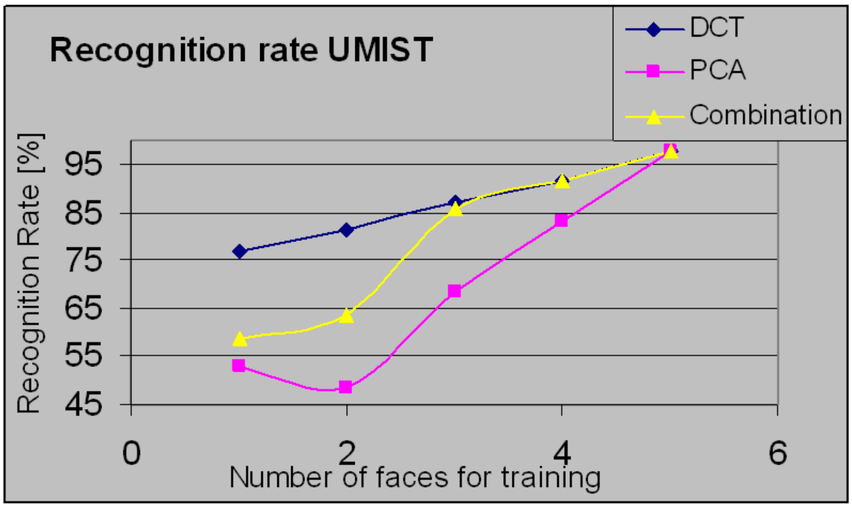


Figure 6.15: Results for UMIST database

Again the DCT approach performs better than the PCA due to the very low variations in the image collection. In this image dataset there are hardly any noticeable variations in illumination, and only some small variations in facial expression. Thus it was expected that such high recognition rates would be attained for this particular collection.

It can be observed also from the distance distributions shown in Figure 6.16 that the classification works very well, the intra and inter class distributions are very well separated in both cases but especially for the DCT approach. Again a Gaussian distribution can be used to model the distances.

6.2.5 Yale Database Testing

The Yale database is used in many publications for reporting the results of many face recognition systems. There are large variations of illumination, face expres-

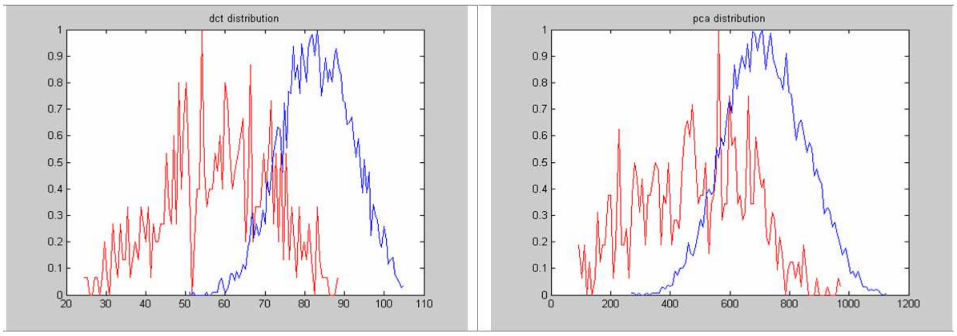


Figure 6.16: Distance distributions for UMIST database

sion and appearance in the database which makes it very useful and suitable for testing the robustness of any face recognition system.

The database is represented in Figure 6.17 and contains 165 pictures of 15 people each with 11 images. The preprocessing steps are the same as for the previous face collections.

The testing procedure is identical to the one described previously and the results obtained are given in the next table and are represented in Figure 6.18.

Table 6.9: Recognition Rates for Yale database

DCT (%)	PCA(%)	Combination (%)	No faces train/test
54.67	44.33	56	1/10
71.11	63.33	72.59	2/9
77.08	67.92	73.75	3/8
80.48	70.95	77.62	4/7
82.22	77.78	78.89	5/6

The recognition rates are as low as 44% when small numbers of faces are available for training due to the large variations across this image collection. However these increase to 75% and higher when 3 or more faces are used for training. The recognition rates are still smaller than for the other databases due to illumination and expression variations. Again the combinational approach seems to perform better when there are smaller numbers of images available for training the models - typically for 3 or fewer training images.

The distance distributions are represented in Fig 6.19

As expected the distributions are very noisy especially for the DCT approach where the intra and inter class are not well separated. The peaks in the PCA distributions are better separated but also a large number of intra class distances are larger than inter class distances. This again is due to the large variations in the collection.

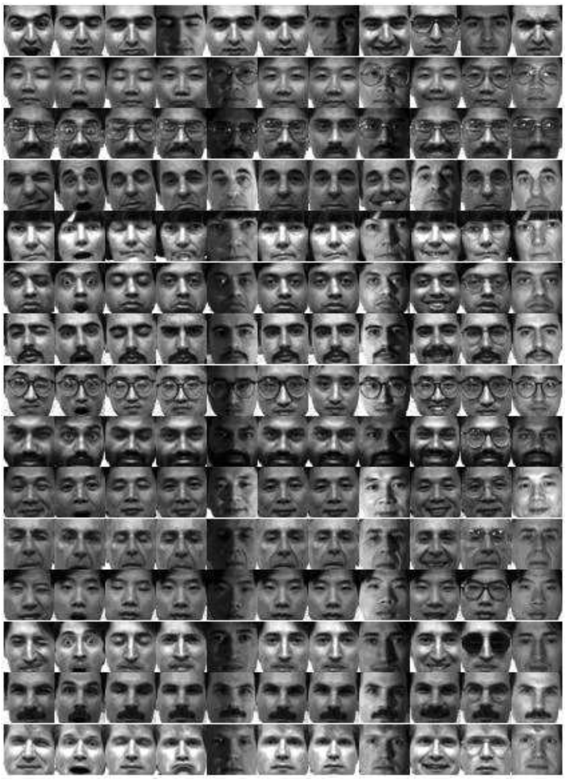


Figure 6.17: Yale database

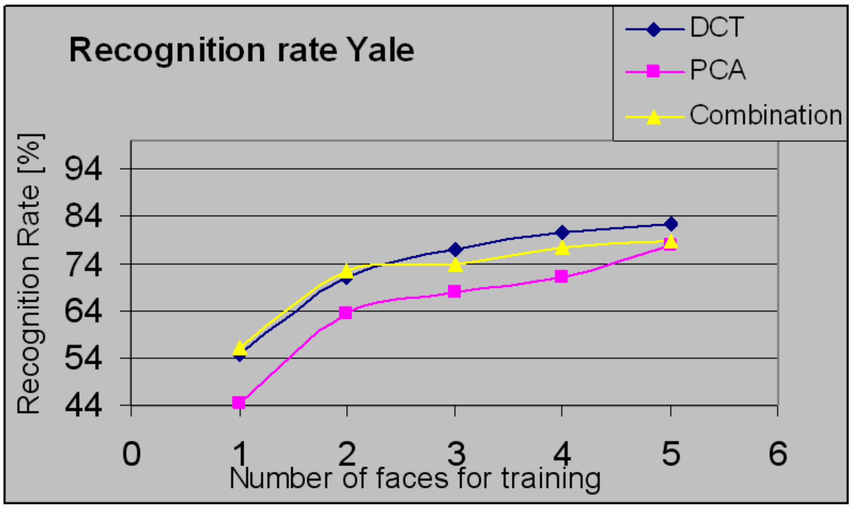


Figure 6.18: Results for Yale database

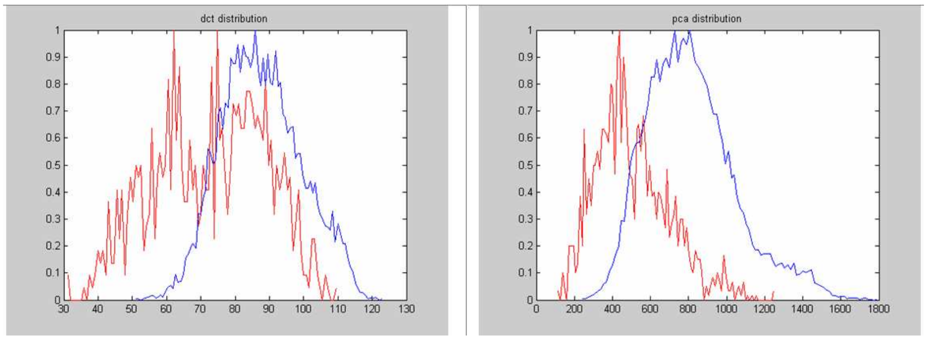


Figure 6.19: Distance distributions for Yale database

6.3 Conclusions after Initial Testing

The initial tests were performed using a database of images which has properties representative of a typical consumer collection of images. This database was chosen in order to design and test the retrieval module.

A first set of tests were then performed to determine the performance of the first basic prototype of our Person Recognition system. These were carried out as follows:

Firstly, the face recognition module has been tested starting with a standard approach, using different combinations of the number of DCT coefficients for the classification stage.

Secondly, the well known PCA approach was tested using different configurations of extracted PCA coefficients. As the PCA method is very sensitive to small variations inside the face region, wavelet decomposition was used prior to analysis in order to filter the image of noise and small variations, such as face orientation or face expression.

It is difficult to rely only on face recognition when dealing with consumer images due to large variations present in the images, so additional information was used for classification, particularly information extracted from regions surrounding the main face region. These regions are automatically defined from the face region using predetermined geometrical and size constraints. These were deduced from statistical analysis of several consumer image collections. Colour information is extracted from these regions using techniques such as the colour histogram or correlogram and our tests demonstrated the improvement brought about by using this additional information.

A second set of detailed tests were then performed to see how the face recognition module performs on more conventional, standardized face recognition databases. Four such standardized collections were chosen: BioID, Achermann, Yale and UMIST. The recognition rates depend on the variations present inside the collections and also on how many images are used to train the model for each person.

Based on these initial tests an optimal configuration was determined for implementing the final retrieval system. This configuration consists in using for face region classification the DCT approach combined with PCA approach applied on the 1 level LL sub-band from wavelet decomposition. Prior to recognition, the face regions are normalized to 32x32 pixels and transformed to grey scale images. For robustness to face orientation, the mirror face image is also used in the classification stage. The information extracted from peripheral regions is based on banded colour autocorrelogram.

With regard to the face recognition components we remark that the DCT approach, when used on its own, actually appears to perform better than the PCA or combined DCT/PCA techniques especially when illumination variations are very low in the database. The combined DCT/PCA technique shows an improvement especially when the number of images per person used in the training stage is small.

The distance distributions intra and inter class can be modeled as Gaussian distributions in most cases and are well separated when the recognition rates are high.

It seems that the face recognition module performs quite well on standard face recognition databases, but is affected by the extent of variations in facial regions that are present in the image database. Possible solutions to normalize these variations in order to increase the robustness of the module are presented in the next chapter. We shall also present the results of additional advanced testing of the system.

Chapter

7 Compensating for Variations in Pose and Illumination in Face Recognition

Although our initial results presented in **Chapter 6** are promising, they can still be improved; many of the failed recognition cases are due to the wide variations in pose and illumination that affect the face region.

This chapter presents several techniques that can be used to refine the data presented to the face recognition module described in earlier chapters. These techniques enable us to improve the robustness and accuracy of the face recognition module to illumination and pose variations.

It should be noted that some refinements were already described in the previous chapter and that, due to time pressures, not all of the described refinements have been incorporated in the application software described in the next chapter, and provided with this thesis.

A summary of refinements we already presented includes:

- applying wavelet decomposition prior to PCA analysis
- combining PCA/DCT distances for final classification
- using also the mirror image in the classification stage for pose robustness
- using additional (peripheral region) information for retrieval.

In order to improve the robustness of the system to different variations we have to minimize them. The most important factors that affect the robustness of face recognition as discussed in **Chapter 3** are represented by variations in illumination and variations in pose, or face orientation.

This chapter is organized in two sections: the first section presents solutions for normalization of the illumination component in the image and the second section present a possible solution to normalize the orientation of the face using Active Appearance Modeling techniques (AAM) and was undertaken as joint research with a fellow PhD researcher, Mr. Mircea Ionita.

The last section of this chapter is dedicated to conclusions on the effectiveness of applying these additional refinements to the face recognition module.

7.1 Illumination Normalization

As mentioned in **Chapter 3** and in the testing section, one of the preprocessing stages for the face regions is histogram equalization which is known to reduce the impact of variations in illumination on face recognition.

As previously mentioned we can deal with the illumination variations in two ways: we can use for classification features that are not affected by changes in illumination or we can minimize the variations by using an illumination normalization technique. Features that are not affected by the illumination conditions are geometrical parameters extracted from the face region: like distances between different corresponding points (**Chapter 3** gives a more detailed description of this approach).

The most common method for normalizing the illumination inside an image is to perform an histogram equalization, method described in detail in **Chapters 3** and **4**.

It is a known fact that the histogram equalization has limited applicability and will not have a huge impact on robustness when the variations in illuminations are too large.

Other variants of histogram equalization were developed like the Adaptive Histogram Equalization which in order to compute the new value for each pixel performs histogram equalization on small blocks having the given pixel in the center. This approach has the disadvantage that isolated peak values of the histogram inside the small blocks could affect the overall aspect of the resulted image.

Another approach that eliminates this disadvantage is CLAHE (Contrast Limited Adaptive Histogram Equalization) [88], and it has been proved to perform better than histogram equalization in applications like enhancing medical images [87] so we decided to test it.

7.1.1 CLAHE Algorithm Description

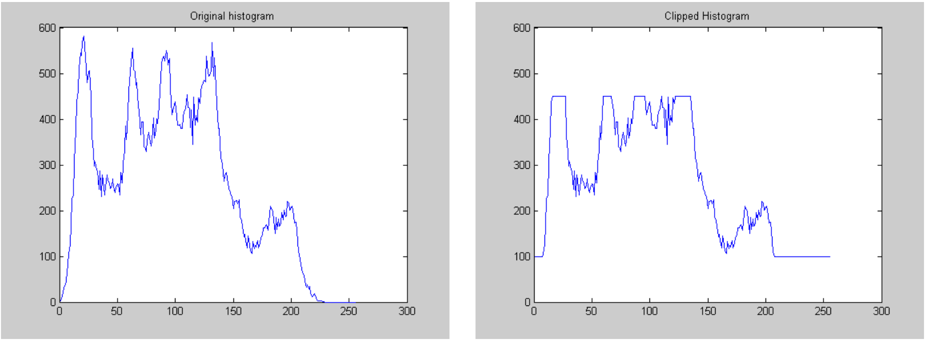
The stages of the CLAHE algorithm can be deduced from its name:

- Histogram equalization is performed on small blocks in the image like normal adaptive histogram equalization.

- In order to limit the noise that appears in homogeneous regions (regions with many pixels of the same value, where the histogram can have a high peak) the contrast of the final image is limited by clipping the original histogram to a superior and inferior limit.

- Using the clipped histogram the new values for each pixel representing the center of the small blocks are computed.

An example of clipping histogram is given in Figure 7.1



(a)

(b)

Figure 7.1: Clipping the original histogram

An example of applying CLAHE normalization is given in the next figure. It will be noticed that the image normalized using CLAHE compared with the one normalized using histogram equalization has a noticeably more even illumination on both sides whereas the original image and the one obtained using HE have uneven illumination on the right-hand and left-hand sides. As this form of *unbalanced* or *uneven* illumination is the source of many failed recognitions in our experiments it was decided to explore the potential of CLAHE as a preprocessing filter for our main face recognition module.

We tested the algorithm on two of the databases used to initially test the face recognition module: the Achermann and Yale databases.

The testing procedure employed was the same as that used in **Chapter 6** for testing the main face recognition module. Thus we used:

1. a combination of DCT/PCA feature distance

2. nearest neighborhood classification approach - only the first score returned was considered.

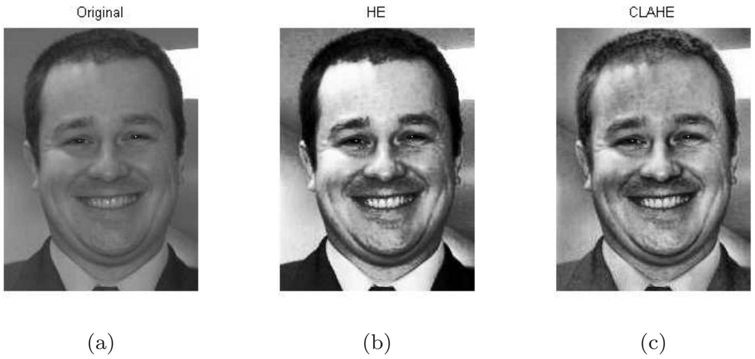


Figure 7.2: Example of applying CLAHE normalization

The parameters for CLAHE normalization were set to: 8x8 pixels block processing, and the limits for the histogram clipping are inferior limit 5 and superior 50 using 64 bins on 256 gray levels. These values were deducted from visual inspection of the resulted images and also from practical considerations. More tests need to be performed in order to obtain the optimal values.

7.1.2 Achermann Database

The Acherman database has limited illumination variations so we test the normalization technique using this database initially to see how it influences the results. The original and normalized Achermann databases are represented in the next figure.

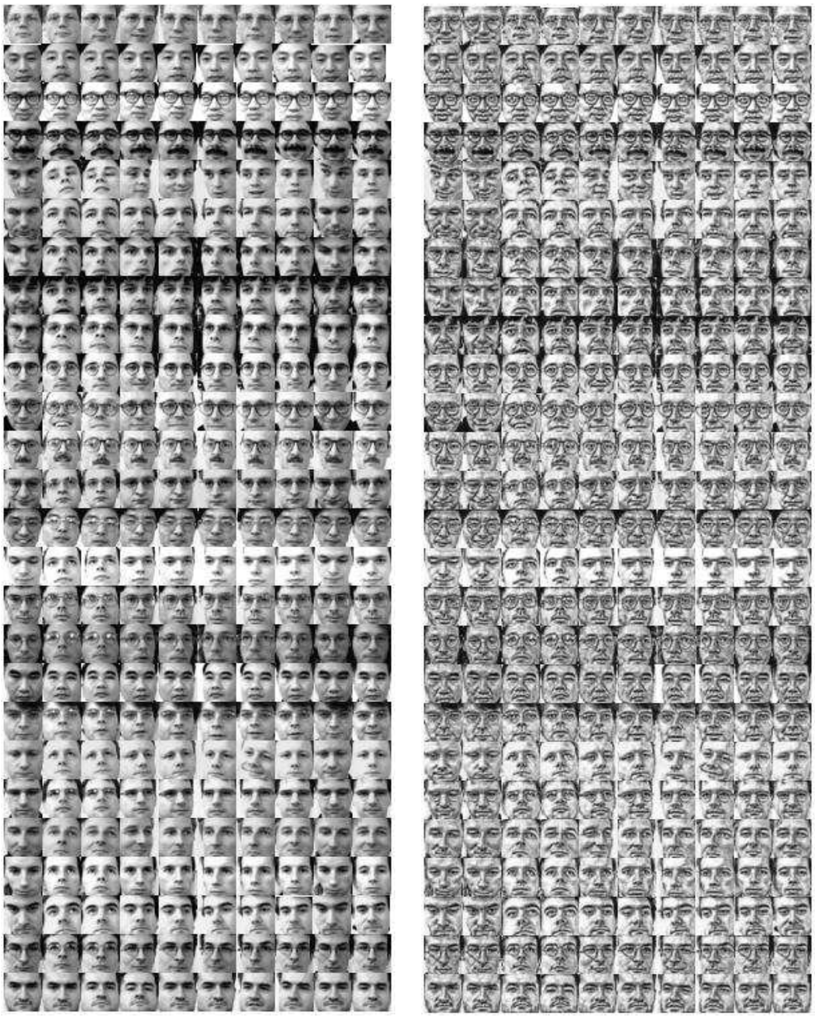
The recognition rates obtained are given in the next table.

Table 7.1: Recognition rates for Achermann database after CLAHE

DCT (%)	PCA(%)	Combination (%)	No faces train/test
67.09	54.7	66.24	1/9
73.56	57.21	70.19	2/8
80.77	70.33	81.87	3/7
83.33	75.64	82.69	4/6
90	81.54	88.64	5/5

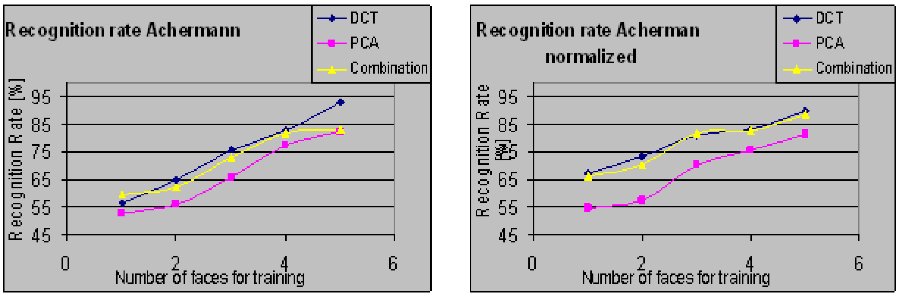
Graphical representations of the recognition rates for the original and normalized database are given in Figure 7.4.

It can be observed that in most cases the CLAHE illumination normalization improves the recognition rate.



(a) (b)

Figure 7.3: Achermann database a) original b) CLAHE normalized

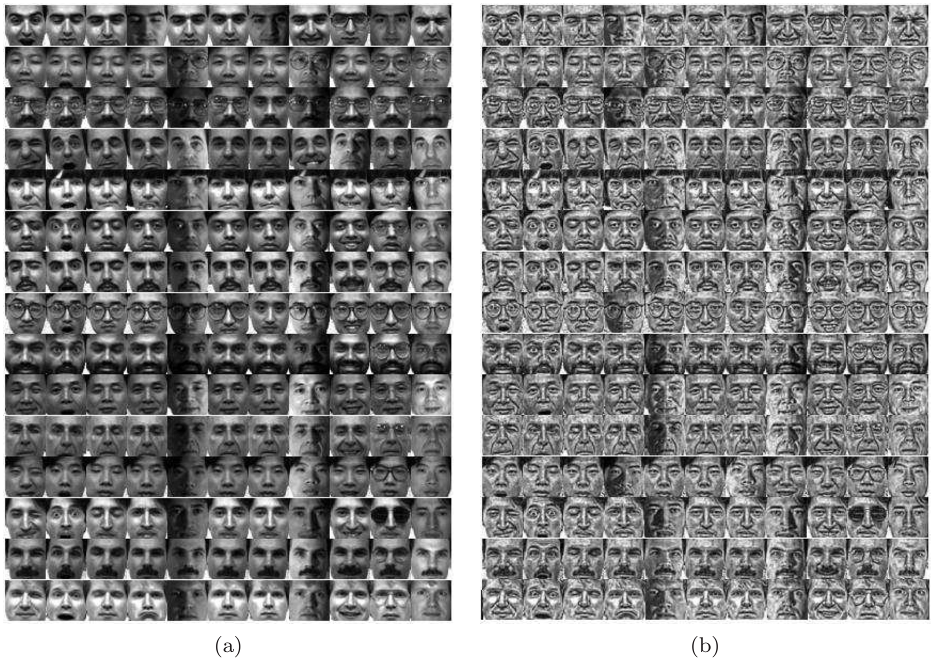


(a) (b)

Figure 7.4: Results for Achermann database a) original and b) normalized

7.1.3 Yale Database

The Yale database presents very large variations of illumination so it is the perfect collection to test the improvement of the normalization technique. The same test was performed on the normalize Yale database which is represented in Figure 7.5.



(a)

(b)

Figure 7.5: Yale database a) original b) CLAHE normalized

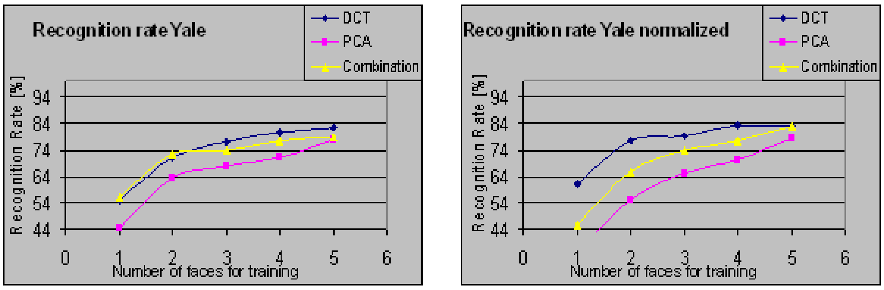
The recognition rates are given in the next table.

Table 7.2: Recognition rates for Achermann database after CLAHE

DCT (%)	PCA(%)	Combination (%)	No faces train/test
61.33	35	46	1/10
77.41	55.19	65.56	2/9
79.17	65.42	73.75	3/8
83.33	70	77.62	4/7
82.78	78.33	82.78	5/6

For comparison, graphical representations of the recognition rates for the original and normalized database are given in Figure 7.6.

It can be noticed that CLAHE normalization leads to significant improvements in DCT recognition rates when less than 5 faces are used for training. However there are not noticeable improvements for the PCA recognition technique. Overall, the combination of the two methods works better when using



(a)

(b)

Figure 7.6: Results for Yale database a) original and b) normalized

CLAHE normalized images apart from the cases when a single training image is used. In practice there will almost always be a requirement for at least two training faces so this case is probably not so important in practical applications.

In [27] we also tested CLAHE normalization on another face recognition algorithm studied by another fellow researcher, Ms Claudia Iancu, namely the Hidden Markov Models technique. The recognition rates in the case of HMM models using CLAHE normalization are noticeably superior to the results obtained without normalization.

7.2 Pose Normalization

Pose normalization was discussed in **Chapter 3** but most algorithms imply the use of a manual technique for straightening the face to a certain position by marking feature points such as the positions of the eyes or mouth.

If we want to keep the retrieval system fully automated we have to use an automatic pose normalization method which is non-trivial. Fortunately some parallel work within my research group provided us with an opportunity to apply a working and automated pose-normalization approach. It should be noted that this work was preliminary and that the topic of automated pose normalization within consumer image collections could form the basis of a PhD research topic in its own right.

In our research group the Active Appearance Models are studied for automatic face modeling [28] by a fellow researcher, Mr. Mircea Ionita. Our initial idea was to use the AAM to generate virtual faces with different poses starting from a single image and to use these virtual faces in the recognition process along with the original and mirror face. A second idea was to use the AAM modeling technique in order to normalize (i.e. ”straighten up”) the orientation of each face in both training and recognition stage to have the same pose.

7.2.1 AAM Description

The AAMs are based on statistical analysis of the variations of the shape of the face, along with the variations of the texture of the face region in order to build a statistical model which combines the information from both domains. The statistical analysis of the shape and texture is based on principal component analysis described in details in **Chapter 3**. The model consists of the parameters from the PCA algorithm eiegenvalues and eigenvectors from both shape and texture analysis.

The shape is represented as a vector of concatenated x and y coordinates of the landmark points. PCA analysis is applied on shape vectors as described in **Chapter 3**, so each shape in the collection can be expressed as:

$$s = \bar{s} + P_s b_s \tag{7.1}$$

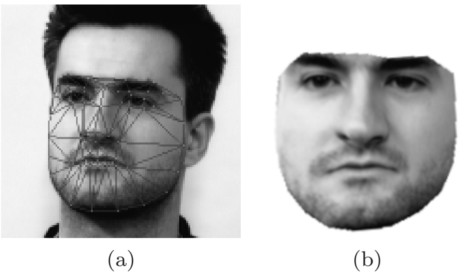
where $\bar{s}$ is the mean shape, P_s are the orthogonal modes of variation - eigenvectors, and b_s are the shape model parameters - PCA coefficients.

Texture vectors are built by warping each face patch into the mean (reference) shape - based on a triangulated mesh - and sampling the grey-level intensity values. Applying PCA once more on the texture coefficients we can describe each face texture as:

$$g = \bar{g} + P_g b_g \tag{7.2}$$

where $\bar{g}$ is the mean texture vector, P_g are the orthogonal modes of variation - eiegenvectors and b_g the texture parameters - PCA coefficients.

An example of applying the model on a face is given in Figure 7.7



(a) (b)

Figure 7.7: Converged model mesh and the pose-normalized face patch.

A combined model of appearance is further built by merging the two sets of parameters, subsequent to a suitable weighting operation. A third PCA analysis can also be applied to take into account the correlation that exists between shape and texture for a specific face image dataset.

Once the model is build for a collection of images, it can be fit to any new test image using the AAM optimization algorithm. The fitting parameters to be found, which best match the model to the test face, are the intrinsic parameters of the model (the PC coefficients of the shape model) and the set of parameters that localize the face patch inside the image (translation, scale and 2D rotation parameters). In order to find these parameters a regression approach can be used, which updates the model parameters based on the residual image error between the original images and the modeled one. This method is called analysis-through-synthesis.

Theoretically the AAM model should fit onto the face in the image independent of the initialization parameters. But this random initialization can be very time consuming and also could cause bad/incomplete or non fitting situations. We can use extra information from the face detection module in order to initialize the model. This information regards the relative position of the face and also the size of the face. Using this information the time needed to fit the model to a face is drastically reduced and also the fitting rate is improved.

Once the shape model is fitted to the image, using a warping technique we can change the orientation of the test image into any orientation desired.

This method is influenced by fitting errors when the test image is different (regarding illumination, pose and appearance) from the images that were used to build the model.

Our first idea to use the AAM technique in face recognition was to generate virtual faces with different orientations and to use these in the classification stage along with the original and mirrored face. The final distance between a face in the database and a test face will be the minimum distance between the original and mirrored test face, and a set of artificially generated faces at different pose angles generated using the main training face. Using this algorithm the fitting errors should not negatively influence the recognition rate because if the faces are falsely (visually incorrect) generated the recognition module should return higher distances when compared with the distances between any original faces.

We perform tests of the AAM approach using the Acherman database which presents the same person with different face orientations. The face pose having the main variation in the collection makes the Acherman database suitable for testing pose normalization techniques.

When testing the first idea of generating multiple faces with different orientations from the test faces the results did not show any improvement at all.

We generated for each test face 9 artificial faces with different orientation and used them in the recognition process along with the original image. A representation of such 9 artificial faces is given in Figure 7.8.

One possible explanation for obtaining the same results can be that the

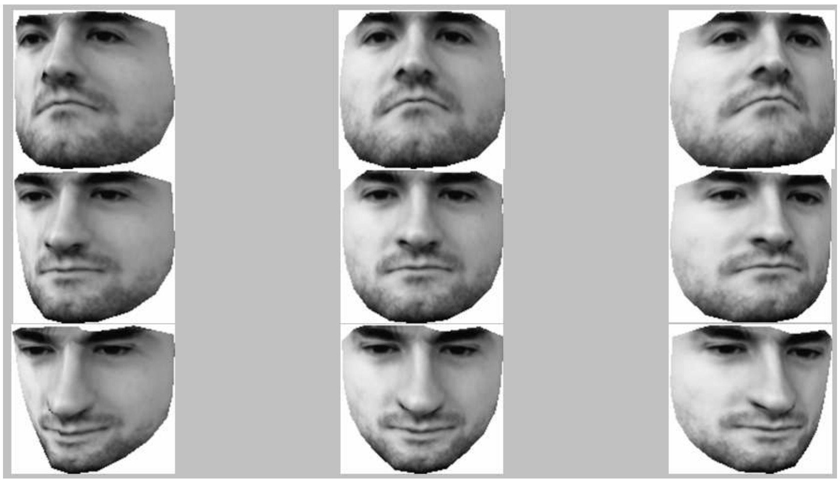


Figure 7.8: Nine generated faces with different orientation using a random face

artificially generated faces have a different appearance from the unmodified trained faces so the recognition module didn't take them into consideration. This is because the distance between a face and any of the original faces is smaller than any distance between a face and any artificially generated faces.

The next idea was to normalize the orientation of each face in both training and testing procedures in order to have a frontal orientation for each face for optimal recognition.

For this second series of tests we again used the Acherman database described in the previous chapter. The face images were manually annotated using 62 landmark points and the PCA analysis was designed to retain 95% of the shape variation and grey-levels variation, resulting in 14 shape parameters and 27 texture parameters. Please refer to [28] for a more detailed description.

The model was trained using 12 individuals from the database, each with 5 samples as per the example case illustrated in Figure 7.9. Note that this number of training cases selected from a database of the size of Acherman was considered sufficient for the purposes of this initial experiment.

The model was applied on the entire face database, which consist of 300 images of 30 individuals in 10 different poses. Some examples of normalized image are given in Figure 7.10. It can be noted that not all of the images are normalized correctly due to bad fitting. This is not necessarily a bad thing for recognition as we noticed even if the normalized images are not visually correct they are matched together by the recognition module.

The recognition rates obtained using the pose normalization technique are given in the next table.

It can be noted from the graphical representation that the recognition rate

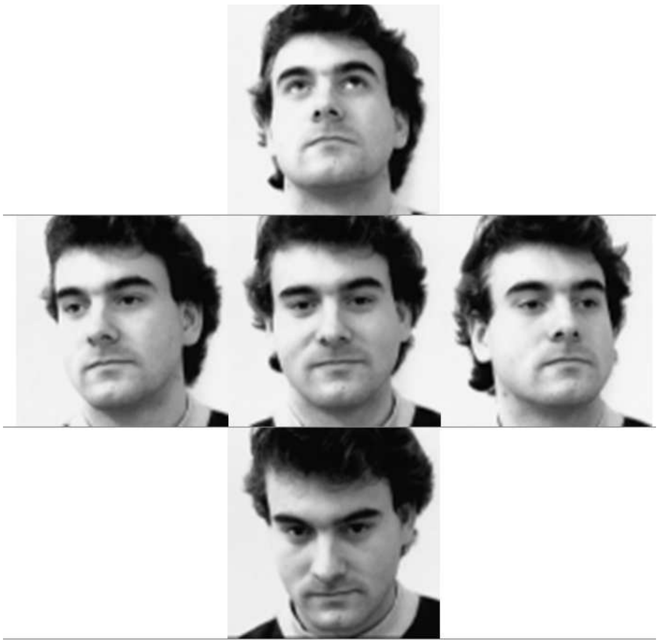
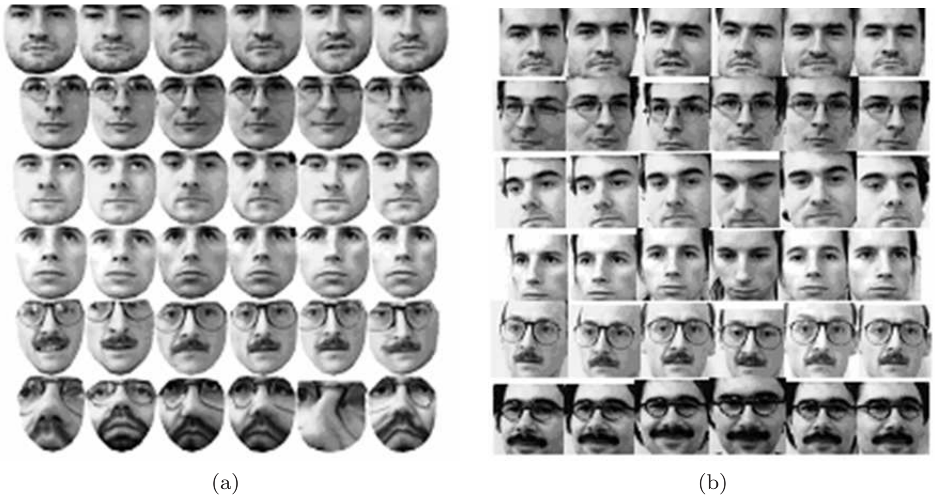


Figure 7.9: Poses used to build the model.



(a)

(b)

Figure 7.10: Examples of a) normalized and b) original images

Table 7.3: Recognition rates for Achermann database after CLAHE

Original Images (%)	Normalized images(%)	No faces train/test
59.82	72.96	1/9
62.5	79.17	1/9
73.2	85.7	3/7
82.1	91.67	4/6
82.93	94.1	5/5

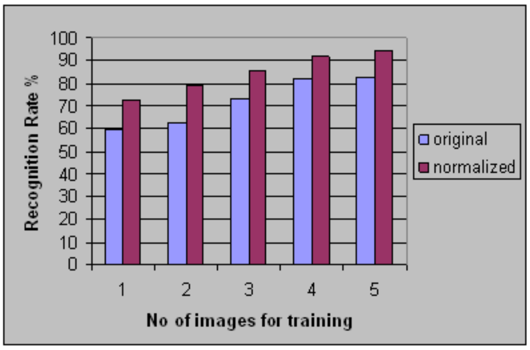


Figure 7.11: Recognition Rates using original and normalized faces

is significantly higher for all the cases when using the AAM normalisation technique prior to feature extraction, which demonstrates that employing such a pose normalisation technique will significantly improve the robustness of the recognition system to variations in facial pose. The fact that the recognition rate increases also with the number of images in the training set is due to the presence of other types of variations in the image collection such as variations in illumination and facial expression which cannot be normalized using this proposed model.

It is worth remarking that we have used conventional, grayscale AAM face models for this work and that recent enhancements to such face models to incorporate colour information and to dynamically recreate local texture and shape variations could provide additional improvements. We also remark that AAM models can be trained with specific model parameters which contain information directly relating to facial expressions, illumination variations and angular rotation about both vertical and horizontal axes. This suggests that such models can provide much useful information about facial regions and the use of such advanced AAM models in characterizing people in images could form the basis for an interesting future research study.

7.3 Conclusions

This chapter was dedicated to methods of improving the robustness of the face recognition module to the most important variations that affect the accuracy of most face recognition systems, namely: variations in illumination and face orientation.

In order to minimize the effect of different illuminations the Contrast Limited Adaptive Histogram Equalization (CLAHE) normalization algorithm was tested. Results showed a slight improvement especially for DCT and combined approach. Initial testing using another algorithm for face recognition based on the use of Hidden Markov Models (HMMs) showed a high improvement which recommends the generic use of CLAHE for illumination normalization.

For face pose normalization we tested a statistical face modeling technique called Active Appearance Modelling (AAM). Using the AAM we can change the orientation of the face to any angle desired. We tested initially an approach based on generating multiple virtual faces using the test image with different orientations and using them in the classification stage along with its mirror image. As this approach didn't give good results we tested the normalization approach where the AAM were used to normalize the face orientation by straightening them to a frontal orientation. This approach gave good results suggesting that improvements of 5%+ could easily be achieved and serves to suggest that further research should be undertaken to explore the use of AAM models for pose normalization of facial regions.

Chapter

8 Practical Application Frameworks for Image Retrieval

This thesis, as described in the first chapter, is dedicated to finding efficient and automated methods for organizing and sorting consumer collections of images. In order to achieve that, we detect people in individual images and determine the similarities, or lack thereof, between the detected persons and other "known" people. Due to the nature of consumer images, classical face recognition techniques are not sufficient when used on their own. In previous chapters we have described how the data presented to classical face recognition techniques can be pre-filtered and how the outputs from several such techniques may be combined. The concept of using regions which are peripheral to the main face region has also been explained and practical implementations have been developed and experimental results presented.

In this chapter the techniques and methods described in these earlier chapters are brought together in practical application frameworks. Such embodiments of the core project technology were important because this research is partly sponsored by an industrial partner who requires tangible outputs that can be tested and verified. But they are also considered part of the core research as it was undertaken within the Consumer Electronics Research Group of NUI, Galway and a key mandate of that group is to test the practical utility and applicability of such technologies. Thus we describe next some of the practical application frameworks that have been implemented using the core outputs of this thesis.

The chapter begins with a general description of the architecture of common retrieval systems and continues with the particulars of the proposed retrieval system given in section 8.1.

The second part of the chapter is dedicated to three applications that use the retrieval module described in the previous sections: the first application described in section 8.2.1 was designed to demonstrate visually the functionality of the Person Recognizer, second, a desktop application, designed to be used by any user on his personal computer is described in section 8.2.2 and third, a web-based application designed to meet the needs of those users that are storing their pictures on remote dedicated image servers is described in section 8.2.3. The latter system was originally intended to demonstrate the concept of a remote server for mobile phone users [29], but it was subsequently adapted and integrated into a home network architecture [32].

8.1 Retrieval System Architecture

The classical definition of a retrieval application is extracting at any time the required information from a collection of data. The most common implementation of an automated retrieval system is that of "query by example" where, based on a given example of required data, the system searches an image collection for other similar data.

Retrieval application architectures can be divided into two main system components: (i) the training module, which analyzes the entire collection of data in order to extract features that will be used by (ii) the search, or retrieval module, which uses an example query to search through a collection for similar data patterns based on feature vectors determined during the training process.

A simplified generic architecture of the training process for a *Person Recognizer* system is given in Figure 8.1.

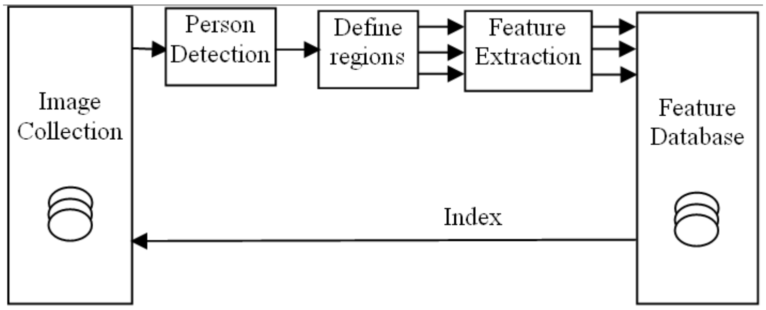


Figure 8.1: Training module architecture

The training stage consists in analyzing the entire collection of data, determining the basis vectors that will be used to project the data samples into the feature space and extracting reference pattern sets represented by the feature vectors which will be stored in the feature recognition database and will be used in the classification stage. More sophisticated embodiments may also perform a grouping phase during which reference patterns are gathered into distinct groups. However this approach does not always provide an overall improvement in recognition rates as discussed in ***Appendix A***.

A generic architecture of the search/retrieval module is given in Figure 8.2.

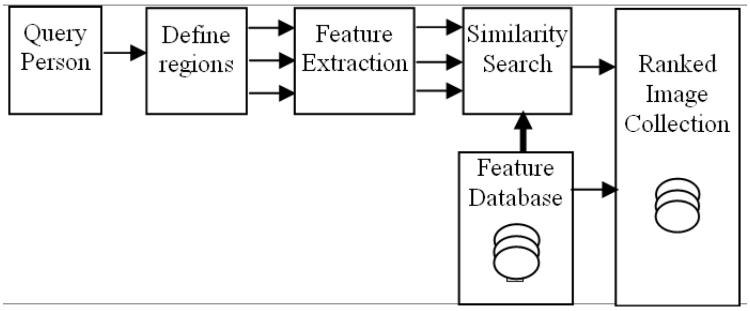


Figure 8.2: Searching module architecture

The search module uses the reference pattern database computed during the training phase and, based on a query example, it computes the similarity between the pattern of extracted features of the query example and all the reference feature patterns in the recognition database for that collection.

At the end of the module, the collection of images is ranked based on the similarity ranking of each known reference pattern(s) with the feature pattern extracted from the face region associated with the given query.

One example of a practical application of such a retrieval system is where a user keeps his image collection(s) on a home computer and wants to retrieve all images of a certain person, of whom he already has one or more example images.

This can be implemented as a desktop computer application which takes care automatically of training the images each time new pictures are added to an image collection, and when the user wants to retrieve images, he will be able to do that very simply and quickly just by clicking on a face region within an example image from the collection.

The detailed architecture of such a computer application is given in Figure 8.3. The core of such a system is the DLL library which takes care of both training and searching modules with all image processing algorithms implemented in the image analysis module. The features database or image collection data is stored on the desktop storage equipment, and the interface with the user is realized through the computer peripherals for both input and output (the results of the retrieval). This application can also work with collections of images which are not on the desktop, but are stored remotely on the network.

Alternatively the retrieval application could be implemented as a remote dedicated server which can be controlled by the user from a user interface implemented on the server and accessed by the user through a web browser interface. This variant is useful for network based services for mobile phones or for wireless home networks where mobile access to remote data and associated services is valued by users.

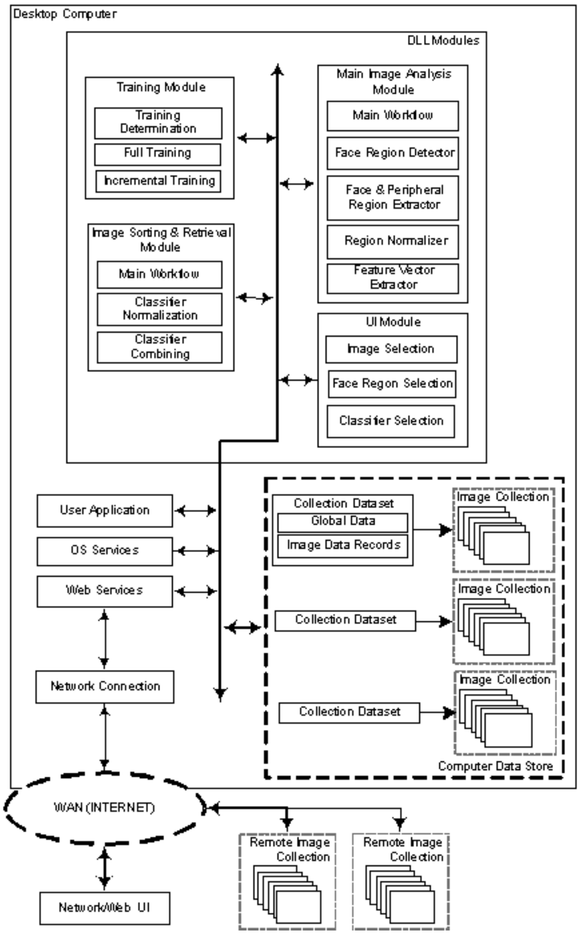


Figure 8.3: General architecture of the retrieval application

The core of the system is of course the main image analysis module and this is, in fact, used in both training and search/retrieval modules of the application. The workflow of this module is detailed in Figure 8.4. The module is in charge of detecting all persons in all images, analyze them and update the collection dataset with the features accordingly. This module is called from a higher level workflow and, in its normal mode of usage, takes as input a set of images which must be analyzed. Where at least one face region is detected, this module extracts and normalizes each detected face region and, where possible, any associated peripheral regions. Finally, it determines feature vector patterns for a plurality of face and non-face classifiers and records this extracted information in an image dataset record.

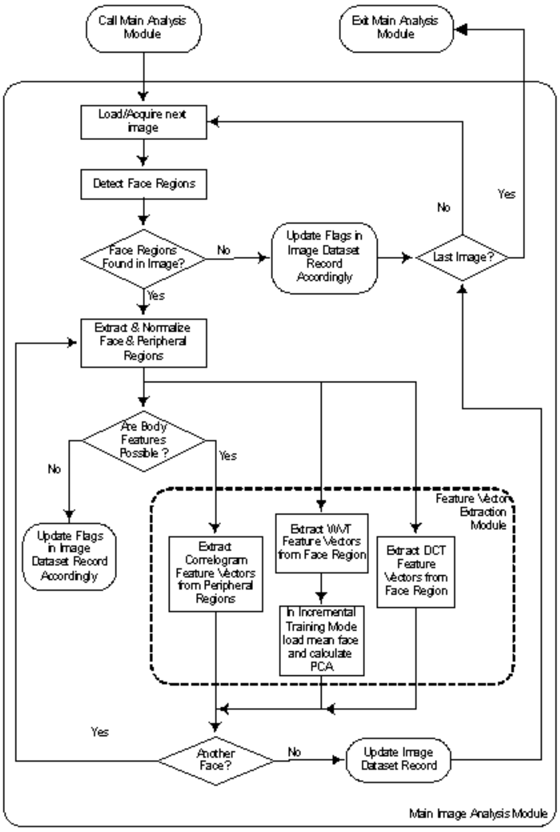

Call Main Analysis Module
Exit Main Analysis Module
Load/Acquire next image
Detect Face Regions
Face Regions Found in Image?
No
Yes
Update Flags in Image Dataset Record Accordingly
Last Image?
No
Yes
Extract & Normalize Face & Peripheral Regions
Are Body Features Possible ?
No
Yes
Update Flags in Image Dataset Record Accordingly
Feature Vector Extraction Module
Extract Correlogram Feature Vectors from Peripheral Regions
Extract WWT Feature Vectors from Face Region
In Incremental Training Mode load mean face and calculate PCA
Extract DCT Feature Vectors from Face Region
Yes
Another Face?
No
Update Image Dataset Record
Main Image Analysis Module

Figure 8.4: Main image analysis module

All modules were previously described in detail except the incremental training module regarding PCA analysis. This considers the property of PCA method to be data dependent, so every time the format of the image collection changes, the PCA data has to be recomputed. This aspect of the system will be discussed further in **Chapter 9** where a new approach which enables incremental training of image datasets for any PCA based techniques is presented.

The top level organization of the image collection dataset which contains these features is given in Figure 8.5.

The dataset contains all the features computed during training and also additional information regarding the properties of the image collection represented

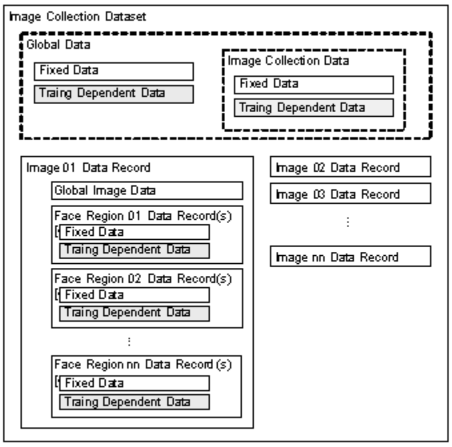


Figure 8.5: Image collection dataset

by global data in the diagram. This could include additional data from external sources such as image EXIF metadata or user annotations. Such data could advantageously be integrated with a more sophisticated application which would enable, for example, user annotation to confirm the identity of persons who are ”recognized” by the automated Person Recognizer system module.

The global collection data in the dataset records properties of the features extraction algorithms such as the number of DCT coefficients, number of PCA coefficients, normalization size and parameters of the banned autocorrelogram features extraction. This data is stable and will not change when the collection changes.

On the other hand, the training dependent data of the image collection dataset, such as the number of images in the collection, the number of persons in the collection, contains information that has to be updated every time the training collection changes. This training dependent data also contains PCA information that has to be updated when the collection changes, such as the mean face and the eigenvectors and eigenvalues from which feature vectors patterns are constructed. All this information describes the global properties of the image collection.

Each image in the collection has also assigned a dataset of features which is divided into fixed data and training dependent data. The *fixed data* contains information such as the number of faces present in the image, the DCT features of these face regions, colour information features from peripheral region and any additional information that will not change. The *training dependent* data is represented by the PCA feature vectors which must be recalculated for each image whenever the main image collection is re-trained.

It was previously mentioned that the training procedures for computing the

DCT and PCA features are different and the difference between them has an important influence in applying the methods in real life applications. These types of applications imply a dynamic procedure for analyzing the data samples. In our case the data samples are represented by the image collections. Usually the users are continuously changing their image collection by adding or deleting images or by merging or splitting collections of images. It is important to see how this behaviour will influence the way the images are analyzed.

DCT training is an absolute procedure, which is independent on the image collection used for training. This means that it doesn't take into consideration the properties of the image collection from which the data sample to be analyzed came. The basis vectors used to project the data are fixed. Once the features are computed they will remain unchanged even when the collection changes its format. For this reason the DCT approach is recommended for application where the data collection is changing.

On the other hand PCA training as described in detail in **Chapter 3** takes into consideration statistical properties of the training data collection. This makes PCA features relative and dependent to data collection. Ideally whenever the data collection changes the PCA algorithm has to be applied again disregarding the previous computed features. This is because the basis vectors used to project the data (the eigevectors) are changing every time the collection changes its format due to insertions/deletions. This makes PCA unsuitable for applications where data changes constantly like our application because it will take to much time/ memory to apply PCA every time new images are added to / deleted from the collection.

Fortunately we analyzed this problem and found a solution for updating the previous PCA features and computing the new basis vectors when new images are added or deleted from the collection or when two or more collections are merged into super collections. This is done by using the previous computed features and basis vectors. The method is presented in **Chapter 9**.

We mentioned that the idea of merging two or more collections is very important especially when working with consumer image collections because usually the consumers take a series of pictures (collection) with the camera and then transfer this collection onto the PC for storage and future retrieval where he keeps his entire collection of images. Thus is very important to combine the two collections without the need to fully retrain the new collection which in some cases for very big collection could take a very long time.

Also the way that the operating systems organize different data into folders and sub-folders which can be regarded as collections and sub collections may prove to be an advantage when the user wants to organize his image collection of the PC. The second application is illustrative for this aspect.

8.2 Applications Using the Retrieval System

This section will describe three applications that use the retrieval module described in the previous sections. The first application was designed to demonstrate visually the functionality of the Person Recognizer.

The second application is intended to be used on desktop computers by users in order to arrange their image collections and to retrieve pictures from the collections by giving examples to search.

The third application is a web based image retrieval system. Using a web server which is capable of storing and training the image collection, the user can upload their pictures on the server and using any java enabled application, for example a standard web browser, they can search for persons (one or more than one person) in the collection by providing examples, and can also download the wanted pictures.

An image collection in context of these applications is considered as all the images that are stored in a folder (directory) on the computer. Image subcollections are image sets stored in a sub-folder (sub-directory) of the main collection folder.

When training a collection in a given folder, all images from all sub-folders (sub-collections) are trained along with the images on the root of the given folder. It is possible to train a collection of images by using the data from its trained sub-collection. This type of training is called incremental training. Special attention has to be paid to data dependent analysis methods where this incremental training is not possible. In our case the PCA method is data dependent but in **Chapter 9** we present the theoretical derivation of a method of combining PCA data incrementally.

Also when newly trained sub-collections of images are added to a collection, the incremental training workflow can be called to update the collection's dataset of features.

8.2.1 Initial Application

The first application was designed for demo purposes for proving visually the functionality of the Person Recognizer. It is a basic QBE application designed and implemented in Matlab. All modules are implemented in Matlab language except the face detector which was implemented in C as a dynamic link library and an interface between the DLL library and the Matlab application was created.

Basically the application will search for all images stored in the same directory where the application is running. These images will become our image collection. The first stage like in any CBIR system is to train the image collection by simply pressing the red Train button in Figure 8.6. During training all faces are detected from each image, and for each face the face descriptors are computed (DCT, PCA) and where possible the surrounding regions color descriptors (banded autocorrelogram). At any moment for any reason the collection can be retrained by pressing the Train button.

After training six random images are displayed in six thumbnails in the application panel. By pressing on any of the thumbnail the selected image is displayed in the main panel with all detected faces surrounded by red rectangles like in Figure 8.7.

A new image from a different location on the computer can be loaded into the main panel by pressing the Load button and face detection will be applied as well for displaying the available faces.

The user can select for classification between face and body features or a combination between the two by using the radio buttons on the bottom side of the application panel.

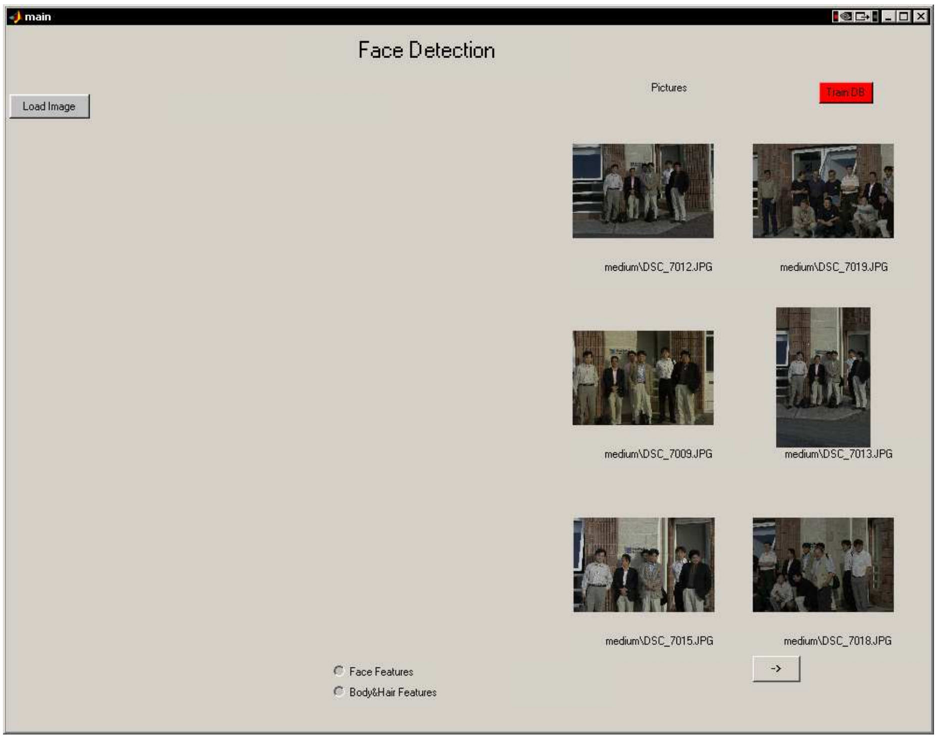


Figure 8.6: Initial screenshot for first application

For searching a person from the main image in the collection the user only has to click with the mouse inside the face of the person. If the user selects body features but for the selected person the body features can not be computed in the main image (because the person is at the margin of the image and peripheral regions can not be defined) than the body features radio button will be unselected, the option will be ignored and only face features will be used for classification.

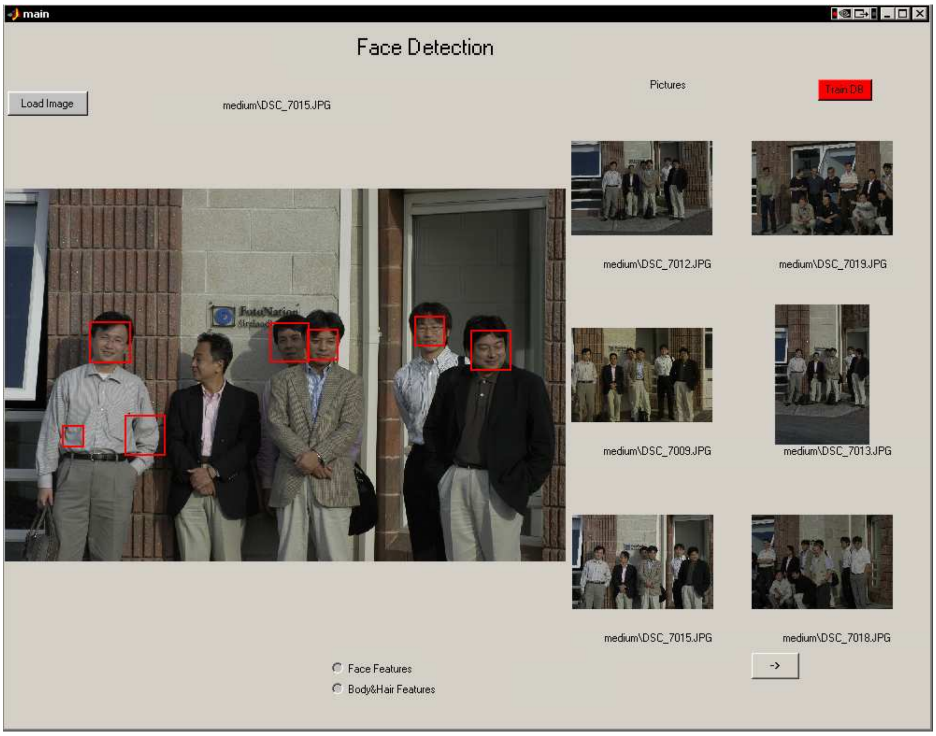


Figure 8.7: Representation of the detected faces

After the system finishes to search the trained collection all the images in the collection will be ranked accordingly to the similarity of the persons inside them with the given person. The order of the small thumbnails will be changed as well starting with the first ranked image. Buttons for browsing through all image collection are on the bottom of the thumbnails panel.

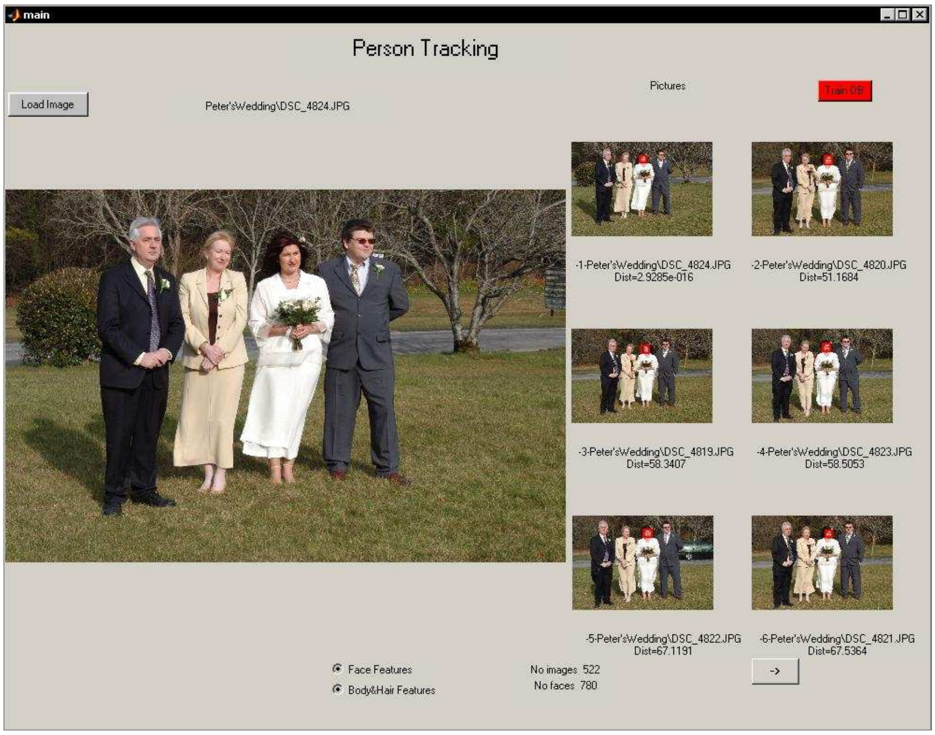
One example of the results after searching is given in the next figure using body and face features.

The user has also the possibility to search for multiple persons by keeping pressed the Ctrl key while selecting with the mouse the persons for searching.

One example of retrieving multiple person is given in Figure 8.9

8.2.2 Desktop Application

This application was implemented using C++ programming language for the retrieval library and Windows Template Library (WTL) for designing the Graph-



Peter'sWedding\DSC_4824.JPG

-1-Peter'sWedding\DSC_4824.JPG
Dist=2.9285e-016

-2-Peter'sWedding\DSC_4820.JPG
Dist=51.1684

-3-Peter'sWedding\DSC_4819.JPG
Dist=58.3407

-4-Peter'sWedding\DSC_4823.JPG
Dist=58.5053

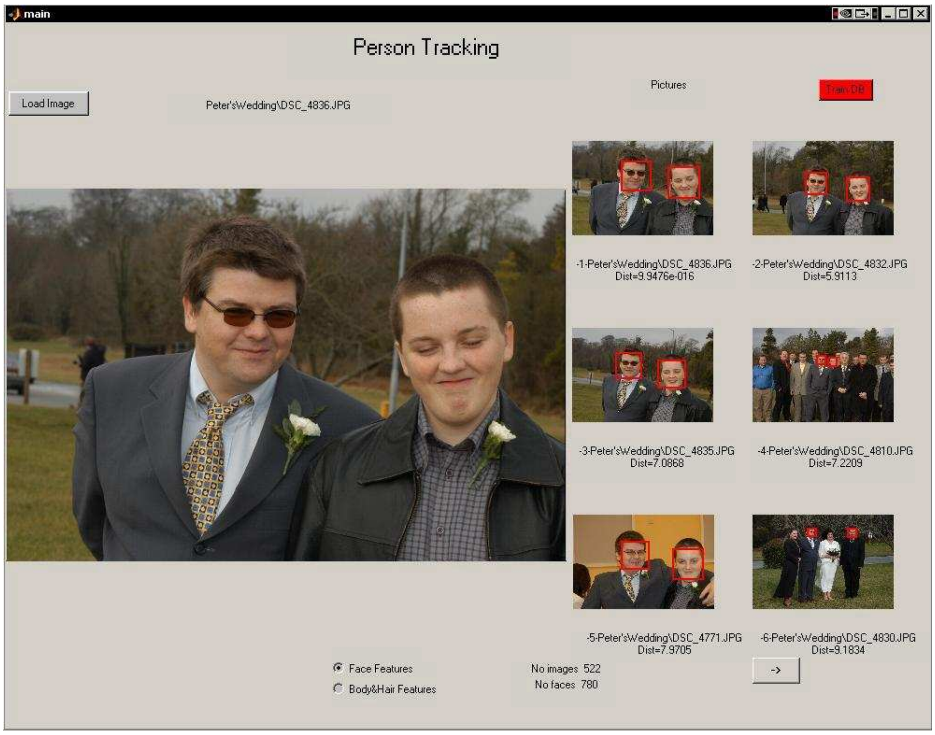
-5-Peter'sWedding\DSC_4822.JPG
Dist=67.1191

-6-Peter'sWedding\DSC_4821.JPG
Dist=67.5364

● Face Features
● Body&Hair Features

No images 522
No faces 780

Figure 8.8: Results of the retrieval



Peter'sWedding\DSC_4836.JPG

-1-Peter'sWedding\DSC_4836.JPG
Dist=9.9476e-016

-2-Peter'sWedding\DSC_4832.JPG
Dist=5.9113

-3-Peter'sWedding\DSC_4835.JPG
Dist=7.0868

-4-Peter'sWedding\DSC_4810.JPG
Dist=7.2209

-5-Peter'sWedding\DSC_4771.JPG
Dist=7.9705

-6-Peter'sWedding\DSC_4830.JPG
Dist=9.1834

● Face Features
○ Body&Hair Features

No images 522
No faces 780

Figure 8.9: Retrieving multiple persons

ical User Interface (GUI).

The interface is similar to the Windows Explorer interface, which is very familiar to many users. The collections of images are represented in the interface by the folders in the explore view (left tab). Each collection dataset is stored in two files in the root of the collection folder. These files contain binary representation of all features and also binary representation of the training dependent collection data.

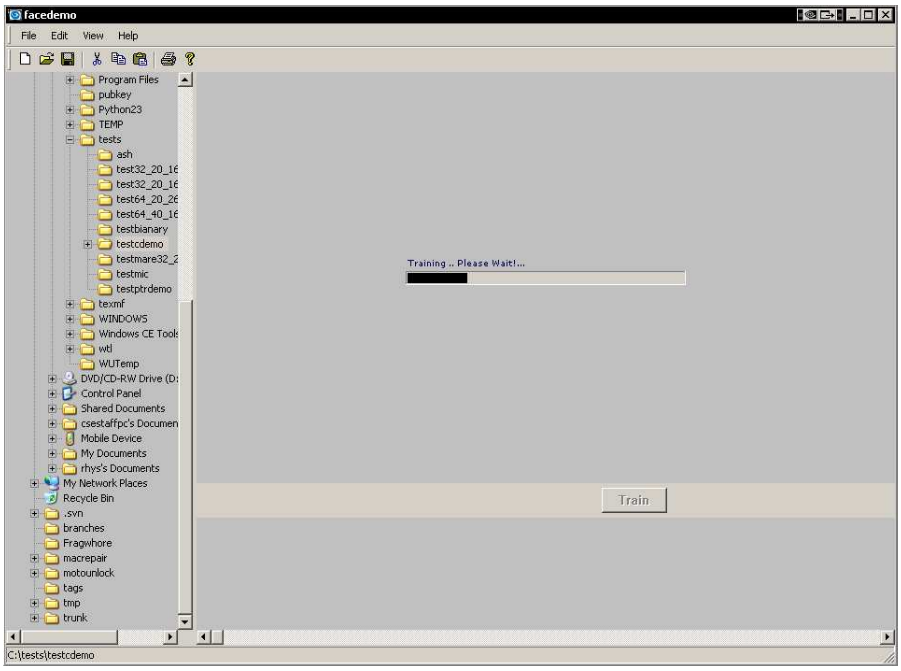
One of the files is associated with the collection specific data (e.g. the mean data sample used for PCA anlysis, the eigenvectors and eigevalues also computed during PCA analysis and also general information like the types of coefficients computed in the training stage (PCA, DCT), parameters that were used to compute these coefficients: the size of normalized images, the number of DCT coefficients, the number of PCA coefficients). This file is also used to check the compatibility between the training module and the searching module (e.g. we cannot search into a collection that was trained using a smaller number of coefficients or smaller normalized face regions than the parameters used by the current searching module).

The second file contains the dynamic training data (e.g. the number of images and faces in the collection and for every face in every image the features calculated during training). This file is also used to check if a collection is trained or not or if new images were added to the collection or deleted since the previous training.

The first stage of the application is to train the collections of images from the user's computer. This is done by choosing any folder from the computer using the left side explorer window from the GUI and pressing the Train button which is active if the folder was not already trained. A screenshot of the training stage is given in Figure 8.10.

The training procedure is multi-thread enabled so the user can choose another folder for training/searching even if the training procedure is not over for the current folder. A progress bar is present in the application in order to see the status of the application.

The training stage includes detecting all the faces present in all images and for each face computing DCT and PCA features as described in the previous sections and when possible, defining body&hair regions and computing colour correlogram features. All the features are stored in one file which is associated to the given collection. Once the user selects a trained folder, the pictures in the folder are displayed on the main panel using the same appearance as the "FilmStrip" view present in the Windows XP environment when selecting folders with images in it. The training flag is set by searching the feature file in the folder and checking to see if it contains the features for every picture in the



Training .. Please Wait!...

C:\tests\testcdemo

Figure 8.10: Training screenshot for the desktop application

collection.

By selecting a thumbnail from the lower part of the GUI the full image will be displayed on the main panel of the application and all faces that were detected in the image are displayed using a blue rectangle. When the user presses a blue rectangle, the Find button will be activated and the blue rectangle is transformed into a red rectangle. The user has also the possibility to choose what features to use in the classification stage, using the radio buttons for ”face only” or ”face and body” options which will include peripheral region features as part of the similarity analysis.

Once the user has pressed the ”Find” button, the application will re-arrange the thumbnails according to the similarity of the faces present in each image with the face region selected by the user in the main image. A screen shot with the searching result of a query is given in Figure 8.11.

The user may also choose to search for a group of persons in the same time: by pressing the ”Shift” key when selecting the face rectangle, the system waits for one (or more) additional rectangles to be chosen and will search in the collection for images that contain at least two (or N) individuals, and rank them according to their similarity with the given rectangles. This multi-person search capability can produce very impressive search results on some image collections.

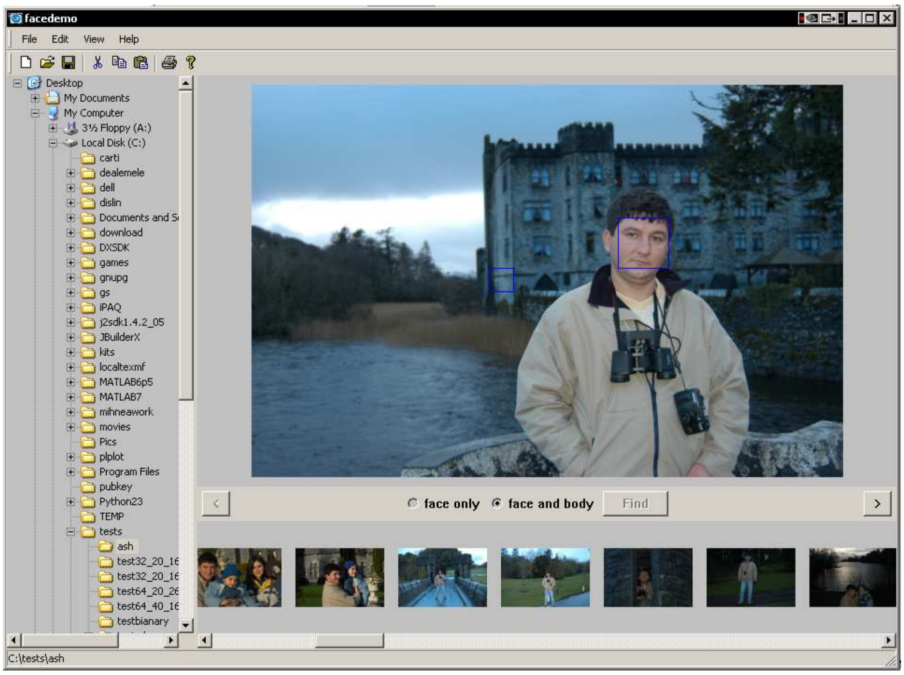


Figure 8.11: Search result screenshot for the desktop application

8.2.3 Web-based Application

This application was implemented in collaboration with my research colleague Rhys Mulryan, from the Consumer Electronic Research Group who is undertaking M.Eng.Sc. studies in the same field of research.

In the last few years, web servers dedicated to storing, managing, processing and printing digital images have become very popular with users. These types of servers are named photoservers. However, most of these servers give a user limited access to their images. Even if a user has full access to all images on the server, if he is interested in some pictures he will have to search those in a manually or semiautomatic manner using tags. For large image collections, it is very difficult to manually find the images that are of interest. If the server is using tags that describe the content of the images, these tags have to be manually created which is a very tedious process for the user.

The *Person Recognizer* technology described in this thesis offers an alternative method of organizing and sorting the images on the server in a fully automatic manner which is very intuitive for users.

In a related, yet distinct field of application this server-side imaging architecture can be easily integrated with a multimedia home network architecture, and research has been undertaken in this direction by the Consumer Electronic Research Group [32, 78]. In this home network architecture the training module can be implemented in any computing capable device. The training module has access to any storage equipment on the network, where collections of images can be stored, and also has access to remote collections on the Web.

For such applications a remote, Web based user interface can be implemented and the results of the sorting module can be displayed for the user on any display capable device connected to the home network or to a mobile Internet connection. The same architecture can be used to offer additional web services for mobile devices. For instance, a mobile device can be used to upload and retrieve images from a remote dedicated photoserver. Please see [29] for more details of our research in this field. This work has also been undertaken in the CERG research group, leveraging the core system modules of the Person Recognizer.

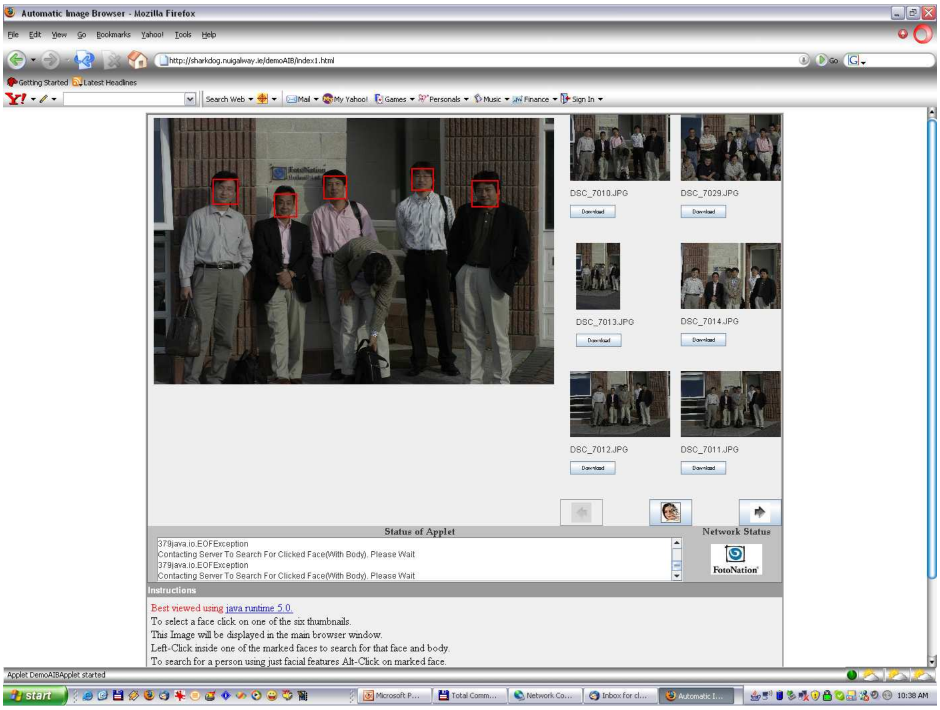
The resulting application framework is designed to meet the needs of those users that are using a dedicated network server for managing images. By implementing our Person Recognizer modules on the server we can store and train the user image collection and using a browsing web client the user can search for images given query examples. The users can upload their images on the server using the same client and can also download the images if required after searching.

The application was implemented using Java language for both server side and application side. The server side consists of a Java servlet implemented on the TomCat Java server application, and the client side is a simple Java applet inserted into a web page. All modules of the Person Recognizer were ported to Java language for this application.

The graphical interface of the applet is intuitive enough to be used by someone with moderate computer experience encountering it for the first time. The main panel is on the left side and the thumbnails panel is on the right side, and it is capable of displaying 6 thumbnails at one time. The reader will remark the similarity to our first desktop application.

When the user clicks inside a thumbnail that image is displayed on the main panel and all faces detected are indicated by a red rectangular border. If the user wants to search one of the persons inside the red rectangles, he just has to click on it. The image and the coordinates of the rectangles are then sent to the server-side application which searches the collection for the given face and returns a ranked list of thumbnail images to the applet. This list is ranked according to the similarity of the face region(s) in each image with the selected face region in the main image. Even though the thumbnail panel only displays 6 images the user can navigate the entire collection using ”forward” and ”back” navigation buttons.

For simplicity a training procedure is invoked every time a new image is


Automatic Image Browser - Mozilla Firefox
File Edit View Go Bookmarks Yahoo! Tools Help
http://sharkdog.nuigalway.ie/demoAIB/index1.html
Getting Started Latest Headlines
Search Web Mail My Yahoo! Games Personals Music Finance Sign In
DSC_7010.JPG
DSC_7029.JPG
DSC_7013.JPG
DSC_7014.JPG
DSC_7012.JPG
DSC_7011.JPG
Download
Network Status
FotoNation
Applet DemoAIBApplet started
start
Microsoft P...
Total Comm...
Network Co...
Inbox for cl...
Automatic I...
10:38 AM

Status of Applet

```
379java.io.EOFException
Contacting Server To Search For Clicked Face(With Body). Please Wait
379java.io.EOFException
Contacting Server To Search For Clicked Face(With Body). Please Wait
```

Instructions

Best viewed using java runtime 5.0.
To select a face click on one of the six thumbnails.
This Image will be displayed in the main browser window.
Left-Click inside one of the marked faces to search for that face and body.
To search for a person using just facial features Alt-Click on marked face.

Figure 8.12: Web-based application

uploaded to the server. As before, this involves detecting all the faces in the collection and computing features for each detected face region and, where possible, for body&hair regions.

The default searching option is to use only face region searching. When the user wants to use also body&hair features they have to press ”Ctrl” key along with a click inside the desired rectangle. For searching multiple persons in the same time the user have to keep the ”Shift” key pressed when selecting rectangles, similar to the desktop application.

8.3 Conclusions

This chapter demonstrated the potential use of the research presented in this thesis. This potential relates to a variety of different applications where the Person Recognizer modules can be integrated for user interaction and also relates to the flexibility of basic toolset of the Recognizer which can be adapted for multiple architectures.

The main applicability of the Recognizer is for organizing digital image collections. These type of applications can use the Person Recognizer with additional automatic tools for image classification (classical Content-based Image Retrieval CBIR systems) or manual techniques currently used for organizing/retrieving images (manual applying tags to images). Many ideas can be developed on how can the user annotate, or at least confirm the automatically generated annotations presented by the Person Recognizer but this ultimately depends on how the final application for organizing the images is designed.

Another important issue that was mentioned in this chapter involves the training module and how can we minimize the time needed to train the user image collection taking into consideration how the user acquires multiple small collection of images (subcollections) and transfer them to his main storage component where the main collection of images (or the supercollection) is kept. This issue is further addressed in the next chapter where a possible solution is presented.

Chapter

9

Combining PCA collections of data

In **Chapter 8** it was indicated that it is possible to combine training data from two or more image collections to determine a common set of basis vectors without a need to retrain from the original face data. This technique offers a significant saving in computational effort and based on extensive testing it appears to be quite robust and offers high repeatability in the context of eigenface analysis.

In this chapter we develop a formal theoretical explanation of this method and demonstrate that it should be broadly applicable to other fields of application where PCA based analysis is employed.

We begin by outlining how consumer image collections will naturally comprise multiple subcollections. Now to provide useful end-user tools it is desirable to be able to combine these into larger units, or *supercollections* on an ad-hoc basis. This, in turn leads to consideration of the retraining problem for consumer image collections. We then review the theoretical basis of *principle component analysis* as it is applied to *eigenface* analysis of face regions.

Our approach to combine the training data from multiple pretrained datasets is then presented and discussed. From our review of the literature it would appear that this approach is novel, at least within the context of face data. We present next the results of some experiments with image collections drawn from very different sources and demonstrate the practical results of applying our technique. Finally we discuss the benefits of this approach for the field of application which is the subject of this thesis and suggest some other fields of application where this application technique might also prove beneficial.

9.1 The Retraining Problem for Image Collections

A typical consumer gathers images in semi-organized clusters. Often these will be associated with a single event, or perhaps with a defined period of time, such as a vacation. Images will normally be captured until a memory card is full and then off-loaded to a computer. Typically a user will save these clusters of images in folders and, if they are proactive they may even sort them into a series of major subcategories such as vacations, social events, family outings, etc.

From our perspective what is relevant is that an entire image collection is unlikely to be in a single folder on a computer, but instead will be organized in some manner into a series of main image folders each of which may contain additional subfolders. It is this arrangement of images that inspired our consideration of a personal image collection as a set of subcollections.

An overview of the descriptor dataset for a single image collection is given in Figure 9.1. This represents the data extracted after training of a PCA basis vector set and performing PCA, DCT and peripheral region analysis of each image in the collection. Note that we have separated the data into fixed data which is derived from the DCT and colour correlogram analysis techniques, and training dependent data which is derived from our PCA-based analysis of each face region in the image.

Now realizing that a practical user application will not retrain the entire collection every time a new cluster of images is added it is evident that it would makes sense to train the images in each individual folder of a collection independently. Thus, when a new cluster of images is added it is only necessary to train that cluster, which is a much faster process than retraining an entire image collection.

The disadvantage of this approach is that, from conventional wisdom, each cluster, or *subcollection* of images will be described only *in its own context.* In other words, each subcollection is analyzed and a set of PCA basis vectors is determined based on the face regions detected in that subcollection. Each image in that subcollection is subsequently analyzed and a feature vector, or descriptor is determined in terms of that set of basis vectors. Thus we can obtain a similarity measure between images of each subcollection, but if we need to measure the similarity of face regions across several subcollections we should, again according to conventional wisdom, begin by determining a common set of PCA basis vectors across all of the images from each subcollection. In other

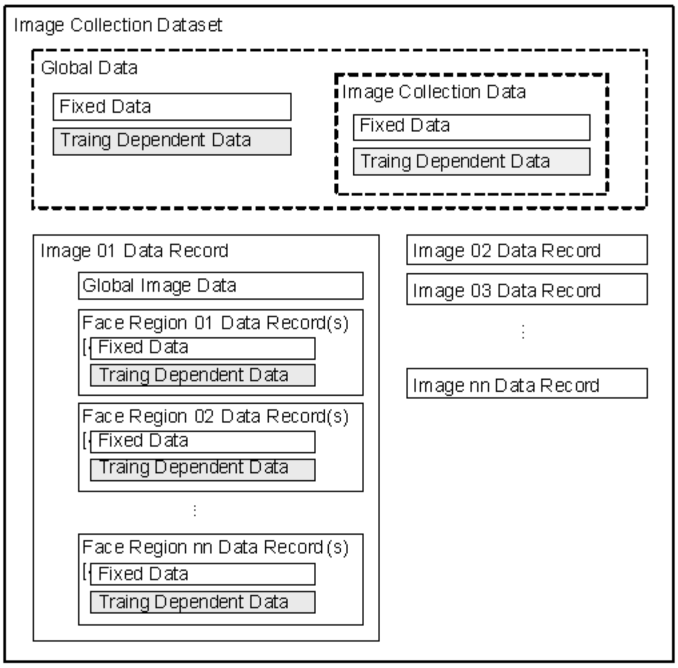


Figure 9.1: Image Collection Descriptors

words, discarding the initial training for each image set and retraining across the combined image set.

There are several disadvantages to this approach, the main one being the additional time required for retraining each time a user wishes to initiate a new search. From a user perspective a delay of even a few 10s of seconds is undesirable. A second disadvantage is that each new search will generate a new set of PCA basis vectors and a description of each face region in terms of those basis vectors. If we attempted to store search results for each potential combination of subcollections in order to speed up subsequent searches, then the size of the descriptor dataset would grow larger at a factorial rate.

This problem does not occur for the DCT or correlogram data which are determined from an absolute frame of reference. We illustrate the situation of combining multiple subcollections in Figure 9.2 where the PCA data is, essentially undetermined and requires a retraining process to solve this.

Now we remark that the training data for each subcollection includes the mean face for that subcollection. This represents an average of all the face regions within that collection and can be updated across a set of subcollections by a weighted linear combination of the mean faces from each subcollection. Further, as PCA components within each subcollection are measured relative to the mean face, led us to consider if there might be some mechanism to reuse

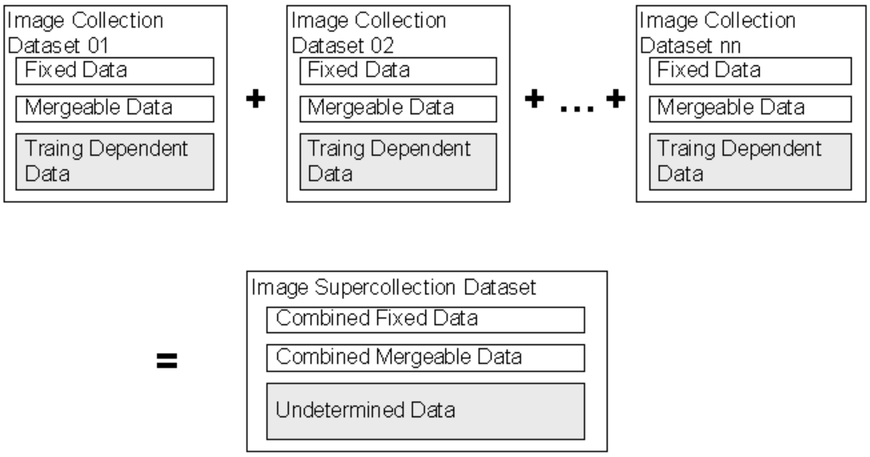


Figure 9.2: Combining Image Collection Datasets

existing training data without requiring a full retraining process.

9.2 Mathematical Model

Let us assume we have a collection of N data samples $\mathbf{S}_i$ $(i = 1,\bar{N})$. Each data sample is a vector of dimension m with $m < N$. The first step consists in changing the collection of data so that it will have a zeros mean, which is done by subtracting the mean of the data samples from each sample. The PCA algorithm consist in computing the covariance matrix $\mathbf{Cov}_{SS}$ using (9.1)

$$\mathbf{Cov}_{SS} = \frac{1}{N}\mathbf{S} \times \mathbf{S}^T \tag{9.1}$$

The matrix S is formed by concatenating each data vector S_i. The size of the covariance matrix is equal to $m \times m$ and is independent of the number of data samples N. We can compute the eigenvector matrix $\mathbf{E} = [\mathbf{e}_1\mathbf{e}_2 \ldots \mathbf{e}_m]$ where each eigenvector $\mathbf{e}_i$ has the same dimension as the data samples m and the eigenvalues $[v_1 v_2 \ldots v_m]$ of the covariance matrix using (9.2):

$$\mathbf{Cov}_{SS} = \mathbf{E} \times \mathbf{V} \times \mathbf{E}^T \tag{9.2}$$

where the matrix V has all the eigenvalues on the diagonal and zeros in rest.

$$\mathbf{V} = \begin{pmatrix} v_1 & 0 & \cdots & 0 \\ 0 & v_2 & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & v_m \end{pmatrix} \tag{9.3}$$

We can reconstruct each data samples using a linear combination of all eigenvectors as shown in (9.4).

$$\mathbf{S}_i = \mathbf{E} \times \mathbf{P}_i \tag{9.4}$$

where $\mathbf{P}_i$ represents the principal component coefficients for data sample $\mathbf{S}_i$ and can be computed by projecting the data sample in the coordinates given by the eigenvectors using (9.5)

$$\mathbf{P}_i = \mathbf{E}^T \times \mathbf{S}_i \tag{9.5}$$

By arranging the eigenvalues in descending order we can approximate the covariance matrix by keeping only a small number of eigenvectors corresponding to the largest values of the eigenvalues. The number of eigenvalues is usually determined by applying a threshold to the values, or maintaining the energy of the original data. If we keep the first $n < m$ eigenvalues and their corresponding eigenvectors, the approximation of the covariance matrix can be computed as

$$\hat{\mathbf{Cov}}_{SS} = \hat{\mathbf{E}} \times \hat{\mathbf{V}} \times \hat{\mathbf{E}}^T \tag{9.6}$$

with $\mathbf{E} = [\mathbf{e}_1 \mathbf{e}_2 \ldots \mathbf{e}_n]$ The data can be as well approximated using only the first n principal coefficients which correspond to the coordinates (eigenvectors) with largest variations inside the data collection.

$$\hat{\mathbf{S}}_i = \hat{\mathbf{E}} \times \mathbf{P}_i \tag{9.7}$$

9.3 Combination scenarios

Let us assume that we have one collection of data C^1 which is analyzed using PCA algorithm which means we have its eigenvectors $\mathbf{E}^{C1} = [\mathbf{e}_1^{C1} \mathbf{e}_2^{C1} \ldots \mathbf{e}_{n_1}^{C1}]$, the eigenvalues $[v_1^{C1} v_2^{C1} \ldots v_{n_1}^{C1}]$, the PCA coefficients for each sample in the collection $\mathbf{Pc}^{C1}$ and supplementary we also stored the mean data sample in the collection $\mathbf{Mean}^{C1}$. We also can assume at this moment that the original data samples are no longer available for analysis or is not practically viable to access them again. There are two different situations in which the collection will be changed:

(a) a new collection of data which was already analyzed using PCA has to be added to the original one in order to form a super collection

(b) a new set of data samples has to be added to the collection in order to update it.

(a) Combining two collections of PCA data

Let us assume we want to combine collection C^1 described above with another collection of data C^2 also PCA analyzed with eigenvectors $\mathbf{E}^{C2} = [\mathbf{e}_1^{C2}\mathbf{e}_2^{C2}\dots\mathbf{e}_{n_2}^{C2}]$, eigenvalues $[v_1^{C2}v_2^{C2}\dots v_{n_2}^{C2}]$, the PCA coefficients Pc^{C2} and the mean data sample $\mathbf{Mean}^{C2}$. We want to combine the two collections into a super collection C without accessing the original data from the two collections $\mathbf{S}^{C1}$ and $\mathbf{S}^{C2}$ ($\mathbf{S}^{C1}$ and $\mathbf{S}^{C2}$ are data matrices where each column $\mathbf{S}_j^i$ represented a vector data sample). The mean sample in the collection can be computed as:

$$\mathbf{Mean} = \frac{N^{C1} * \mathbf{Mean}^{C1} + N^{C2} * \mathbf{Mean}^{C2}}{N^{C1} + N^{C2}} \tag{9.8}$$

where N^{C1} and N^{C2} represents the number of data samples in each collection. It is easy to prove [82] that the covariance of the super collection $\mathbf{Cov}_C$ can be computed as:

$$\begin{aligned}
\mathbf{Cov}_C = \frac{1}{N^{C1}+N^{C2}}[(\mathbf{s}^{C1} - \mathbf{Mean})(\mathbf{s}^{C2} - \mathbf{Mean})] \times [(\mathbf{s}^{C1} - \mathbf{Mean})(\mathbf{s}^{C2} - \mathbf{Mean})]^T \\
= \frac{1}{N^{C1}+N^{C2}}([(\mathbf{s}^{C1} - \mathbf{Mean}^{C1}) \times [(\mathbf{s}^{C1} - \mathbf{Mean}^{C1})]^T + \\
+[(\mathbf{s}^{C2} - \mathbf{Mean}^{C2})] \times [(\mathbf{s}^{C2} - \mathbf{Mean}^{C2})]^T + \\
+\frac{N^{C1}N^{C2}}{N^{C1}+N^{C2}}(\mathbf{Mean}^{C1} - \mathbf{Mean}^{C2})(\mathbf{Mean}^{C1} - \mathbf{Mean}^{C2})^T) \\
= \frac{N^{C1}}{N^{C1}+N^{C2}}\mathbf{Cov}_{C1} + \frac{N^{C2}}{N^{C1}+N^{C2}}\mathbf{Cov}_{C2} + \\
+\frac{N^{C1}N^{C2}}{(N^{C1}+N^{C2})^2}(\mathbf{Mean}^{C1} - \mathbf{Mean}^{C2})(\mathbf{Mean}^{C1} - \mathbf{Mean}^{C2})^T)
\end{aligned} \tag{9.9}$$

We stated above than we cannot recompute the covariance matrix using (9.2) which uses all original data sample. We have two options: first we can store also the covariance matrices from the two collection and use them to compute the exact values of the covariance matrix of the super collection or we can approximate each covariance matrix using the eigenvalues and eigenvectors from each collection

$$\begin{aligned}
\hat{\mathbf{Cov}}_C = \mathbf{E}^{\hat{C}1} \times \mathbf{V}^{\hat{C}1} \times \hat{\mathbf{E}}^{C1T} + \mathbf{E}^{\hat{C}2} \times \mathbf{V}^{\hat{C}2} \times \hat{\mathbf{E}}^{2CT} + \\
+\frac{N^{C1}N^{C2}}{(N^{C1}+N^{C2})^2}(\mathbf{Mean}^{C1} - \mathbf{Mean}^{C2})(\mathbf{Mean}^{C1} - \mathbf{Mean}^{C2})^T
\end{aligned} \tag{9.10}$$

If we assume that from each collection the eigen decomposition (number of eigenvectors kept) was done so that the energy remains almost the same we can assume (prove later in tests) that the estimated covariance matrix of the super collection will be close to (9.9). Another important issue that needs to be addressed is what happens if the two collections of data contains samples of different dimension. One example could be in face recognition if we want to combine two collection of faces that were analyzed using different standard sizes of the face region. In this case we should choose a standard size for the super collection (like the minimum of the sizes between the two collections) and resize the eigenvectors to this standard size.

Once we have the covariance matrix of the super collection we can use the eigen decomposition again and compute the eigenvectors $\mathbf{E} = [\mathbf{e}_1\mathbf{e}_2 \dots \mathbf{e}_n]$ and the eigenvalues $V = [v_1 v_2 \dots v_n]$ of the super collection. The number of eigenvalues kept for analysis n is independent of n_1 and n_2. Once we have the eigenvectors we can project the data samples. Remember that we don't have the data samples to project them easily so we have to recuperate them from the old PCA coefficients. If we want to recuperate a data sample we can use (9.4). The result represents the data sample from which the mean data sample in collection 1 was subtracted so the exact value of the data sample is computed as:

$$\hat{\mathbf{S}}_i^{(1,2)} = \mathbf{E}^{(C1,C2)} \times \mathbf{P}_i^{(C1,C2)} + \mathbf{Mean}^{(C1,C2)} \tag{9.11}$$

We have to subtract the mean of the super collection **Mean** (9.8) from this data sample. We can re-estimate the PC coefficients for each data sample in the super collection as:

$$\hat{\mathbf{Pc}}_i^{(C1,C2)} = \mathbf{E}^T \times (\mathbf{E}^{(C1,C2)} \times \mathbf{P}_i^{(C1,C2)} + \mathbf{Mean}^{(C1,C2)} - \mathbf{Mean}) \tag{9.12}$$

The same remark again if the size of the data sample in the two collection are different the mean sample has to be resized as well.

(b) Adding new data sample to a collection of PCA data

In this case we assume that we have the collection of data C^1 described and we want to add new N^{C2} data samples that are not analyzed already. The sample matrix is $\mathbf{S}^{C2}$ and their mean value is $\mathbf{Mean}^{C2}$. The new covariance matrix will be computed as:

$$\hat{\mathbf{Cov}_C} = \mathbf{E}^{\hat{C}1} \times \mathbf{V}^{\hat{C}1} \times \hat{\mathbf{E}}^{1CT} + \frac{1}{N^{C1}+N^{C2}}[(\mathbf{S}^{C2} - \mathbf{Mean}^{C2})] \times [(\mathbf{S}^{C2} - \mathbf{Mean}^{C2})]^T + \\ + \frac{N^{C1}N^{C2}}{(N^{C1}+N^{C2})^2}(\mathbf{Mean}^{C1} - \mathbf{Mean}^{C2})(\mathbf{Mean}^{C1} - \mathbf{Mean}^{C2})^T \tag{9.13}$$

Applying the same algorithm as described in (a) we compute the new eigenvectors and eigenvalues. Using (9.12) we can update the PCA coefficients of the initial collection and to compute the PCA coefficients of the new data samples we use:

$$\hat{\mathbf{Pc}}_i^{C2} = \mathbf{E}^T \times (\mathbf{S}_i^{C2} - \mathbf{Mean}) \qquad (9.14)$$

where the **Mean** values can be computed using (9.8)

9.4 Application in Face Recognition

We applied the method described in the previous section in a face recognition task. We used our collection of 560 faces (56 individuals each with 10 faces). We split the images into two parts: the training batch containing half of the faces 280 and the other half for testing. We performed two tests by splitting the training faces into two collections with different number of faces. The faces were randomly attached to one of the initial collections:

(i) Test A: we split the training collection into 2 collections each with 140 faces and performed classification tasks using three cases: simple concatenation of PCA coefficients without updating, classical combination using data samples and the proposed method.

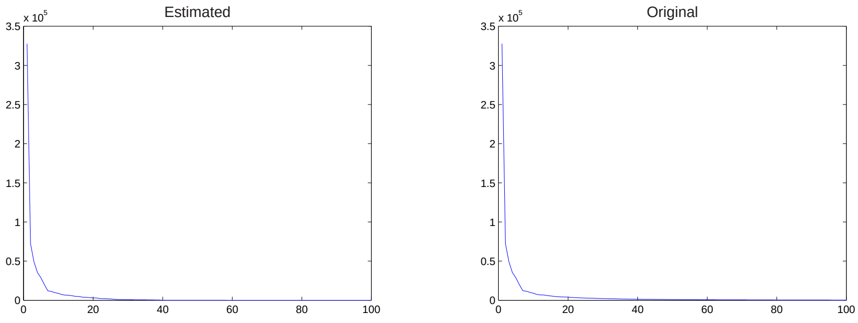
(ii) Test B: using two collections one with 240 face and the other with the remaining 40 in two cases: the classical approach and the proposed method. Also for this test we used both scenarios: combining two collections of PCA data and adding new data to a trained collection.

All faces were resized to 16x16 pixels, gray scale images were used and the number of PCA coefficients used in the classification stage was always 20. For classification the nearest neighborhood method was preferred and the Euclidean distance was used as distance between feature vectors.

Figure 9.3 represents the variation of the eigenvalues using the classical approach of combining two collections and the proposed method. It can be noticed that the variations are quite similar.

In order to see how the eigenvectors differs from the classical combination compared with the proposed method Figure 9.4 shows the first eigenfaces from each collection along with the first eigenfaces obtained using the classical combination and the proposed method. It can be noted that the representation using the two methods have almost identical distributions.

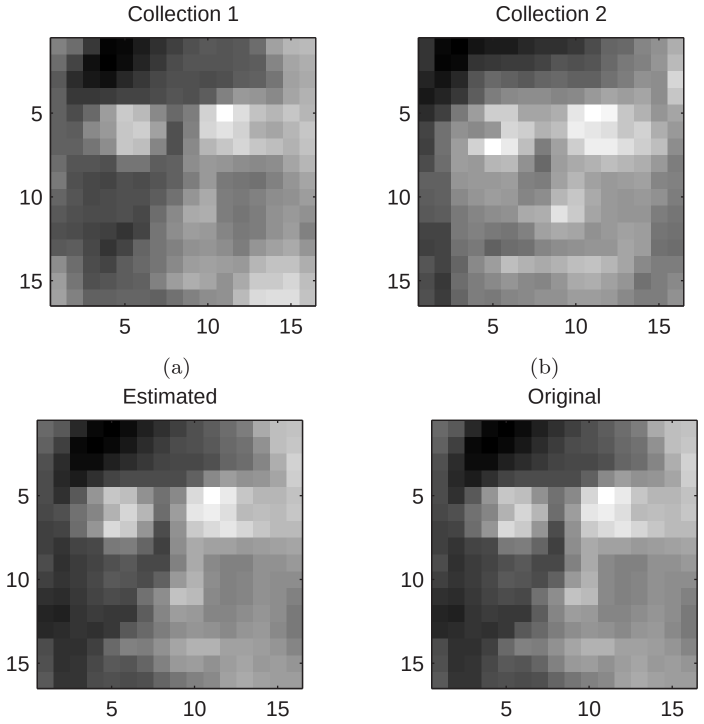
For Test A the second scenario is unlikely because the collections have the same number of samples so we tested only the first scenario. The results are given in the first table.



(a)

(b)

Figure 9.3: Eigenvalue representations a) original and b) estimated



(a)

(b)

(c)

(d)

Figure 9.4: First eigenfaces from both collections and the ones obtained using the classical and the proposed methods

Table 9.1: Recognition Rates for Test A

	Combining collections
Simple concatenation	48.32%
Classical combination	84.28%
Proposed method	85.23%

For Test B we used both scenarios: combining two collections (one having 240 data samples and the other having only 40 samples) and adding new sample to one collection. The results are given in Table 2.

Table 9.2: Recognition Rates for Test B

	Combining collections	Adding samples to collection
Simple concatenation	62.14%	na
Classical combination	84.28%	84.28%
Proposed method	84.64%	83.14%

It can be observed that in both tests the proposed combination method had results close to the classical approach of re-analyze the combined collection using PCA. On the other hand, as expected, if the PCA coefficients are not updated after combination or multiple sample or added to the collection the recognition rate could get much lower.

9.5 Conclusions

The chapter presents a method of updating the PCA coefficients of data samples when the data collection changes due to new data addition or after combining two collections of data previously analyzed using PCA into a super collection. The main advantage of the proposed method is the fact that it doesn't require the original data sample in order to update its coefficients which is very helpful when analyzing very large collections of data and it is mandatory when the original data is lost. Another advantage is that the proposed method is much faster than the classical method of recomputing the PCA coefficients using the original data samples because the dimension of the PCA data is smaller than the dimension of the original data, one of the property of the PCA algorithm being dimensionality reduction of data collection.

The results of our tests proved the reliability of the method which gives results closed to the results of the classical combination method of recomputing the PCA coefficients and as expected if the PCA coefficients are not updated when data collection is modified the results of the classification can get very low.

Future tests will consist in using larger collections of data in order to investigate the robustness of the method and also use other type of data than face images.

CHAPTER

10 Conclusions

In this concluding chapter I will review the outputs of my research, detailing what I feel is the relevance of this work and my major achievements in section 10.1 below. I will also give a comprehensive step by step review of my research activity (section 10.2) and publication outputs (section 10.3). Section 10.4 attempts to place this research in a broader context and to explain its relevance to consumers and its potential impact on the "information society". Finally, perhaps the most important section is that on future research directions (section 10.5) where I review some of the ancillary research activity that has been spawned from this work.

10.1 Summary of Research Achievements

This thesis tackles a topic of considerable interest to many people: the automatic organization/sorting of consumer digital image collections based on the people in those images. In fact this approach is one of the most natural ways for most users to categorize images, although it should be augmented by other mechanisms such as time and location of image capture, etc, for a comprehensive solution.

From a research perspective this thesis has presented me with a huge personal challenge. Indeed it is only now as I have recently begun working full-time in the company that sponsored my research that I realize how large a can of worms I opened entering into this research. It is clear to me now that there are no silver bullets to solve completely the problems of facial or person recognition, particularly as the quality and capture metrics of consumer images can vary so widely. Nevertheless I believe that the work embodied within this thesis can provide a working solution for many users and a useful starting point for future research.

I have managed to quantify many of the principle problems, particularly those of facial pose and illumination, and provide reasonably effective solutions to those for particular test datasets. This work has helped influence the research of a colleague of mine, Mircea Ionita, who is currently developing an illumination independent Active Appearance Model [28]. Using this face modeling technique we can change the orientation of any given face to a standard orientation thus eliminating the huge influence of face pose variations over the recognition accuracy. Also the AAMs can model illumination variations and can be used to normalize these variations.

In **Chapter 6** I have presented test results from what I believe is the first example of a working *Person Recognizer* system which employs both face recognition data and peripheral region data to analyze and sort people within an image collection. Several working embodiments of this core system are further described in **Chapter 8** and a patent has been filed [106] on certain aspects of this system.

I believe that I have also presented an original and detailed comparative analysis of DCT and PCA techniques used individually and in combination across a series of four independent research data sets in **Chapter 7**. This chapter also details approaches to handle extremes of (frontal) pose and illumination and I believe that some aspects of the use of AAMs in this context may also be novel.

Another interesting aspect of this research is described in **Chapter 9**. This approach was considered because of the training problem which arises when PCA based techniques are applied to consumer image collections. Ideally the image collection should be retrained as it grows in size, but this leads to slow response times which are unacceptable for consumer applications. Thus I developed the hypothesis that we can combine several collections of data already trained using PCA without full re-training of the combined collection. So we can update the PCA data of the collection using the old PCA data and some statistical properties of the new formed collection without using the original data samples. This approach can also be used to update the PCA coefficients for a collection of data when adding new data samples to it - incremental PCA training. My initial results with this technique proved very promising and we have recently submitted a journal paper on this topic to IEEE Image Processing Letters. I am currently completing additional experiments in response to reviewer's comments and we expect to publish the results later this year.

10.2 Comprehensive Review of Research Achievements

In this section I will provide more details on the evolution of my research and some of the logical progression of this work over the past 3 years.

The core theme of my research was to design, implement and test an adaptable *Person Recognizer* system which achieves the practical goal of analyzing and categorizing consumer image collections. This system was to provide practical automatic solutions for this problem by using recent advances in computer vision, signal processing and pattern recognition. A secondary goal for my research, given that I am an engineer and this is an engineering PhD, was to address the practical issues of implementation for each reported technique and for the system as a whole.

The key idea for this work was to use a biologically inspired method of browsing images using the persons in that set of images (or a larger set) to provide the primary focus. This is based on the observation that when we review our personal pictures we are mainly looking at the people present in the images: children, family members, friends or colleagues. Thus, it is intuitive to use information specific to each person to sort and manage collections of images. Because the goal was to use people as classification patterns, the system was named the *Person Recognizer.*

In my introduction to this thesis I presented the key goal of this thesis as being the design, implementation, testing and practical realization of an adaptable *Person Recognizer* system which achieves the practical goal of analyzing and categorizing consumer image collections. In addition to this I described a series of sub-goals including:

(i) a study of state of art in face detection and face recognition techniques and implementation of working system modules;

(ii) a study of alternative image classification techniques and implementation of same;

(iii) the development of classifier combination techniques to improve the performance of the system when more than one face recognition or image classification technique is employed;

(iv) the integration of the above into a single system to enable comprehensive system testing;

(v) enhanced testing and refinement of the individual techniques, particularly over a range of facial poses and illumination conditions;

(vi) the study and development of techniques to compensate for extremes of (frontal) pose and illumination;

In addition I had expected that this research would lead to novel refinements of certain aspects of the facial recognition and image classification techniques we employ. I will next work through the various chapters of this thesis illustrating how each of these sub-goals has been achieved and commenting from time to time on specific aspects of the research that I feel were particularly interesting or which involved an unconventional or novel approach.

The first goal is treated in the second and third chapter as follows: person detection is discussed in the second chapter and is based on using a state of the art face detector. The chapter starts with a review of the current status of face detection algorithms and continues with a detailed description of the face detection algorithm used in the *Person Recognizer* system. In the end of the chapter, possible improvements to the detection are discussed along with the main advantages of the detection algorithm. These advantages mainly concern the detection rate, the detection speed and the complexity and portability of the algorithm.

In order to analyze and distinguish between different persons, the obvious choice is to use the person's face region because this is what humans use to classify persons. By using only the face region the problem of classifying person becomes a classical face recognition problem. Unfortunately there are many problems when using classical face recognition algorithms on consumer images. These problems mainly concern the huge variability of face orientation and face illumination that is usually present in common consumer images. Under these conditions it is very difficult to design and implement a reliable face recognition algorithm to be used in our system.

Chapter 3 is dedicated to face recognition. It starts by giving a global description of the problem and continues with a review of face recognition algorithms reported in the literature. Next the proposed face recognition architecture is described. This system is based on combining two classical face recognition algorithms using a statistical approach: DCT and PCA applied on a sub-band from the wavelet decomposition. By combining two algorithms we can increase the robustness and accuracy of the overall face recognition method compared with using each recognition algorithm separately as we verified in testing. The proposed face recognition system is another contribution of this thesis.

Initially we chose the DCT for face recognition due to its simplicity and good results in constrained conditions. Then we realized that for high variations in the datasets the performance of DCT decreases. Next the classical PCA was tested but it was noticed that it is very sensitive to small variations and noise in the face regions. The next step was to reduce these variations using for analysis only the LL sub-band from the wavelet decomposition which corresponds to low pass filtering of the face region. As we noticed that PCA and DCT gave complementary results on different image collections we decided to combine the two algorithms for a more robust system.

The next goal of the thesis, to improve the person recognition using additional classification methods, is realized by using extra information extracted from regions other than the face region. This information should be more robust to the variations that affect the accuracy of the face recognition algorithms, and should also be discriminative between individuals. For this reason two surrounding regions were automatically defined from the face region containing information about the individual: body region and hair region.

The body region information can be used when searching for people in a collection taken on short periods of time (special events, holidays) when people are wearing the same clothing. Also the hair region remains the same over longer periods of time and can be used successfully for recognizing small groups of people. Both regions are also useful for sorting different occurrences of the same person within a large image collection.

The analysis and classification of the additional region used for person classification is described in **Chapter 4** and it is based on analyzing the spatial distribution of the colors inside these regions. By using the spatial distribution (computed using color correlograms) the results are better than using the classical global distribution given by the well known image histogram.

A key factor of the *Person Recognizer* is how to combine the scores obtained from multiple classifiers (DCT and PCA for face recognition and color distribution for additional regions). This contribution is discussed in **Chapter 5**. The proposed statistical approach is presented along with its advantages simple solutions for merging of multiple scores.

Another problem addressed in this thesis is how to evaluate the overall system. Multiple databases were used in order to test the performance of the *Person Recognizer* and its different modules. To test the accuracy of the face recognition module we used classical face recognition databases along with our own collections of images with large variations. To test the overall performance of the *Person Recognizer* we used only our own database. Our database tries to simulate more accurately a common consumer collection with large and different types of variations compared with the classical databases used in more conventional face recognition research.

As the face recognition module is still sensitive to large variations in illumination and orientation we tested different approaches to improve the robustness of the module to these factors. These approaches are discussed in **Chapter 7** and are based on using a variant of the histogram equalization method for illumination normalization called: Contrast Limited Adaptive Histogram Equalization which proved to be effective. We also tested this normalization technique on a different face recognition method in our group [27] with good results.

For pose normalization we analyzed the use of AAM face modeling technique [28] which also proved to be a good option for increasing the robustness of the face recognition to pose variation.

Three working embodiments of the *Person Recognizer* that were implemented during this thesis are presented in **Chapter 8**. Two of the applications are designed to be used as desktop applications for organizing the pictures stored on the computer and the third application is designed for network use and can be implemented on photo servers with the possibility to search the images on the severs using a web interface.

Chapter 9 describes a novel approach for incremental updating of the PCA coefficients of collections of data when these collections are modified. This type of training is needed when working with large collections of data, where PCA training can be very time consuming. Also considering our application, organizing consumer image collections, where these collections of images are changing all the time, which implies re-training of the database. This re-training, if it is time consuming, can be annoying for the user. Note that the algorithm for incremental retraining can be applied to any application that uses PCA training algorithm and is not specifically restricted to face datasets.

10.3 Papers Published & Conferences Attended

After the completion of each stage from the *Person Recognizer* we published papers to report our research progress to the academic world.

The first paper [79] presented and published as digest paper at the IEEE ICCE conference in 2005 reports the generalities of the system, how it should work and why it will be very useful for consumers. We introduce the person as being the main pattern used to organize the images and also we introduce the idea of using multiple regions to describe and classify people.

Next we were invited to present the idea of using persons and multiple regions for retrieving images of the same person along with our initial results to the conference organized by the European Space Agency specialized in image processing and information retrieval held in Frascati Italy [31].

After we implemented and tested the proposed face recognition method along with the face detection we presented the results to different conferences which are focused on image processing and pattern recognition [Optim - Romania,GSPX].

Once the *Person Recognizer* system was implemented and we had tested it on different databases a full journal paper[30] was prepared, accepted and published in the IEEE CE Transaction on, journal.

Possible applications of the system were presented to specialized conferences: in GSPx [81] we first introduced the idea of an embedded implementation of the system inside a digital photo camera in order to browse for images directly on the camera. At the same conference we presented a possible application of the system using mobile phones to browse image collections on a remote server.

In 2006 at the IEEE ICCE conference we presented [108] the idea of using a distributed architecture for the system that can train images on a server or embedded in a digital camera and using a different device to browse through the images using this training data. This was also published in the conference digest but was not yet developed into a full journal paper as certain elements of this work are the subject of pending patent filings.

In Sibiu [32] we presented a possible integration of the *Person Recognizer* in a home networking environment for photo management which is a topic of real interest these days.

The tested solutions for improving the robustness of the face recognition approach were presented at Optim Brasov: using CLAHE for illumination normalization prior to our recognition approach [80], using CLAHE prior to HMM based face recognition approach [27] and using AAM technique to model the face pose [28].

One problem of using PCA based techniques for signal analysis is that PCA is a data dependent method. This means that every time the data collection changes (due to addition, deletion, merging or splitting) the PCA algorithm has to be re-applied. For a consumer application this could mean extra waiting time and extra resources. In order to overcome this problem we designed a PCA re-computing technique that uses previous computed PCA data to recalculate the new PCA data without applying PCA all over again. This could greatly reduce the time needed for image collection analysis. This technique is described in **Chapter 9** and we submitted a journal paper on this topic to IEEE Image Processing Letters. I am currently completing additional experiments in response to reviewer's comments and we expect to publish the results later this year.

10.4 Application and Relevance of the Research

Interest in the topic of organizing/cataloging consumer images is driven primarily by the rapid evolution of digital photography over the last decade. As users switch to digital photography, they find themselves with rapidly growing collections of digital images. This growth is further accelerated by falling prices for acquisition equipment such as digital cameras and high-resolution camera-phones and electronic storage such as memory cards, and hard drives peripherals. People are encouraged to take many pictures but few consumers have the time, personal discipline or technical knowledge to manually catalog and organize these growing personal image collections in a manner which will enable them to easily retrieve images in the future.

Thus there is an increasing need for simple, yet powerful, automatic tools for sorting and cataloging digital image collections. Such automatic tools should be easy to integrate into different platforms and must be intuitive to use even for non-technical people. The goal of my research has been to provide both the simplest and most useful of such tools - the basic query-by-example type of application: ”Find me this person!”

Although it is a simple concept it is nevertheless a powerful tool. If properly realized and integrated into a workflow, it can truly empower users. It supports our own natural ability to rapidly scan an image and accurately determine the people who are participating in the captured scene. By automating this process we can help users to pre-filter from many thousands of images the handful of images they really want to view and evaluate.

In order to minimize the time needed to analyze an image collection such tools should eventually support in-camera image analysis. Thus a picture would be analyzed as soon as it is captured by the digital camera. Once they are copied onto the storage equipment, the data resulting from this image analysis is copied as well. My current employment with the sponsoring company is, in fact, directed towards this research goal. Not surprisingly this presents a new set of challenges and difficulties, but I am optimistic that I can overcome these and that eventually we will see smart cameras that will learn about our friends and members of our family and will be capable of interacting directly with remote storage applications to help us creating a comprehensive record in pictures of our individual lives. And not just a static record, but rather a dynamic and interactive one. A key component of this vision has its humble beginnings in the work presented in this thesis.

Other applications that can use these tools are web servers dedicated to photo management: photo servers which can incorporate the tools so the user can better manage their image collections. I built some prototype network implementations of the *Person Recognizer* working with a colleague, Rhys Mulryan. This implementation was described in **Chapter 8**. Rhys also implemented a *Person Recognizer* application for mobile phones that uses this prototype network description [29].

As intelligent homes are a very popular subject these days, these tools can also be integrated, as shown in some further work I undertook with another colleague, Frank Callaly [78]. His research is focused on home network architectures and we successfully demonstrated how my modular *Person Recognizer* could be integrated into such a home network architecture, behaving as an intelligent home service for photo management.

Person Recognition/Retrieving could also be employed in interactive video applications where the user can easily and quickly browse video content for preferred sequences/scenes based on the presence of certain actors or characters. Although we did not explicitly explore this option it should provide the basis for follow-on research to extend these techniques to the analysis of digital video and movie sequences.

10.5 Future Research Directions

As stated above there are special requirements for the face recognition algorithms when working with consumer images or consumer applications. Unlike security applications based on face recognition where the most important requirement is the accuracy of the algorithm, for the case of consumer applications the requirements differ somewhat. The complexity of the algorithm is important because it is likely to eventually be implemented in embedded imaging systems where resources are more constrained than on desktop computers. Thus any implementation must be modular and lightweight. Ideally it should not require a significant training phase, or it should be possible to train separately - ideally on a desktop computer. An incremental training mechanism is, in fact, desirable so that an entire image collection need not be available in order to retrain recognition classifiers, but rather small additional groups of images can be trained separately and the results combined. Also the speed of the algorithm, in particular the apparent time delay introduced into the image workflow will affect the user impression of the performance and the utility of the *Person Recognizer* algorithm. Finally the apparent accuracy and robustness of the algorithm must be maintained. All of these aspects of the *Person Recognizer* are the basis for research topics in their own right.

Some preliminary theoretical work on an incremental training approach has been presented in **Chapter 9** and initial tests have been encouraging. A new colleague, Pawel Puslecki, has begun a more comprehensive study of this technique to determine its strengths & weaknesses.

Another research theme is on the topic of Hidden Markov Models applied to face recognition. This work is being undertaken by another colleague, Claudia Iancu, who has demonstrated some promising initial results which support a training workflow where as few as two images of a single person may be sufficient to create an initial person model which can cope with different variations in the face images.

I, myself, am working on the implementation of a lightweight embedded algorithm to support an initial recognition process in current digital cameras. For this work I have to take into account the properties of consumer images collections that usually contain plenty of variations including: variations in illumination and face orientation. These types of variations can influence the accuracy of any recognition system if they are not addressed but in consumer images the variations are wider and they have to be handled more accurately and minimized more efficiently. In this thesis I addressed these issues and managed to build a robust recognizer but practical consumer images with wider variations still affect the accuracy of the classification algorithms. I have realized over the last couple of months that significantly more work must be done in order to build a fully robust *Person Recognizer*.

One promising field of research being undertaken by another colleague, Micea Ionita, is to build a completely generic Active Appearance Model which is independent of illumination and pose. Mircea has had some good results in tackling both of these key variations independently and is eventually hoping to achieve a combined model. Such a model would enable a detected face region to be more accurately matched by the generic AAM model which could then extract a frontal facial image which was accurately normalized for both illumination and pose.

10.6 Concluding Remarks

My first impression of PhD research back when I was an undergraduate student was that this research should be extremely complex, and restricted to a very specific field or domain of research. Only a few people in the world would ever understand what that research was trying to achieve and how it would do that. And, naturally, the work should be addressed only to the academic world with no social or industrial relevance, at least not in the short term.

On reflection I have been very lucky. The main topic of my research addresses a real issue of our times created by the rapid technological evolution that characterize this first decade of the 21^{st} century. I have been able to leverage some recent advances in a number of related research topics in the fields of face detection, recognition and pattern matching. By improving some techniques and proposing new algorithms for combining multiple pattern recognition methods I have been able to present several practical solutions to the problem of person recognition in large collections of consumer images.

All modules, algorithms and architectures were designed, written and tested with a view to practical implementations. Some of these solutions are already implemented or are on the way to be implemented in consumer applications by our industry partner that is interested in this topic and I have to thank them for providing a substantial amount of feedback, comments and suggestions which greatly improved the results and outcome of this thesis.

I was very happy to be able to combine high level concepts and algorithms that I learned during my classes as an undergraduate student and afterwards as a junior researcher with recent advances in technology which were always my hobbies.

I hope this work will help consumers get greater enjoyment and benefits from their growing digital picture collections and inspire others to improve on the tools and techniques described herein and perhaps develop even more helpful and powerful tools in the future.

Appendix

A Face recognition - will extra knowledge about the training data always help?

Face recognition methods can be divided into two major groups if we analyze the way the algorithm takes into account the knowledge about the data available for training. First group of methods ignores, even if available, how the data sample is distributed in the classes in the training stage while the second group uses this information in both analysis and classification stages.

This knowledge about data distribution can be used when the pattern is analyzed in order to extract features that incorporate this information and can also be used in the classification stage, when a classifier is trained , to distinguish between classes having more than one sample from the same pattern.

Simple face recognition algorithms from the first group of methods that don't use the extra knowledge include for face analysis discrete cosine transform (DCT), Fourier transform (FFT), principal component analysis (PCA), extracting geometrical features, extracting texture features or color descriptor features. These types of features do not take into account how the training data is distributed in classes and, except PCA, they don't even consider any kind of information about the data available for training, they are extracted independent of the training collection and only depend on the data sample that is analyzed. PCA uses statistical properties of the entire group of datasamples (mean and covariance).

Some face analysis algorithms from the second group which use the knowledge to extract information that includes this knowledge are: Fisher faces method and Linear Discriminate Analysis (LDA), where the features are computed to be more discriminative between patterns of different classes and closer to sample from the same class using the extra knowledge.

The most common classification algorithm from the first group of methods consists in computing simple distances between feature vectors and using the nearest neighborhood method (takes the smallest distance between the test face and the faces inside a class as the representative distance) determines the final ranking without considering the rest of the distance distributions inside classes.

Most of the current classification algorithms used in face recognition systems belong to the second group of methods and include all the classification algorithms that require a training stage like: artificial neural networks (ANN), hidden Markov models (HMM), support vector machines (SVM), and also distances like Mahalanobis distance which takes into account the distance distribution in classes.

The second group of methods have been proved to perform better than the first group in most cases, which is a logical assumption considering that extra knowledge about the data collection should only improve the recognition rates of the system.

In some cases this assumption is not verified and the performance of the recognition system could get worse when adding this information. In the next paragraph we will try to give a possible explanation of how is this possible.

We started this study after we compared recognition results using PCA approach and Fisher faces approach and against many reports in the literature we obtain better result using PCA features.

Let us consider a case where we have our collection of face images available for training. We use a feature extraction algorithm which transfers each image from the initial color space into a multidimensional feature space, let's call it the "face space". In the face space the features belonging to the same class are grouped into "clusters".

A perfect classification will separate the clusters without intersections, like in Figure A.1 (we use for a better understanding 3 dimensional representation).

However, perfection cannot be achieved in engineering, so the clusters in the face space will have overlapping regions (will intersect). Depending on the variations inside the training data or the feature extracting method or the classification algorithm used the intersections could be smaller or higher.

Such a case is represented in Figure A.2.

If we analyze how variations of the face data influence the clusters in the face space, it is logical to assume that large variations of luminance, pose and size, which are often present in a consumer collection, will make the clusters bigger because the faces will be far away from each other.

If we decide to normalize these factors we will make the clusters smaller and

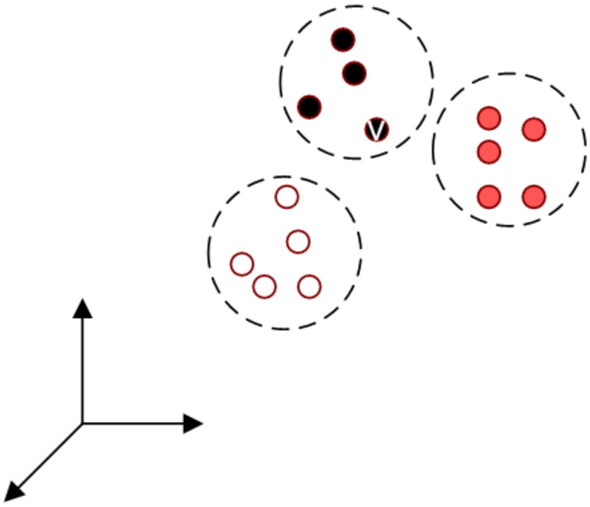


Figure A.1: Perfect classification algorithm - Face Space

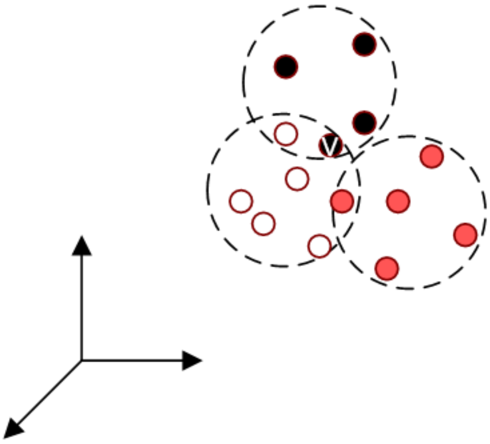
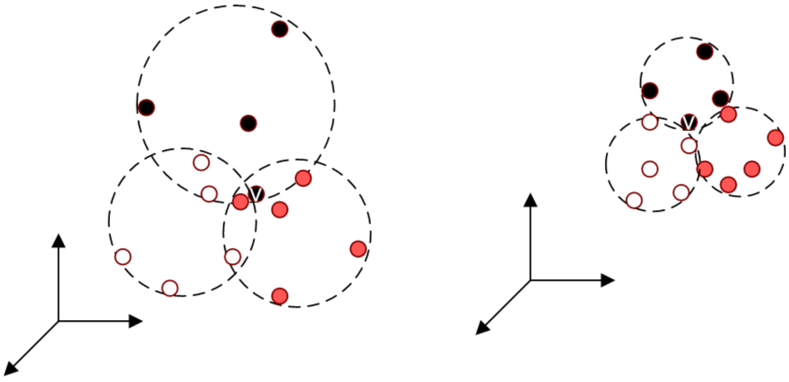


Figure A.2: Real classification algorithm - Face Space

their intersection will be smaller so the classification will be more accurate.



(a) (b)

Figure A.3: Face space a) original b) with normalization

Let us consider the case where we want to classify a test face in the original face space.

First we project the test face in the face space without any knowledge about the training data and use nearest neighborhood method for classification without using any information about the properties of the clusters. This case is represented in Figure A.4. The test pattern to be classified is represented by the blue circle. Let's assume that the test face belongs to the red cluster.

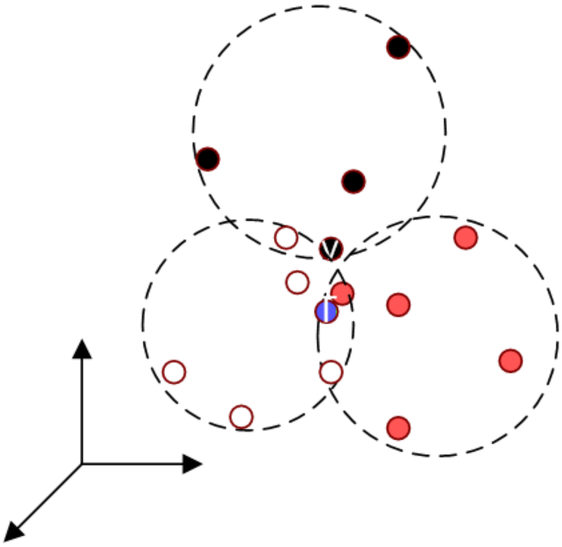


Figure A.4: Face space classification

If we use the nearest neighborhood classification then the decision will be correct as the closest data in the face space belongs to the red cluster.

However if we use any other classification algorithm that is trained with all data of the white cluster or incorporates the statistics of the white cluster, the decision will be that the test data belongs to the white cluster as it is close to many samples in the cluster and it is closer to the center of the cluster.

Also using features that incorporate information about how the data is distributed inside the clusters (FisherFaces) will make the representation of the test data to be closer to the center of the white cluster.

In conclusion, even though the face to be classified is close in appearance to one of the faces used for training, due to the fact that the clusters in the face space are very big and they cross over different regions, the recognition system may take the wrong decision if the test face falls into one of the common regions in the face space, and we use a classification algorithm or a feature extraction algorithm which uses information about how the training data is distributed in clusters.

One method to minimize this effect is to normalize the variations in the training data so that the cluster in the face space will have less or none common regions.

Appendix

A

List of Publications from this Research

This is the list of publications resulted from this thesis:

1. P. Corcoran and **G. Costache**, Automated sorting of consumer image collections using face and peripheral region image classifiers, IEEE Trans. on Consumer Electronics, Vol. 51 , No. 3, pp = 747-754, 2005;

2. **G. Costache** and P. Corcoran and R. Mulryan and E. Steinberg, Method and component for image recognition, US Patent application no: 20060140455, 2006;

3. **G. Costache**, R. Mulryan, E. Steinberg and P. Corcoran, In-camera person-indexing of digital images, Consumer Electronics, 2006. ICCE '06. 2006 Digest of Technical Papers. International Conference on.

4. P. Corcoran, **G. Costache** and R. Mulryan, Automatic indexing of consumer image collections using person recognition techniques, Consumer Electronics, 2005. ICCE '05. 2005 Digest of Technical Papers. International Conference on.

5. P. Corcoran, **G. Costache** and E. Steinberg, Automatic System for In-Camera Person Indexing of Digital Image Collections, GSPx, Santa Clara, Ca, 2005.

6. P. Corcoran, M. Ionita and **G. Costache**, A Pose Invariant Face Recognition using AAM Models, OPTIM 2006, Vol. 4, pp 95 - 109 Brasov, Romania, 2006.

7. P. Corcoran, C. Iancu and **G. Costache**, Improved HMM based Face Recognition System, OPTIM 2006, Vol. 4, pp 143 - 146, Brasov, Romania,

2006;

8. P. Corcoran, **G. Costache** and R. Mulryan , Hybrid Techniques for Automatic Person Retrieval from Image Collections, Col. 4, pp 155 - 160, OPTIM 2006, Brasov, Romania, 2006.

9. P. Corcoran and **G. Costache**, Managing Consumer Image Collections on a Home Network, Vol. 1, pp 266 - 271, RoEduNet, Sibiu, Romania, 2006.

10. P. Corcoran and **G. Costache**, Automatic Person Retrieval from Image Collections, ESA-EUSC Conference, Frascati, Italy, 2005.

Bibliography

[1] http://www.digitalcamerainfo.com/content/Fujifilm-Displays-Face-Recognition-and-Detection-Ability-Available-in-new-Electronic-Photo-Book.htm.

[2] http://www.myheritage.com/FP/Company/face-recognition.php.

[3] http://www.technologyreview.com/read_article.aspx?id=16882&ch=infotech&sc=&pg=1.

[4] http://www.infowars.com/articles/bb/total_surveillance_behind_wheel.htm.

[5] http://www.dpreview.com/news/0502/05021604nikonfaceaf.asp.

[6] http://www-cs-students.stanford.edu/ robles/ee368/skincolor.html.

[7] http://www.intel.com/technology/computing/opencv/index.htm.

[8] Identix Corporation, FaceIt Developer Kit version 2.0, http://www.identix.com/.

[9] HumanScan AG, BioID SDK, http://www.humanscan.com.

[10] http://www.cl.cam.ac.uk/ jgd1000/combine/combine.html.

[11] http://www.humanscan.de/support/downloads/facedb.php.

[12] B. Achermann. The face database of university of bern, switzerland. *http://iamwww.unibe.ch/fkiwww*, 2000.

[13] T. Agui, Y. Kokubo, H. Nagashashi, and T. Nagao. Extraction of facerecognition from monochromatic photographs using neural networks. *Proc. Second Int'l Conf. Automation, Robotics, and Computer Vision*, 1:1881–1885, 1992.

[14] K. Baek, B. A. Draper, J. R. Beveridge, and K. She. Pca vs ica: A comparison on the feret data set. *presented at Joint Conference on Information Sciences, Durham, N.C.,*, 2002.

[15] N. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *Proceedings of the European Conference on Computer Vision - ECCV*, pages 45–58, 1996.

[16] S. Ben-Yacoub, Y. Abdeljaoued, and E. Mayoraz. Fusion of face and speech data for person identity verification. *IEEE Transactions on Neural Networks*, 10(5):1065–1075, 1999.

[17] D. Beymer and T. Poggio. Face recognition from one example view. *In Proc. 5th Intl. Conf. on Computer Vision*, pages 500–507, 1995.

[18] R. Chellappa B.S. Manjunath and C.V.D. Malsburg. A feature based approach to face recognition. *Proc IEEE Conference on Computer Vision and Pattern Recognition*, pages 373–378, 1992.

[19] G. Burel and D. Carel. Detection and localization of faces on digital images. *Pattern Recognition Letters*, 15(10):963–967, 1994.

[20] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[21] K.-P. Chan and A. W.-C. Fu. Efficient time series matching by wavelets. *ICDE*, pages 126–133, 1999.

[22] C.C. Chibelushi, F. Deravi, and J.S.D Mason. A review of speech based bimodal recognition. *IEEE Transactions on Multimedia*, 4(1):23–38, 2002.

[23] CIE. Colorimetry. *2nd Ed., CIE Publ. No. 15.2, Recommendation on Uniform Color Space, Colour Difference Equations, Psychometric Color Terms, Central Bureau of the CIE, Vienna*, 1986.

[24] P. Colantoni and al. Color space transformations. *http://colantoni.nerim.net/download/colorspacetransform-1.0.pdf*, 2004.

[25] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *Proceedings of the European Conference on Computer Vision - ECCV*, 2(3):484–498, 1998.

[26] T.F. Cootes and C.J. Taylor. Locating faces using statistical feature detectors. *Proc. Second Int'l Conf. Automatic Face and Gesture Recognition*, pages 204–209, 1996.

[27] P. Corcoran, C. Iancu, and **G. Costache**. Improved hmm based face recognition system. *OPTIM 2006, Brasov, Romania*, 4:143 – 146, 2006.

[28] P. Corcoran, M. Ionita, and **G. Costache**. A pose invariant face recognition using aam models. *OPTIM 2006, Brasov, Romania*, 4:95–109, 2006.

[29] P. Corcoran and R. Mulryan. A distributed infrastructure for person recognition using mobile phones. *GSPx, Santa Clara, Ca*, 2005.

[30] P. Corcoran and **G. Costache**. Automated sorting of consumer image collections using face and peripheral region image classifiers. *IEEE Trans. on Consumer Electronics*, 51(3):747–754, 2005.

[31] P. Corcoran and **G. Costache**. Automatic person retrieval from image collections. *ESA-EUSC Conference, Frascati, Italy*, 2005.

[32] P. Corcoran and **G. Costache**. Managing consumer image collections on a home network. *RoEduNet, Sibiu, Romania*, 1:266–271, 2006.

[33] I.J. Cox, J. Ghosn, and P.N. Yianilos. Feature-based face recognition using mixture distance. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 209– 216, 1996.

[34] Y. Dai and Y. Nakano. Extraction for facial images from complex background using color information and sgld matrices. *Proc. First Int'l Workshop Automatic Face and Gesture Recognition*, pages 238–242, 1995.

[35] O. Deniz, M. Castrillon, and M. Hernndez. Face recognition using independent component analysis and support vector machines. *Pattern Recognition Letters*, 24(13):2153–2157, 2003.

[36] R. Feraud and O. Bernier. Ensemble and modular approaches for face detection: A comparison. *Advances in Neural Information Processing Systems, MIT Press*, 10:472–478, 1998.

[37] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video: The qbic system. *IEEE Computer*, 28(9), 1995.

[38] R. W. Floyd and L. Steinberg. An adaptive algorithm for spatial gray scale. *International Symposium Digest of Technical Papers, Society for Information Displays*, page 36, 1975.

[39] Y. Freund and R.E. Schapire. A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[40] I. Gavat, **G. Costache**, C. Iancu, and C.O. Dumitru. Svm-based multimedia classifier. *WSEAS Trans on Inf Science & Applications*, 2(3), 2005.

[41] Th. Gevers and A.W.M. Smeulders. Content-based image retrieval: An overview. *Survey on content-based image retrieval from the book Emerging Topics in Computer Vision, G. Medioni and S. B. Kang (Eds.), Prentice Hall*, 2004.

[42] Th. Gevers and H. M. G. Stokman. Classification of color edges in video into shadow-geometry, highlight, or material transitions. *IEEE Trans. on Multimedia*, 5(2), 2003.

[43] A. Girgensohn, J. Adcock, and L. Wilcox. Leveraging face recognition technology to find and organize photos. In *Proc. of the 6th ACM SIGMM international Workshop on Multimedia Information Retrieval*, pages 99–106. ACM Press, October 2004.

[44] H.P. Graf, T. Chen, E. Petajan, and E. Cosatto. Locating faces and facial parts. *Proc. First Int'l Workshop Automatic Face and Gesture Recognition*, pages 41–46, 1995.

[45] D. B Graham and N. M Allinson. Face recognition: From theory to applications. *NATO ASI Series F, Computer and Systems Sciences*, 163:446–456, 1998.

[46] R. Gray. Content-based image retrieval: Color and edges. *Dartmouth College Department of Computer Science Tech Report TR95-252*, 1995.

[47] S. Baluja H. Rowley and T. Kanade. Neural network-based face detection. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 203–208, 1996.

[48] R.M. Haralick, K. Shanmugam, and I. Dinstein. Texture features for image classification. *IEEE Trans. Systems, Man, and Cybernetics*, 3(6):610–621, 1973.

[49] S. Haykin. Neural networks: A comprehensive foundation. *Prentice Hall*, 1999.

[50] B. Heisele and T. Poggio P. Ho. Face recognition with support vector machines: Global versus component-based approach. *Proceedings of the European Conference on Computer Vision - ECCV*, 2:688–694, 1998.

[51] L. Hong and A.K. Jain. Integrating faces and fingerprints for personal identification. *IEEE Transactions PAMI*, 20(12):1295–1307, 1998.

[52] J. Huang, S. R. Kumar, M. Mitra, and W. J. Zhu. Spatial color indexing and applications. *Proc. IEEE International Conference on Computer Vision ICCV*, 1998.

[53] H.Wangand and S.-F. Chang. A highly efficient system for automatic face region detection in mpeg video. *IEEE Trans. Circuits and Systems for Video Technology*, 7(4):615–628, 1997.

[54] O. Dumitru I.Gavat and **G. Costache**. Speech signal variance reduction by means of learning systems. *Procedings of Medinf 2003 Conference*, 2003.

[55] T.S. Jebara. *3D Pose estimation and normalization for face recognition.* PhD thesis, McGill University, Montreal, Quebec, Canada, 1996.

[56] T.S. Jebara and A. Pentland. Parameterized structure from motion for 3d adaptive feedback tracking of faces. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 144–150, 1997.

[57] T. Kanade. Computer recognition of human faces. *Interdisciplinary Systems Research*, 47, 1977.

[58] L.M. Kaplan, R. Murenzi, and K.R. Namuduri. Fast texture database retrieval using extended fractal features. *Proc of SPIE; Storage and Retrieval for Image and Video Databases*, 3312:162–173, 1998.

[59] M. D. Kelly. Visual identification of people by computer. *Tech. rep. AI-130, Stanford AI Project, Stanford, CA*, 1970.

[60] M. Kirby and L. Sirovich. Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE Trans. Patt. Anal. Mach. Intell.*, 12, 1990.

[61] C. Kotropoulos and I. Pitas. Rule-based face detection in frontal views. *Proc. Int'l Conf. Acoustics, Speech and Signal Processing*, 4:2537–2540, 1997.

[62] J.-M. Fellous L. Wiskott and C.V.D. Malsburg. Face recognition by elastic bunch graph matching. *IEEE Trans. Patt. Anal. Mach. Intell*, 9:775–779, 1997.

[63] J. H. Lai, P. C. Yuen, and G. C. Feng. Face recognition using holistic fourier invariant features. *Pattern Recognition*, 34:95–109, 2001.

[64] A. Laine and J. Fan. Texture classification by wavelet packet signature. *IEEE Transactions on PAMI*, 15(11), 1993.

[65] S. Lawrence, C. L. Giles, A. Tsoi, and A. Back. Face recognition: A convolutional neural-network approach. *IEEE Trans. on Neural Networks*, 8:98–113, 1997.

[66] K. Lee, J. Ho, and D. Kriegman. Nine points of light: Acquiring subspaces for face recognition under variable lighting. *In Proc. of CVPR*, pages 519–526, 2001.

[67] L. H. Liang, X. X. Liu, Y. Zhao, X. Pi, and A. V Nefian. Speaker independent audio-visual continuous speech recognition. *IEEE International Conference on Multimedia and Expo*, 2(3):25–28, 2002.

[68] S.-H. Lin, S.-Y. Kung, and L.-J. Lin. Face recognition/detection by probabilistic decision-based neural network. *IEEE Trans. Neural Networks*, 8(1):114–132, 1997.

[69] S.-H. Lin, S.-Y. Kung, and L.-J. Lin. Face recognition/detection by probabilistic decision-based neural network. *ECVP Eur. Conf. on Visual Perception*, 2004.

[70] C. Liu and H. Wechsler. Evolutionary pursuit and its application to face recognition. *IEEE Trans. Patt. Anal. Mach. Intell*, 22:570–582, 2000.

[71] Y. Miyake, H. Saitoh, H. Yaguchi, and N. Tsukada. Facial pattern detection and color correction from television picture for newspaper printing. *J. Imaging Technology*, 16(5):165–169, 1990.

[72] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 19:696–710, 1997.

[73] H.M. Lades M.S. Bartlett and T. Sejnowski. Independent component representation for face recognition. *Proc, SPIE Symposium on Electronic Imaging: Science and Technology*, 24(13):528–539, 1998.

[74] A.V. Nefian and M.H. Hayes. Hidden markov models for face recognition. *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, pages 2721–2724, 1998.

[75] T. Ojala, M. Pietikainen, and D. Harwood. A comparison study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29:51–59, 1996.

[76] K. Okada, J. Steffans, T. Maurer, H. Hong, E. Elagin, H. Neven, and C.V.D. Malsburg. The bochum/usc face recognition system and how it fared in the feret phase iii test. *In Face Recognition: From Theory to Applications, H. Wechsler, P. J. Phillips, V. Bruce, F. F. Soulie, and T. S. Huang, Eds. Springer-Verlag*, pages 186–205, 1998.

[77] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: An application to face detection. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 130–136, 1997.

[78] A. Cocos P. Corcoran and F. Callaly. A universal home multimedia environment i: Network infrastructure and a/v device architecture. *RoEduNet, Sibiu, Romania*, 1:34–38, 2006.

[79] **G. Costache** P. Corcoran and R. Mulryan. Automatic indexing of consumer image collections using person recognition techniques. *Consumer Electronics, 2005. ICCE '05. 2005 Digest of Technical Papers. International Conference on*, page 2, 2005.

[80] **G. Costache** P. Corcoran and R. Mulryan. Hybrid techniques for automatic person retrieval from image collections. *OPTIM 2006, Brasov, Romania*, 4:155 – 160, 2006.

[81] **G. Costache** P. Corcoran and E. Steinberg. Automatic system for in-camera person indexing of digital image collections. *GSPx, Santa Clara, Ca*, 2005.

[82] D. Marshalland P. Hall and R. Martin. Adding and subtracting eigenspaces. *British Machine Vision Conference*, 2:463–472, 1999.

[83] G. Pass, R. Zabih, and Justin Miller. Comparing images using color coherence vectors. *ACM Conference on Multimedia*, pages 65–, 1996.

[84] P. Penev and J. Atick. Local feature analysis: A general statistical theory for objecct representation. *Netw.: Computat. Neural Syst*, 7:477– 500, 1996.

[85] A. Pentland, B.Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 1994.

[86] P.J. Phillips. Support vector machines applied to face recognition. *Adv. Neural Inform. Process. Syst.*, 11(13):803–809, 1998.

[87] E. Pisano, S. Zong, M. Hemminger, M. De Luca, R. Johnsoton, K. Muller, M. Braeuning, and S. Pizer. Contrast limited adaptive histogram equalization image processing to improve the detection of simulated spiculations in dense mammograms. *Journal of Digital Imaging*, 11(4):193–200, 1998.

[88] S. Pizer, E. Amburn, J. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. Romeny, J. Zimmerman, and K. Zuiderveld. Adaptive histogram equalization and its variations. *Computer Vision, Graphics, and Image Processing*, 39:355–368, 1987.

[89] R. Versachae C. Castillo R. Baeza-Yates, J. Ruiz-del-Solar and C. Hurtado. Content-based image retrieval and characterization on specific web collection. *Image and Video Retrieval third International Conference, CIVR, Dublin, Ireland,Springer LNCS3115*, 3115/2004:189–198, 2004.

[90] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[91] V. Raghavan, P. Bollmann, and G.S. Jung. A critical investigation of recall and precision as measures of retrieval system performance. *ACM Trans. Inf. Syst.*, 7:205–229, 1989.

[92] R.Lienhart and J Maydt. An extended set of haar-like features for rapid object detection. *In Proc. of the IEEE Conference on Image Processing (ICIPó2)*, 1:900–903, 2002.

[93] A. Ross and A. K. Jain. Multimodal biometrics: An overview. *Proc. of 12th European Signal Processing Conference EUSIPCO*, pages 1221–1224, 2004.

[94] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 203–208, 1996.

[95] F. Samaria and S. Young. Hmm based architecture for face identification. *Image Vis. Comput*, 12:537–583, 1994.

[96] F.S. Samaria. *Face Recognition using Hidden Markov Models.* PhD thesis, Univ. of Cambridge, 1994.

[97] D. Saxe and R. Foulds. Toward robust skin identification in video images. *Proc. Second Int'l Conf. Automatic Face and Gesture Recognition*, pages 379–384, 1996.

[98] J. Smith and S-F. Chang. Tools and techniques for color image retrieval. *SPIE proceedings*, 2670:1630–1639, 1996.

[99] J. R. Smith. *Integrated Spatial and Feature Image Systems: Retrieval, Analysis, and Compression*. PhD thesis, Columbia University Department of Computer Science, 1997.

[100] J. R. Smith and S. F. Chang. Automated binary feature sets for image retrieval. *Proc of ICASSP, C. Faloutsos, editor Atlanta. Kluwer Academic*, 1996.

[101] W A. P. Smith and E R. Hancock. Single image estimation of facial albedo maps. *BVAI*, pages 517–526, 2005.

[102] M. Stricker and A. Dimai. Color indexing with weak spatial constraints. *SPIE proceedings*, 2670:29–40, 1996.

[103] K.-K. Sung. *Learning and Example Selection for Object and Pattern Detection*. PhD thesis, Massachusetts Inst. of Technology, 1996.

[104] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7:11–32, 1991.

[105] D.L. Swets and J. Weng. Using discriminant eigenfeatures for image retrieval. *IEEE Trans. Patt. Anal. Mach. Intell*, 18:831– 836, 1996.

[106] **G. Costache**, P. Corcoran, R. Mulryan, and E. Steinberg. Method and component for image recognition. *US Patent application no: 20060140455*, 2006.

[107] **G. Costache** and I.Gavat. Multimedia classifier. *ESA-EUSC Conference*, 2004.

[108] E. Steinberg **G. Costache**, R. Mulryan and P. Corcoran. In-camera person-indexing of digital images. *Consumer Electronics, 2006. ICCE '06. 2006 Digest of Technical Papers. International Conference on*, page 2, 2006.

[109] M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3(1):71–86, 1991.

[110] Yale University. Yale face database. *http://cvc.yale.edu/prjects/yalefaces/yalefaces.html*, 1997.

[111] V. Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5):988–1000, 1999.

[112] V. N. Vapnik. The nature of statistical learning theory. *Springer*, 1998.

[113] P. Viola and M. Jones. Robust real-time object detection. *Technical Report 2001/01, Compaq CRL*, 2001.

[114] Erald Vuçini, Muhittin Gökmen, and Meister Eduard Gröller. Face recognition under varying illumination, 2007.

[115] Y. Wang, T. Tan, and A. K. Jain. Combining face and iris biometrics for identity verification. *Proc. of 4th Int'l Conf. on Audio- and Video-Based Biometric Person Authentication AVBPA*, pages 805–813, 2003.

[116] G. Yang and T. S. Huang. Human face detection in complex background. *Pattern Recognition*, 27(1):53–63, 1994.

[117] M.-H. Yang and N. Ahuja. Detecting human faces in color images. *Proc. IEEE Int'l Conf. Image Processing*, 1:127–130, 1998.

[118] M.H. Yang, D. Kriegman, and N. Ahuja. Detecting faces in images: A survey. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 24, pages 34–58, 2002.

[119] L. Zhang, L. Chen, M. Li, and H. Zhang. Automated annotation of human faces in family albums, 2003. to be appeared in ACM Multimedia, 2003.

[120] W. Zhao, R. Chellappa, A. Rosenfeld, and P.J. Phillips. Face recognition: A literature survey. *ACM Computing Surveys*, pages 399–458, 2003.