Face Recognition Using Hidden Markov Models

Ferdinando Silvestro Samaria

Trinity College

A dissertation submitted
in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
at the University of Cambridge

Declaration

I hereby declare that the work described in this dissertation is my own original work, unless otherwise indicated in the text. Some of the work covered in this dissertation has appeared in the papers listed below, of which I am the first author. This dissertation does not exceed the limit in length set out by the Degree Committee and has not been submitted, in part or as a whole, for a degree or diploma or other qualification at any other university.

The length of this dissertation, including bibliography, footnotes, tables and equations, is approximately 28,000 words.

Ferdinando Silvestro Samaria
7 October 1994

List of Publications

F.Samaria and F.Fallside. Face Identification and Feature Extraction Using Hidden Markov Models. In *Image Processing: Theory and Applications*, edited by G.Vernazza, Elsevier, San Remo (Italy), June 1993.

F.Samaria and F.Fallside. Automated Face Identification Using Hidden Markov Models. *Proceedings of the International Conference of Advanced Mechatronics*, The Japan Society of Mechanical Engineers, Tokyo, August 1993.

F.Samaria. Face Segmentation For Identification Using Hidden Markov Models. *Proceedings of 1993 British Machine Vision Conference*, BMVA Press, Guildford, September 1993.

F.Samaria and S.Young. A HMM-Based Architecture For Face Identification. *Image and Vision Computing*, 12(8):537-543, Butterworth-Heinemann, October 1994.

F.Samaria and A.Harter. Parameterisation of a Stochastic Model for Human Face Identification. `To appear in:` *IEEE Workshop on Applications of Computer Vision*, Sarasota (Florida), December 1994.

Summary

This dissertation introduces work on face recognition using a novel technique based on Hidden Markov Models (HMMs). Through the integration of a priori structural knowledge with statistical information, HMMs can be used successfully to encode face features. The results reported are obtained using a database of images of 40 subjects, with 5 training images and 5 test images for each. It is shown how standard one-dimensional HMMs in the shape of top-bottom models can be parameterised, yielding successful recognition rates of up to around 85%.

The insights gained from top-bottom models are extended to pseudo two-dimensional HMMs, which offer a better and more flexible model, that describes some of the two-dimensional dependencies missed by the standard one-dimensional model. It is shown how pseudo two-dimensional HMMs can be implemented, yielding successful recognition rates of up to around 95%.

The performance of the HMMs is compared with the Eigenface approach and various domain and resolution experiments are also carried out. Finally, the performance of the HMM is evaluated in a fully automated system, where database images are cropped automatically.

Keywords: face recognition, face segmentation, automatic feature extraction, Hidden Markov Models, stochastic modelling.

Acknowledgements

I wish to thank my supervisor, Steve Young, for his valuable and close supervision. I am also very grateful to Andy Harter of Olivetti Research Ltd, for his continuous help and support.

I am grateful to many people in Cambridge, for the ideas discussed together: Andy Hopper and Frazer Bennett of Olivetti Research Ltd; Gábor Megyesi of the Department of Pure Mathematics and Mathematical Statistics; Tony Heap, Andy Ward, Rob Walker and Gavin Stark of the Computer Laboratory; Tat Jen Cham, Roberto Cipolla and Steve Hodges of the Department of Engineering. Many thanks also to Barney Pell of NASA Ames Research Centre and Junji Yamato of NTT Research in Yokosuka. Finally, I wish to remember my former supervisor, the late Prof. Frank Fallside, who originally supervised this work with ingenuity and enthusiasm.

Many thanks to all those who kindly allowed me to take pictures of them to build the ORL database of faces. I wish to thank Olivetti Research Ltd for providing the facilities to capture and store the database images.

This work was supported by a Trinity College Internal Graduate Studentship and an Olivetti Research Ltd CASE award. Their support is gratefully acknowledged.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Face recognition plays an important part in human activities. The way we interact with other people is firmly based on our ability to recognise them. One of the striking aspects of face identification in humans is its robustness. Humans are able to identify distorted images (as in the case of a caricature, as studied by Benson and Perrett [8]), coarsely quantised images (for example, the face in figure 1.1), faces with occluded details (a person wearing sun glasses) and even inverted face images, as reported by Diamond and Carey [21]. Humans perform the task of face recognition effortlessly and this has induced

Figure 1.1: Low resolution image of a famous scientist

some researchers to argue that the human brain contains a processing region dedicated to recognising faces. Humans are aware of their ability to recognise faces with ease. Early postage stamps carried the face of Queen Victoria, as this would make it easy to detect forgeries. Today's bank-notes still show the face of the Queen and other famous people on the reverse side.

Understanding the human mechanisms employed to recognise faces constitutes a challenge for psychologists and neural scientists. In addition to the cognitive aspects, understanding face recognition is important because the same underlying mechanisms could be used to build a system for the automatic identification of faces by machine. There are numerous applications for a robust automated recognition system. Gallery and Trew [25] investigated face recognition for the purpose of workstation security: their system would grab an image of the user at log-in time and then periodically compare the image of the person currently sitting at the terminal with the initial face. General security tasks, such as access control to buildings, can be accomplished by a face recognition system. Banking operations and credit card transactions could also be verified by matching the image encoded in the magnetic strip of the card with the person using the card at any time. Finally, a robust system could be used to index video-documents (video-mail messages, for example) and image archives. An image archive indexed in such a way would be useful for criminal identification by the police.

1.1 Biometric Measures

Banking institutions lose millions of pounds every year through fraudulent use of cash and credit cards, because the available identity checks (personal identity numbers and signatures) can be easily circumvented. As a result, there is increasing interest in collecting biometric measures of people to strengthen existing identity checks. Biometric systems are automated methods for identifying people through physiological or behavioural characteristics. With the advances in automated technology, biometric systems have expanded and now fuel an industry generating in excess of $100 million in sales of identity verification products (figures reported by Miller [51]). Presently, most people are still identified through their passport, driver's licence, or, while at work, through photo-badges. A more advanced system in use at Olivetti Research Ltd. in Cambridge is based on the Active Badge^{TM} system described by Want and Hopper [70]. Active badges are infra-red transmitters which allow the wearer to be located and recognised in a controlled environment. It is possible, in principle, to collect patterns of usage for each person and employ them as a biometric measure for recognition.

A biometric-based system was developed by Recognition Systems Inc., Campbell, California, as reported by Sidlauskas [64]. The system was called ID3D Handkey and used the three-dimensional shape of a person's hand to distinguish people. The side and top view of a hand positioned in a controlled capture box were used to generate a set of geometric features. Capturing took less than two seconds and the data could be stored efficiently in a 9-byte feature vector. This system could store up to 20,000 different hands.

Another well-known example of a biometric measure is that of fingerprints. Various institutions around the world have carried out research in the field, including the FBI. Fingerprint systems are unobtrusive and relatively cheap to buy. They are used in banks and to control entrance to restricted access areas. Fowler [24] has produced a short summary of the available systems.

Fingerprints are unique to each human being. It has recently been observed that the iris of the eye, like fingerprints, displays patterns and textures unique to each human and that it remains stable over decades of life as detailed by Siedlarz [65]. Daugman [19, 20] designed a robust pattern recognition method based on two-dimensional (2D) Gabor transforms to classify human irises.

Speech recognition also offers one of the most natural and less obtrusive biometric measures, where a user is identified through his or her spoken words. AT&T have produced a prototype that stores a person's voice on a memory card, details of which are described by Mandelbaum [50].

Least obtrusive of all, a face recognition system would allow a user to be identified by simply walking past a surveillance camera. Hutcheson [38] reported that NeuroMetric Vision Systems Inc., Pompano Beach, Florida, has designed a single-camera system with one DSP card, one frame grabber and a 5,000 face database, the cost of which is approximately \$30,000. However, no implementation or performance details are available.

1.2 Aims of this Work

Face recognition poses a challenging problem. Images that look alike to humans can be very difficult to match using computers, because of lighting, appearance and background changes. Much of the work in the computer vision literature has concentrated on improving the two basic approaches to the problem, namely geometric features and template matching. In this dissertation, a novel approach based on Hidden Markov Models (HMMs) is proposed. HMMs have been used with success in the area of speech recognition, where a word spoken twice by the same person may result in two substantially different signals. The ability of HMMs to handle this variability has prompted some computer vision researchers to experiment with HMMs in image recognition applications. To date, however, no work in face recognition using HMMs has been found and this dissertation investigates some of the aspects involved in classifying faces using this method. The work shows that, through the integration of a priori structural knowledge with statistical information, HMMs can successfully describe the content of face images.

The aims of this dissertation can be summarised as follows:

- To introduce a novel modelling technique for face images based on Hidden Markov Models.
- To demonstrate the characteristics of the proposed approach through a detailed collection of experiments.
- To study how the recognition performance is affected by the variation of model parameters.
- To extend the insights gained from experimenting with standard one-dimensional (1D) HMMs to pseudo-2D HMMs.
- To investigate how the HMM performance is affected by varying the image resolution and by representing images in the edge domain.
- To compare the performance of the HMM-based method with a well-known approach (Eigenfaces).
- To investigate the performance of the HMM approach in a fully automated system, in which face images are cropped automatically.

1.3 Structure of the Dissertation

This dissertation consists of nine chapters.

Chapter 2 surveys relevant literature in the field of face recognition. A brief account on the work done in psychology is presented first, followed by a more comprehensive description of the advances in the field of automated recognition.

Chapter 3 introduces the proposed approach based on HMMs. The HMM is defined first and other work by computer vision scientists using HMMs is briefly reviewed. The proposed HMM architecture and its motivations are then described.

Chapter 4 presents experimental results using basic 1D HMMs. Two model topologies are investigated. For ergodic models, only preliminary results are presented, as they do not appear to model the data successfully. For top-bottom models, the parameters are analysed through a series of detailed experiments.

Chapter 5 extends the insights gained from experimenting with 1D HMMs to pseudo-2D (P2D) HMMs. P2D-HMMs are defined and implemented for face recognition, giving improved experimental results which indicate that the model benefits from using a more efficient 2D representation.

Chapter 6 compares the HMM-based approach with the Eigenface approach. Experiments are carried out on the same database, drawing a statistical comparison between the best results obtained with HMMs and Eigenfaces.

Chapter 7 is a summary of some representation domain and resolution experiments. The results obtained by the best HMMs using edge-detected images and images at different resolutions are analysed.

Chapter 8 completes the experimental section of the dissertation. An automatic system for locating the head in uncropped images is presented. The system is used to crop images automatically. These images are then used to train and test a P2D-HMM, the results of which are described.

Chapter 9 concludes the dissertation, by summarising the results obtained and indicating the future directions of HMM-based face recognition.

Chapter 2

Summary Of Related Work

The task of recognising faces has attracted much attention both from psychologists and from computer vision scientists. This chapter reviews some of the approaches that researchers from both fields have investigated. The section on psychology presents some basic ideas and contains a number of useful references. However, a discussion of the models and techniques used in the field is beyond the scope of this dissertation. A more detailed section follows on progress in the field of automated face recognition, with a survey of some of the most popular and successful algorithms to date.

2.1 Face Recognition in Psychology

2.1.1 Describing face information

The recognition of familiar faces plays a fundamental role in our social interactions. Humans are able to identify reliably a large number of faces, and psychologists are interested in understanding the perceptual and cognitive mechanisms at the base of the face recognition process. A recent description of the progress made by psychologists can be found in Bruce [9]. The apparent complexity of the identification process and its high rate of success for humans have induced some researchers, for example Yin [77], to argue that neural specialisation has evolved to support a processor specific to faces. In support of this view, Perrett *et al.* [54] have reported the detection of special neurons responsive to faces in the cerebral cortex of monkeys.

Faces consist of the same elements (nose, mouth, eyes, etc.) and recognition of individuals happens when we discriminate between the same basic configurations. Each individual must therefore be distinguishable from others because of the way the basic element configuration varies, as pointed out by Young and Bruce [78]. The dichotomy in the kind of information carried by a face was also emphasised by Diamond and Carey [21] who distinguished between isolated and relational features. Isolated features are those features which can be described by themselves (e.g. eye colour, hair texture, etc.). Relational features are those features which describe aspects of the shape and how different facial elements relate to one another (e.g. the position of the nose relative to the eyes).

The description of relational features is useful to discriminate between different people, and also to determine different kinds of facial behaviour or expression for the same person. Various efforts were made to develop a system able to distinguish and model different types of facial expression. One such system was the Facial Action Coding System (FACS) proposed by Ekman and Friesen [22], and later used also by Rydfalk [60] to create a data-set describing a parameterised face. The system was based on the fact that every facial expression is the result of muscular action. A study of the anatomical basis of facial movement gave an insight into how facial muscles act to change the physical appearance of the face. "Action units" were used to model the different muscles in the face, with some muscles contributing to more that one action unit.

2.1.2 Recognition models

Hay and Young [32] proposed one of the early models to explain recognition of familiar faces based on "face recognition units" (FRUs). FRUs are units which will respond by reaching a threshold of excitation when a known face is seen. There is one FRU for each known face. Faces are firstly encoded using a suitable representation and then passed to the FRUs. The matching FRU gives access to the semantic information of the recognised person and the his or her name is generated. FRUs also formed the basis of a later model proposed by Bruce and Young [10].

Cantor and Mischel [12] analysed the decision process behind visual object categorisation. Three approaches to decide if a visual stimulus could be identified as a certain visual object were presented: (1) classical, (2) exemplar and (3) prototypical. The decision processes for each approach were summarised as follows:

1. The perceiver used a list of necessary attributes that described the visual object; recognition occurred if the perceiver was satisfied that the visual stimulus possessed all the necessary attributes.

2. The perceiver used an exemplar model which was a collection of known instances of the visual object; a visual stimulus was matched on the basis of its similarity to the known instances.

3. The perceiver used a prototype model which was an abstract image or set of features (prototype) of the visual object; a visual stimulus was matched on the basis of its similarity to the prototype.

In their work on person perception, Cantor and Mischel restricted their attention to applications of the prototype approach.

Light *et al.* [49] endorsed the view that natural categories such as those of faces are better defined in terms of a prototype. A prototype face is created from all the faces an individual encounters in daily life and each new face is encoded by reference to the prototype. The various face features are conceived as axes extending in all directions from the prototype, which represents the origin of this feature space. A typical face, which will have a close resemblance to the prototype, will be located close to the origin. Since typical faces are assumed to be more common than distinctive faces, the space in the vicinity of the origin will have a high density. In contrast, distinctive faces will be located at some distance from the prototype and are less likely to be in the proximity of other faces. Light *et al.* presented a set of four studies which showed evidence that recognition memory for faces similar to a prototype was inferior to memory for unusual faces. Distinctiveness was therefore deemed to be an influencing factor for the successful recognition of faces. Further work on distinctiveness was presented by Winograd [72] and Shepherd *et al.* [63].

2.2 Automatic Face Recognition Survey

This section presents a concise literature survey for the field of automatic face recognition. Much work has been done in the field. However, the comparison and description of the various methods is often a complex task because the results reported are obtained using different image sets and because there is no common terminology to describe the methods. In order to create a coherent presentation framework, the following terminology will be used throughout this thesis. The task under discussion usually consists of identifying an unknown face image from a set of known images. A database of face images consists of images of S distinct subjects. In general, in order to balance the database, the S subjects should span gender, ethnic origin and age as evenly as possible. However, this is often very difficult to achieve because of a shortage of subject availability and sometimes computer storage limitations. The database should contain the same number I of images for each subject. In this case the overall size of the database would be $I \times S$. Of the I images available for each subject, J are used for training and K are used for testing. The size of the training set is therefore $J \times S$ and the size of the test set is $K \times S$. Also $J + K = I$ and the training set has an empty intersection with the test set. Recognition is usually performed by scoring a test image against the images in the training set using some distance metric and selecting the highest score. A set of experiments $\mathcal{E}$ can be described using the shorthand notation:

$$\mathcal{E} = (S, J, K, r)$$

where r is the correct recognition performance expressed as a percentage between 0 and 100.

Different techniques have been proposed over the last 25 years. Two general strategies for solving the problem of computer face recognition have been identified in the literature, as pointed out by Brunelli and Poggio [11] and Robertson and Craw [59]: geometric, feature-based matching and template matching. The approaches reviewed in the following sections are therefore collected in these two categories. Most of these approaches are constrained by a number of assumptions for the training and test data, some of which have been summarised by Samal and Iyengar [61]. The most common assumptions are:

- The face images are in either frontal or profile view;
- The face is in upright position;
- No or very little tilt is tolerated;
- No occlusions, facial hair, glasses or scars;
- Lighting and background are controlled;
- Most of the test cases are white males;
- The size of the test set is limited to at most a few hundred.

2.2.1 Early recognition work

Automatic face recognition has attracted much interest from the computer science community since the late 1960s. However, initial work in automatic face processing dates back to the end of the 19th century, as reported by Benson and Perrett [7]. In his lecture on personal identification at the Royal Institution on 25 May 1888, Sir Francis Galton [26] F.R.S., an English scientist, explorer and a cousin of Charles Darwin, explained that he had "frequently chafed under the sense of inability to verbally explain hereditary resemblances and types of features". In order to relieve himself from this embarrassment, he "took considerable trouble and made many experiments". He described how French prisoners were identified using four primary measures (head length, head breadth, foot length and middle-digit length of the foot and hand respectively). Each measure could take one of three possible values (large, medium or small), giving a total of 81 possible primary classes. Galton felt it would be advantageous to have an automatic method of classification. For this purpose, he devised an apparatus, which he called a *mechanical selector*, that could be used to compare measurements of face profiles. In choosing the best measures to describe the form of the profile, Galton reported that most of the measures he had tried were fairly efficient.

2.2.2 Geometric, feature-based approach

The idea of comparing measurements introduced by Galton has also been used in more recent work based on computing a set of distinctive features from the picture of a face. The features are usually obtained from either profile or front view images.

Profile features

Harmon *et al.* [31] proposed an approach based on geometric profile features of the human face. The features were calculated from some automatically placed fiducial marks along the profile trace. Starting from a database of 112 subjects with three training images and one test image for each subject, they represented a face using a 17-element feature vector. A success rate of 96% was obtained. Najman *et al.* [52] also used geometric features from profile images. A profile outline was constructed using between 8 and 100 control points. Tests were carried out on a database of 10 subjects, with 31 training images and 10 test images for each subject. Three classification methods were tried: Principal Component Analysis followed by Quadratic Discrimination, k-Nearest Neighbour and Gradient Back Propagation. Success rates of about 90% were reported. Wu and Huang [75] used profile geometric features in their work with six control points obtained using a cubic B-spline. The database contained 18 subjects with three training images and one test image for each subject. Nearly 100% recognition rates were obtained.

Front view features

In one of the first attempts at automatic face recognition, Kanade [40] devised a system that would extract 16 front view geometric features (subsequently reduced to 13). He used a database of 20 subjects with one training image and one test image for each subject, reporting a 75% correct recognition performance. Brunelli and Poggio [11] have recently implemented a geometric, feature-based recogniser loosely based on Kanade's work. The recogniser was tested on a larger database of 47 subjects and recognition rates of about 90% were reported. Another front view feature-based approach was implemented by Wong *et al.* [73] using various distances (eye to eye, left and right eye to nose, nose to left and right edge) as features. Perfect recognition results were reported on a database of only 6 people.

Mixed front view and profile features

In their early work, Goldstein *et al.* [29] used a set of 34 mixed front view and profile features (subsequently reduced to 22). The features comprised amongst others, hair length, hair texture, nose length, mouth width and chin profile. Features were scored on a scale of 1-5 (low-medium-high) and were located manually. The model predicted that for a population of 255 subjects 6 features were sufficient for identification.

2.2.3 Template matching

Another technique often used consists of representing an image as single or multiple arrays of pixel values. The arrays are compared with single or multiple templates representing the faces in the training set via a suitable metric. The features of interest can be located manually or by using a more sophisticated automatic approach based on a multi-layer perceptron as detailed in Hutchinson and Welsh [39], a deformable template as described by Yuille *et al.* [80] or an active contour model (snake) as reported by Huang and Chen [37] and as originally described by Kass *et al.* [41].

Principal component analysis

The simplest version of template matching is obtained when the whole face image is used as a single template. A test image is recognised by computing its distance (in Euclidean terms, for example) from the templates generated from the images in the training set and selecting the closest match. The Karhunen-Loève procedure of Kirby and Sirovich [42] and the Principal Component Analysis approach of Turk and Pentland [69] are based on this simple template matching method. The array and the template, however, are not the original face images but their projection onto an optimal coordinate system. The set of basis vectors which make up this coordinate system are the eigenvectors of the covariance matrix of the ensemble of training faces. Using this method, Turk and Pentland reported successful recognition rates of up to 96% with a database of 16 subjects. This method will be analysed in more detail and implemented for comparison purposes in chapter 6.

Isodensity line maps

A different template-based approach was proposed by Nakamura *et al.* [53]. The technique they presented made use of grey-level isodensity line maps to represent face images. Summarised in their own words, if the brightness of an image is viewed as the height of a mountain, then an isodensity line corresponds to contour lines of equal altitude. A database of 10 subjects with one training image and one test image for each subject was used. Three subjects wore glasses, two men had a thin beard and two women had different make-up and hair styles in the test and training images. Recognition experiments were carried out and perfect recognition rates were reported.

Multiple template correlation methods

One of the first studies based on multiple template representation was carried out by Baron [4]. A database of 42 subjects was used and each was represented by up to five manually selected face features (full face, mouth, right eye, chin and hair), and each face feature contained up to four distinct templates. A total of up to 20 pictorial templates were stored for each subject, with each template being a 15x16 array of pixels. A test image was first reduced to a 15x16 full face array and then compared with each full face template in the training set. If the correlation value between the reduced test image and one of the full face templates exceeded a threshold of recognition, the test image was recognised as the corresponding subject. If the correlation value fell between the threshold of recognition and a lower value called the threshold of recall, then the other face features were recalled and used for recognition. If for at least three out of four of the features the correlation value exceeded the threshold of recall, the test image was recognised as the current subject. Baron reported a recognition accuracy of 100%.

More recently Brunelli and Poggio [11] presented results based on a similar approach. They used a database of 47 subjects, where each subject was represented by a full frontal image and a set of four templates (eyes, nose, mouth and the whole face). Recognition of a test image was performed by computing a normalised cross correlation for each template and by finding the highest cumulative score. Perfect recognition rates were reported.

Vector quantised templates

Sutherland *et al.* [67] used a template-based approach, where each of the original eight feature templates they selected was substituted with an approximately similar template drawn from a code-book via vector quantisation. Various algorithms can be used to generate useful code-books and two such algorithms were presented by Ramsay *et al.* [58]. Using a database of 30 subjects with 10 training images and 10 test images for each subject, a successful recognition rate of 89% was reported.

Neural network based template matching

Templates have been used as input to neural network based systems. Allinson *et al.* [2] used a 32x32 full image template and two 64x32 templates for the eye and mouth regions respectively. These templates were used as inputs to Kohonen's [43] self-organising feature maps. The maps produced a topology which preserved the structure of the input templates. The maps were used as input to a multi-layer perceptron which carried out the classification. Other work by Cottrell and Fleming [16] studied the performance of a network that automatically extracted features (the output of the hidden units) from a 64x64 full face template and input them to a one-layer network for identity and also gender classification. Test images were perfectly identified with a database of 11 subjects. A gender recognition success performance of 37% was reported.

Stonham [66] detailed experiments on face recognition using a general purpose pattern recognition machine called WISARD. A database of 16 subjects was used and full image 153x214 templates were input to a self-adapting single layer network. Subjects were asked to appear before a camera, face on, for approximately 20 seconds. On average, 200-400 images were required to complete the training. Real time testing results were reported with error free recognition rates.

Hybrid template methods

There may be advantages in using a combination of both geometric features and template information. In the work of Craw and Cameron [18] and in the work of Craw [17], a hybrid approach was proposed based partly on template matching and partly on geometric features. A face was modelled using a mask with 59 control points and each face was described by two vectors:

1. A shape vector, i.e. the location of each control point. The shape vector contained information about the geometric features of the face.

2. A texture vector, i.e. the grey-levels used to texture the face after the control points were aligned with the average face. The information contained in the texture vector was the equivalent of that of a template.

Successful identification results were reported even when there were significant differences between test and training images.

Lanitis *et al.* [47] also used a combination of shape and grey-level information to encode the appearance of human faces. They experimented with a database of 30 subjects, with 10 training and 10 test images for each subject. Faces were modelled using three methods:

1. A flexible shape model based on a point distribution model (an introduction to this technique can be found in Cootes *et al.* [15]). This model captured shape variation and could also be used for locating the face in the image.

2. A shape-free grey-level model, obtained by deforming and aligning each training face to the mean face.

3. A local grey profile model, consisting of a large number of local profiles, taken along the perpendicular to the shape model boundary at each shape model point.

Results were best when all three methods were used simultaneously and successful recognition rates of 92% were reported.

2.2.4 Summary of recognition results

Comparing and interpreting recognition results of different face recognition systems is a complex task, because experiments are usually carried out on different data sets. This implies that the size of the database and the constraints applied to the data are different for each experiment. Robertson and Craw [59] discussed the testing of face recognition systems and pointed out that systems which worked well with constrained data, might not perform equally well with data supplied externally. The following gives a number of useful questions when reviewing different approaches:

- Were expression, head orientation and lighting conditions controlled?
- Were the subjects allowed to wear glasses and have beards or other facial marks?
- Was the subject sample balanced? Were gender, age and ethnic origin spanned evenly?
- How many subjects were there in the database? How many images were used for training and testing?
- Were the faces and the face features located manually? Was the scaling controlled?

Answering the above questions contributes to building a better description of the constraints within which each approach operated. This helps to make a fairer comparison between different sets of experimental results. However, the most direct and reliable comparison between two or more approaches is obtained by experimenting with the same database. Brunelli and Poggio [11], for example, implemented a feature-based and a template-based method, and tested and compared them using the same database. Ideally, databases should be made available for other researchers to use.

Table 2.1 summarises the recognition performances of the approaches discussed in the previous sections. The experimental results are reported using the $\mathcal{E}$-shorthand notation introduced in section 2.2. The table is shown for easy reference, but a comparison between the different systems in the terms expressed above is beyond the scope of this dissertation.

Reference	**Experimental Results**				
Harmon *et al.* [31]	$\mathcal{E} = ($	112,	3,	1,	96%)
Najman *et al.* [52]	$\mathcal{E} = ($	10,	31,	10,	90%)
Wu and Wuang [75]	$\mathcal{E} = ($	18,	3,	1,	100%)
Kanade [40]	$\mathcal{E} = ($	20,	1,	1,	75%)
Nakamura *et al.* [53]	$\mathcal{E} = ($	10,	1,	1,	100%)
Sutherland *et al.* [67]	$\mathcal{E} = ($	30,	10,	10,	89%)
Lanitis *et al.* [47]	$\mathcal{E} = ($	30,	10,	10,	92%)
Brunelli and Poggio [11]$^{(feature)}$	$\mathcal{E} = ($	47,	—,	—,	90%)
Brunelli and Poggio [11]$^{(template)}$	$\mathcal{E} = ($	47,	—,	—,	100%)
Wong *et al.* [73]	$\mathcal{E} = ($	6,	—,	—,	100%)
Turk and Pentland [69]	$\mathcal{E} = ($	16,	—,	—,	96%)
Baron [4]	$\mathcal{E} = ($	42,	—,	—,	100%)
Cottrell and Fleming [16]	$\mathcal{E} = ($	11,	—,	—,	100%)
Stonham [66]	$\mathcal{E} = ($	16,	200-400,	—,	100%)

Table 2.1: Summary of surveyed recognition results

Chapter 3

The Proposed HMM Approach

In recent years, research in the field of automated face recognition has focussed on feature-based and template-based methods, as described in chapter 2. Researchers have spent much effort trying to improve these two basic methods.

A novel approach based on HMMs is investigated in this dissertation. The HMM method is based on matching image templates to a chain of states of a doubly-embedded stochastic model. This chapter outlines the basic principles of HMMs and explains how they can be used for face recognition. The sections are organised as follows: first a general overview of HMMs is presented, then some HMM applications in computer vision are briefly reviewed and finally the proposed HMM-based architecture for face recognition is detailed.

3.1 Introducing HMMs

HMMs are generally used for the stochastic modelling of non-stationary vector time-series. As such, they have an immediate and obvious application in speech processing, particularly recognition, where the signal of interest is naturally represented as a time-varying sequence of spectral estimates. Therefore, much of the development of HMMs in recent years has been done within the speech area. Rabiner [56] presented a comprehensive tutorial on HMMs, details of which are summarised in the next section. Moreover, a good modern treatment of HMM-based speech recognition can be found in Rabiner and Juang [57].

3.1.1 One-dimensional HMM definition

A HMM provides a statistical model for a set of observation sequences. In speech applications, the observations are sometimes called *frames*, and the two terms will be used interchangeably throughout this dissertation. Let a particular observation sequence have length T and be denoted as $\mathbf{o}_1 \dots \mathbf{o}_T$. A HMM consists of a sequence of states numbered 1 to N and it is best understood as a generator of observations. The states are connected together by arcs and each time that a state j is entered, an observation is generated according to the multivariate Gaussian distribution $b_j(\mathbf{o}_t)$ with mean $\boldsymbol{\mu}_j$ and covariance matrix $\mathbf{V}_j$ associated with that state. The arcs themselves have transition probabilities associated with them such that a transition from state i to state j has probability a_{ij}. The probability of the model starting in state j is π_j. A HMM is thus defined by the following set of parameters:

- N is the number of states in the model.
- $A = \{a_{ij} : 1 \leq i, j \leq N\}$ is the state transition matrix.
- $B = \{b_j(\cdot) : 1 \leq j \leq N\}$ is the output probability function.
- $\Pi = \{\pi_j : 1 \leq j \leq N\}$ is the initial state probability distribution.

In shorthand notation, a given model can be summarised as $\lambda = \{N, A, B, \Pi\}$. All of the experimental work described here is carried out with the HTK software package described by Young [79], which adopts the convention that a HMM with N-2 states is represented by a model with N states, always starting in state 1 and ending in state N. Both these states are non-emitting and only states from 2 to N-1 emit. In this way, the parameter Π is not used explicitly, but is absorbed by the transition probability matrix. The equations that follow, however, are based on the work by Rabiner [56] and make use of the initial state probability distribution.

3.1.2 Model training and recognition

For a given model λ, the joint likelihood of a state sequence $Q = q_1 \dots q_T$ and the corresponding observation sequence $\mathbf{O} = \mathbf{o}_1 \dots \mathbf{o}_T$ is given by multiplying each transition probability by each output probability at each step t as follows:

$$P(\mathbf{O}, Q|\lambda) = \pi_{q_1} b_{q_1}(\mathbf{o}_1) \left[\prod_{t=2}^{T} a_{q_{t-1},q_t} b_{q_t}(\mathbf{o}_t) \right] \tag{3.1}$$

In practice, the state sequence is unknown, i.e. it is *hidden* and so equation 3.1 cannot be evaluated. However, the likelihood $P(\mathbf{O}|\lambda)$ can be evaluated by summing over all possible state sequences:

$$P(\mathbf{O}|\lambda) = \sum_{Q} P(\mathbf{O}, Q|\lambda) \tag{3.2}$$

The key attraction of HMMs is that there is a simple procedure for finding the parameters λ which maximise equation 3.2. This procedure is usually referred to as Baum-Welch re-estimation introduced by Baum [5] and it depends for its operation on the forward-backward algorithm. The latter allows the so-called *forward* probability $P(\mathbf{o}_1 \ldots \mathbf{o}_t, q_t = j|\lambda)$ and the *backward* probability $P(\mathbf{o}_{t+1} \ldots \mathbf{o}_T|q_t = j, \lambda)$ to be found efficiently via a simple iteration. The forward and backward variables $\alpha_t(j)$ and $\beta_t(j)$ are defined as follows:

$$\alpha_t(j) = P(\mathbf{o}_1 \ldots \mathbf{o}_t, q_t = j|\lambda) \tag{3.3}$$

$$\beta_t(j) = P(\mathbf{o}_{t+1} \ldots \mathbf{o}_T|q_t = j, \lambda) \tag{3.4}$$

The variables can then be found inductively:

1. Initialisation

$$\alpha_1(j) = \pi_j b_j(\mathbf{o}_1), \qquad 1 \leq j \leq N \tag{3.5}$$

$$\beta_T(j) = 1, \qquad 1 \leq j \leq N \tag{3.6}$$

2. Induction

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(\mathbf{o}_{t+1}), \qquad 1 \leq t \leq T-1, \quad 1 \leq j \leq N \tag{3.7}$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j), \qquad t = T-1, \ldots, 1, \quad 1 \leq i \leq N \tag{3.8}$$

3. Termination

$$P(\mathbf{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i) = \sum_{i=1}^{N} \beta_1(i) \tag{3.9}$$

The direct calculation of $P(\mathbf{O}|\lambda)$ according to equation 3.2 involves on the order of TN^T calculations, while the calculation through the forward-backward algorithm of equation 3.9 only requires on the order of TN^2 calculations, as detailed by Rabiner [56].

The product of the forward and backward probabilities when normalised yields the probability of occupying state j at step t given the observation sequence $\mathbf{O}$. This variable is defined as $\gamma_t(j) = P(q_t = j|\mathbf{O}, \lambda)$ and can be simply calculated as:

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)} \tag{3.10}$$

With this state occupation probability, a new set of HMM parameters for each state j can be found, essentially by computing weighted averages. The model described by the new set of parameters is defined as $\overline{\lambda} = \{\overline{\Pi}, \overline{A}, \overline{B}\}$. To estimate the transition parameters, a related quantity $\xi_t(i,j)$ is defined as the probability of being in state i at time t and in state j at time $t+1$, given the observation sequence and the model:

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j|\mathbf{O}, \lambda) \tag{3.11}$$

Using the definitions of the forward and backward probabilities, this can be expressed as:

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(\mathbf{o}_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i)a_{ij}b_j(\mathbf{o}_{t+1})\beta_{t+1}(j)} \tag{3.12}$$

Using the concept of counting occurrences, the model parameters for $\overline{\lambda}$ can be re-estimated as follows:

$$\overline{\pi}_i = \gamma_1(i) \tag{3.13}$$

$$\overline{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{3.14}$$

$$\overline{\boldsymbol{\mu}}_i = \frac{\sum_{t=1}^{T} \gamma_t(i) \cdot \mathbf{o}_t}{\sum_{t=1}^{T} \gamma_t(i)} \tag{3.15}$$

$$\overline{\mathbf{V}}_i = \frac{\sum_{t=1}^{T} \gamma_t(i) \cdot (\mathbf{o}_t - \overline{\boldsymbol{\mu}}_i)(\mathbf{o}_t - \overline{\boldsymbol{\mu}}_i)'}{\sum_{t=1}^{T} \gamma_t(i)} \tag{3.16}$$

where prime denotes vector transpose, and $\overline{\boldsymbol{\mu}}_i$ and $\overline{\mathbf{V}}_i$ are the estimates of the mean and covariance matrix[1] of the Gaussian output probability function for state i. Baum and Sell [6] showed that the new model $\overline{\lambda}$ was equally or more likely than λ, i.e. $P(\mathbf{O}|\overline{\lambda}) \geq P(\mathbf{O}|\lambda)$. Based on the above estimates, $\overline{\lambda}$ is used iteratively in place of λ and the process is repeated until the parameter estimates converge to a critical point, which is at a local maximum of $P(\mathbf{O}|\lambda)$. All of the above has been described in terms of a single observation sequence, but it is trivial to extend this to maximise over a set of observation sequences, by summing the numerators and denominators in the re-estimation formulae over all sequences. Thus, given one or more training observation sequences known to come from a specific class,

[1] The re-estimation equation for the covariance matrix in HTK uses $\boldsymbol{\mu}_i$ instead of $\overline{\boldsymbol{\mu}}_i$. This is found to make no difference, provided the Baum-Welch iteration converges

the parameters of a HMM can be estimated to form a statistical model to represent that class, and a different model is formed for each class.

In order to use HMMs for recognition, an observation sequence is obtained from the test signal and then the likelihood of each HMM generating this signal is computed. The HMM which has the highest likelihood then identifies the test signal. This likelihood should strictly be the total likelihood as defined in equation 3.2. However, in practice, it is more convenient to find the state sequence which maximises equation 3.1 and use the corresponding maximum likelihood instead. This maximisation, known as the Viterbi algorithm described by Forney [23], is a simple dynamic programming optimisation procedure. The advantage of using it instead of the full likelihood computed by the forward-backward algorithm is that it also yields the maximum likelihood state sequence as a by-product. This can be useful in determining which regions of the observation sequences are being modelled by each state. The best score along a single path (intended as a sequence of states) at time t, which accounts for the first t observations and ends in state j, is defined by the quantity $\delta_t(j)$ as follows:

$$\delta_t(j) = \max_{q_1,\ldots,q_{t-1}} \{P(q_1 \ldots q_t = j, \mathbf{o}_1 \ldots \mathbf{o}_t \mid \lambda)\} \tag{3.17}$$

An inductive procedure can be used to calculate the values of $\delta_t(j)$ as follows:

1. Initialisation

$$\begin{aligned} \delta_1(j) &= P(q_1 = j, \mathbf{o}_1 \mid \lambda) \\ &= P(q_1 = j \mid \lambda) P(\mathbf{o}_1 \mid q_1 = j, \lambda) \\ &= \pi_j b_j(\mathbf{o}_1), \qquad 1 \le j \le N \end{aligned} \tag{3.18}$$

2. Induction

$$\delta_{t+1}(j) = \max_{1 \le i \le N} [\delta_t(i) a_{ij}]\, b_j(\mathbf{o}_{t+1}), \qquad 1 \le t \le T-1, \quad 1 \le j \le N \tag{3.19}$$

3. Termination

$$P^\diamond = \max_{1 \le j \le N} [\delta_T(j)] \tag{3.20}$$

The quantity $P^\diamond$ is the joint probability of the optimal state sequence and the observation sequence $\mathbf{O}$ given the model λ. This maximum likelihood value is used for recognition. An array $\psi_t(j)$ is defined to keep track of the states that maximise equation 3.17. The maximum likelihood state sequence $Q = q_1 \ldots q_T$ can be found by backtracking through $\psi_t(j)$ as follows:

$$q_T = \arg \max_{1 \leq i \leq N} [\delta_T(i)]$$

For $t = T - 1$ to 1

$$q_t = \psi_{t+1}(q_{t+1})$$

3.2 HMMs in Vision

HMMs have been used mostly in speech recognition applications, an area where they have been studied in depth and where they are now a well-established technique. As previously pointed out, HMMs model the statistical properties of 1D observation sequences and speech data is naturally 1D along the time axis. HMMs have been successful for speech and researchers in the field of computer vision have recently started to use them for image recognition problems.

He and Kundu [33] used continuous density HMMs combined with an autoregressive data model to classify closed 2D shapes. Each shape was represented using a 1D sequence of radii from the centre of gravity of the shape to the shape contour as shown in figure 3.1. Radii were chosen spaced at an equal curve length along the contour. Each radius was predicted using a linear combination of m previous radii, plus a constant term and an error term. The sequence was divided into T segments with l elements. Each radius feature vector consisted of the m autoregressive coefficients for the current radius, a ratio of the constant term to the error term and the current segment mean. They experimented with eight classes of shapes and trained a distinct HMM for each class using 20 class samples. A further 10 class samples were used for testing and shape recognition accuracy of up to 100% was reported.

Chen and Kundu [14] proposed a combination of quadrature mirror filter (QMF) banks and continuous density HMMs for image texture identification. The QMF bank was used to implement the wavelet transform of textures. A set of features was extracted from the statistics based on the first-order distribution of grey levels of the subband images. These

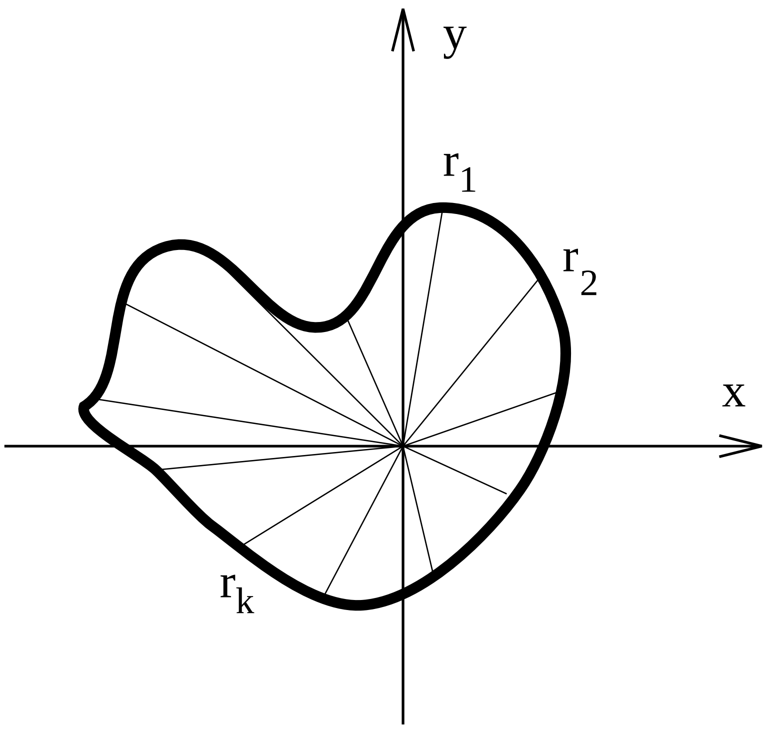


Figure 3.1: Radii sampling technique

subband features were arranged in a sequence starting from the lowest frequency band to the highest frequency band. The sequence was used to train one HMM for each type of texture. Successful recognition rates of up to 93% were reported.

Yamato *et al.* [76] chose discrete HMMs to model human action sequences. They analysed actions of a tennis player by modelling 6 tennis strokes using a different HMM for each one. The tennis strokes were extracted from a video sequence and were quantised into a discrete symbol sequence. With a 36-state model, they achieved successful recognition rates of over 90% if both the training and the test strokes were performed by the same player. For data from different players, the rate dropped to about 60%.

Another application of discrete HMMs was studied by Kundu *et al.* [45] for handwriting recognition applications. One HMM was built to model the letters of the alphabet. Each letter was identified with one of the states of the discrete HMM. For a given test word, each unknown letter in the word was transformed into its representative symbol from a code-book of symbols using a minimum distance criterion. Given the observed symbols, the best sequence of states revealing the letters that made up the test word was determined through one run of the Viterbi algorithm.

Various papers are reported on applications of HMMs to optical character and text recognition. Levin and Pieraccini [48], Agazzi *et al.* [1] and Kuo and Agazzi [46] presented work on text recognition using enhanced planar HMMs. A more detailed account of their work and a report on enhanced planar HMMs are presented in chapter 5.

3.3 Proposed Architecture

In this dissertation, the problem of face identification is addressed from the perspective of statistical pattern recognition. Intuitively a face can be divided into a number of regions such as the mouth, eyes, nose, etc., and if these could be located reliably, then standard pattern matching techniques could be applied to each region individually to compute an overall distance metric. However, accurate location is in practice very difficult. Moreover, the precise demarkation of the regions is fuzzy since it is unclear where, for example, the mouth region ends and the chin begins.

A potential solution to the above-mentioned problem is to associate facial regions with the states of a continuous density HMM. This allows the boundaries between regions to be represented by probabilistic transitions between states and the actual image within a region to be modelled by a multivariate Gaussian distribution. In the general case, the HMM would need to be 2D. However, in the chapters that follow, the assumption is made that a first-order approximation can be used where the facial regions are either restricted to horizontal bands, or are modelled by a pseudo-2D topology. In both cases, simple 1D HMMs can be used.

In the work described in this dissertation, a model is trained with 5 face images of the same subject. Each image generates an observation sequence $\mathbf{O} = \mathbf{o}_1 \ldots \mathbf{o}_T$. An observation $\mathbf{o}_t$ is obtained from a block of pixels in the 2D image by means of a sampling window that scans the image in some order as illustrated in figure 3.2. The pixels in the sampling window are arranged in a column-vector $\mathbf{o}_t$ containing their intensity level values. The sampling process determines how successful a model can be and is described in more detail in the chapters that follow. HMMs represent the statistical distribution of all observation sequences associated with a particular class or in this context with a particular subject. When concerned with face images, a number of different views and expressions of each face can be combined in a single statistical model. Since the HMM associates states with the quasi-stationary regions of its observation sequences, it offers a way of automatically locating and utilising the regions of a face which are important

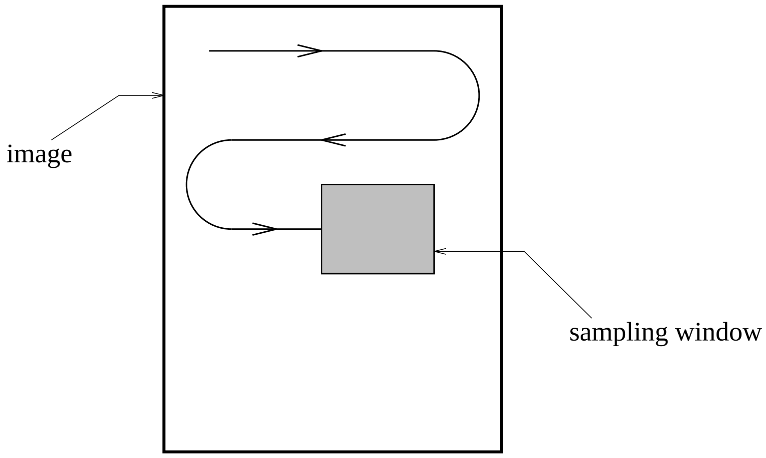


Figure 3.2: Sampling technique for 1D HMM

for identification. In subsequent parts of this dissertation, these regions will be referred to as features. For any given model, the features can be visualised by displaying the mean vector of the model state distributions as a bit-map. Furthermore, the optimal state sequence obtained as a by-product of the Viterbi algorithm can be mapped back to the image, showing how the face image is segmented into face regions. In this way, the segmentation of the face into regions useful for identification is achieved automatically. Other methods for feature location, such as those mentioned in chapter 2 based on neural networks, deformable templates and active contour models, generally require more guidance, sometimes even a substantial initial guess to find features successfully. The features obtained by the HMM are the result of a stochastic optimisation and are not guaranteed to yield the same features as those used by humans. However, by modelling the topology of the HMM after the structure of the face, the features obtained match in most cases those used by humans, as will be shown in the course of this dissertation.

Chapter 4

Experiments With One-Dimensional HMMs

This chapter details experimental results obtained using a 1D HMM. Two simple, basic HMM topologies are investigated: ergodic and top-bottom. The ergodic topology generates a simple model with few constraints on the data. This model makes no use of structural information (i.e. the fact that it is known that the image contains a face). Because of the lack of constraints, the modelling of the data is less successful and only limited experimental results are presented (with model parameters chosen based on subjective intuition). The concept of a top-bottom HMM is introduced and top-bottom models are analysed in more detail, as it is believed that these models represent facial information in a more natural way. The parameterisation of top-bottom HMMs is investigated through a comprehensive set of experiments. Both ergodic and top-bottom models are tested using the same database, which is described in the section that follows.

4.1 Experimental Setup

In order to study the different HMM topologies and parameterisations, a number of experiments with various models were run, the results of which are reported to compare the models. All the experiments were carried out using the same database to provide fair grounds for comparison. In the next sections the database used for the experiments is described and a brief description of the training and testing procedures are presented.

4.1.1 Description of the database

The database described in this section is used throughout this dissertation and will be referred to as the Olivetti Research Ltd (ORL) database of faces[1]. The database consists of 400 images, 10 each of 40 different subjects. The subjects are either Olivetti employees or Cambridge University students. The age of the subjects ranges from 18 to 81, with the majority of the subjects being aged between 20 and 35. There are 4 female and 36 male subjects. Subjects were asked to face the camera and no restrictions were imposed on expression; only limited side movement and limited tilt were tolerated. For most subjects the images were shot at different times and with different lighting conditions, but always against a dark background. Some subjects are captured with and without glasses. The images have been manually cropped and rescaled to a resolution of 92x112, 8-bit grey levels. Five images of each subject were used for training and five for testing, giving a total of 200 training and 200 test images. In order to compare different models, error rates were calculated for each tested model. The error rates are expressed as percentages and are obtained by dividing the number of misclassified test images by 200.

4.1.2 Training and testing procedures

All the HMM-based experiments reported throughout this dissertation were carried out using the *HTK: Hidden Markov Model Toolkit V1.3* developed by Young [79] at the Cambridge University Engineering Department. The training process for each of the S subjects in the database consists of the following steps which are summarised in the diagram of figure 4.1:

1. J training images are collected for the kth subject in the database and are sampled generating J distinct observation sequences.

2. A common prototype HMM model λ_0 is constructed with the purpose of specifying the number of states in the HMM, the state transitions allowed and the size of the observation sequence vectors.

3. A set of initial parameter values using the training data and the prototype model are computed iteratively. On the first cycle, the data is uniformly segmented and matched with each model state. On successive cycles, the uniform segmentation is replaced by Viterbi alignment. The outcome of this process is an initial HMM estimate $\lambda_e^{(k)}$ which is used as input to the re-estimation stage.

[1] The ORL database of faces is available via anonymous `ftp` from `ftp.cam-orl.co.uk` and is stored in `pub/data/orl_faces.tar.Z`

4. HMM parameters are re-estimated using the Baum-Welch method. The model parameters are adjusted so as to locally maximise the probability of observing the training data, given each corresponding model. The outcome of this process is the HMM $\lambda^{(k)}$ which is used to represent subject k in the database.

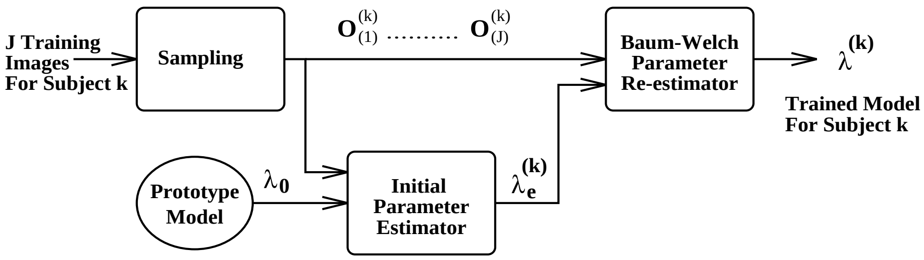


Figure 4.1: Block diagram of training technique

Recognition is carried out via a simple Viterbi recogniser. A collection of HMMs each representing a different subject is matched against the test image and the highest match is selected. The recognition process consists of the following steps which are summarised in the diagram of figure 4.2:

1. The unknown test image is sampled generating an observation sequence $\mathbf{O}_{test}$.

2. The observation sequence is matched against each face model by calculating the model likelihoods:

$$P(\mathbf{O}_{test} \mid \lambda^{(k)}), \;\; 1 \leq k \leq S$$

   In practice, a maximum likelihood value is used instead, as described in section 3.1.2

3. The model with the highest likelihood is selected and this model reveals the identity of the unknown face.

4.2 HMM Topology

A fully connected 2D HMM would be desirable for modelling a 2D image. However, the computational complexity for a fully connected 2D network is exponential as discussed by Levin and Pieraccini [48]. The methods presented in the sections that follow aim to show how to convert 2D images into 1D sequences useful for 1D HMM analysis. Chapter 5 of this dissertation is dedicated to the analysis of a pseudo-2D lattice of HMM states.

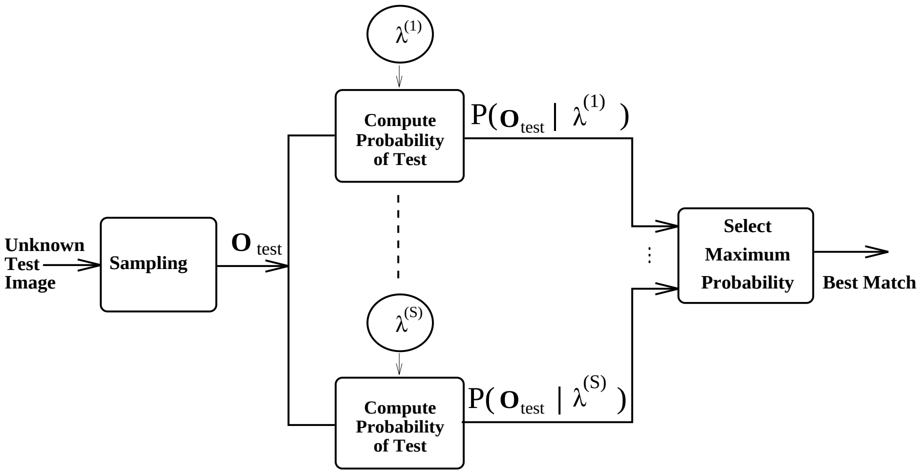


Figure 4.2: Block diagram of face recogniser

The sampling techniques introduced in the following sections serve the purpose of extracting a 1D observation sequence from a 2D image. The observation sequence is constructed by extracting blocks of pixels from an image using a sampling window. Each observation is a column-vector containing the intensity levels of the pixels inside the window. The sequence is formed by scanning the image in some order. The way in which the sampling window scans the image affects the choice of the model topology that best describes the data. In the following sections, two different sampling techniques are analysed. In the first case, a rectangular sampling window scans the image left-right, top-bottom. An ergodic model is used to represent the sequences. Simple experimental results are presented to illustrate this approach. In the second case, the sampling window extracts blocks of lines traversing the face from top to bottom. By taking advantage of the fact that features will occur in a predictable order, a non-ergodic (top-bottom) HMM is built to represent the face image. The ergodic model makes no use of structural information. The top-bottom approach, on the other hand, exploits some of the inherent data patterns.

4.3 Ergodic HMMs

In ergodic models every state can be reached from every other state. This implies that all the coefficients of the transition matrix $\mathbf{A}$ are positive. Ergodic models are generally used when limited constraints can be applied to the signal and are the most general type of HMM. To illustrate how they work, an ergodic HMM is built for image data sampled using the technique shown in figure 4.3, where a PxL sampling window scans the image left to right, top to bottom. As the sampling window moves from left to right on a line, each

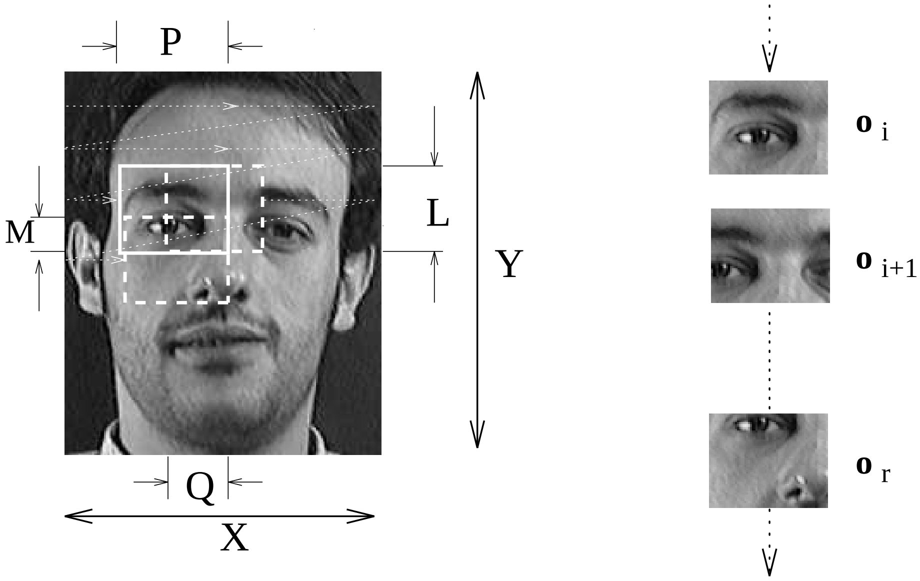


Figure 4.3: Sampling technique for an ergodic HMM

observation has Q columns of overlap with the observation preceding it. When the right edge or the last full frame on the current line is reached, the sampling window moves back to the beginning of the line and shifts down with M rows of overlap between successive lines. Each observation $\mathbf{o}_i$ contains the intensity level values of the pixels sampled by the window, arranged in a column-vector. The parameters chosen for the experiments reported in this section were selected based on subjective intuition. It was decided that an 8-state HMM would be appropriate to model the face, assuming that the forehead, the eyes, the nose, the mouth, the chin, the cheeks and a couple of boundary regions would occupy one state each. The parameters of the sampling method of figure 4.3 were chosen as: $P = 20, L = 16, Q = 5, M = 4$, i.e. a 20x16 rectangular window of small enough size to capture any significant face features. An overlap of 25% was allowed in each direction of sliding. During training the HMM computes the means and standard deviations for each of the 8 state distributions. The means of the distributions represent the face features. These features are the values of $\boldsymbol{\mu}_j$ obtained by the Baum-Welch method for locally maximising the probability of observing the training data, given the model. Figure 4.4 shows the data used to train one such ergodic model and the means of the 8 state distributions of the HMM after training. Displaying the means of the state distributions may help in trying to gain an insight into how the model is segmenting the images and what features are learned. However, the right hand image in figure 4.4 does not display any clearly identifiable features. This is partly due to the fact that by using an ergodic model, no constraints are imposed on the data hence making no use of structural information. In the next section, top-bottom models are introduced and it is shown how,

Figure 4.4: Training data and magnified model means for the ergodic HMM

by making use of structural information, the state distributions can represent features recognisable by humans.

4.4 Top-Bottom HMMs

The states of a HMM can be arbitrarily connected allowing it to represent ergodic signals as detailed in the previous section. However, for pattern recognition applications, it is usually better to imply some constraints on the allowed state transitions in order to reflect known properties of the data. In particular, so-called *left-right* HMM topologies are often employed which have the property that the state index must monotonically increase when progressing through the observation sequence. For faces, the natural order is to traverse the face from top to bottom and hence, *top-bottom* is a more natural designation than *left-right*. Figure 4.5 shows a sampling technique with the window traversing the face from top to bottom. An observation sequence $\mathbf{O}$ is generated from a XxY image using a XxL sampling window with XxM pixels overlap as illustrated in the figure.

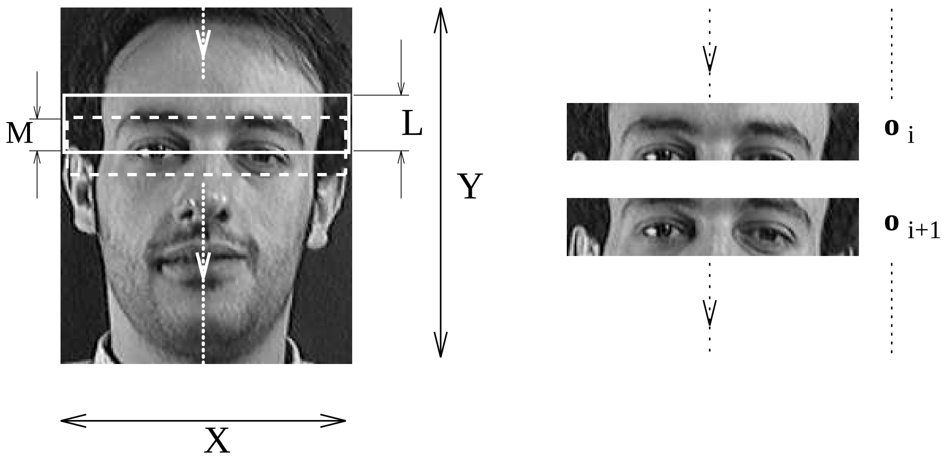


Figure 4.5: Sampling technique for a top-bottom HMM

A 1D vector series of pixel observations is generated, where each observation $\mathbf{o}_i$ contains the values of the pixels in the block of lines arranged in a column-vector. Each observation vector is therefore a block of L lines, and there is an M-line overlap between successive observations. The length of the observation sequence T can be obtained by:

$$T = \phi\left(\frac{Y - L}{L - M}\right) + 1 \tag{4.1}$$

where $\phi(x)$ is the largest integer r such that $r \leq x$ (i.e. ϕ is the round-down function). Assuming that each face is in an upright, frontal position, features will occur in a predictable order, i.e. forehead, then eyes, then nose, and so on. This ordering suggests the use of a top-bottom (non-ergodic) model, where only transitions between adjacent states in a top-bottom manner will be allowed. For face images of fixed size, there are three HMM parameters which affect the performance of the model: the number of HMM states N, the height of the sampling window L and the amount of overlap M. Using shorthand notation, a model with such parameters will be defined as:

$$\mathcal{H} = (N, L, M)$$

Figure 4.6 shows the model for a 5-state HMM, with the expected facial regions as shown. The number of states was chosen to be five based on subjective intuition: by looking at a face image, approximately five horizontal face regions can be identified, namely the forehead, the eyes, the nose, the mouth and the chin. For the experiments presented in this section it is therefore assumed that $N = 5$. The effect of varying N and the other HMM parameters is analysed in detail in the sections that follow. Five images of each subject

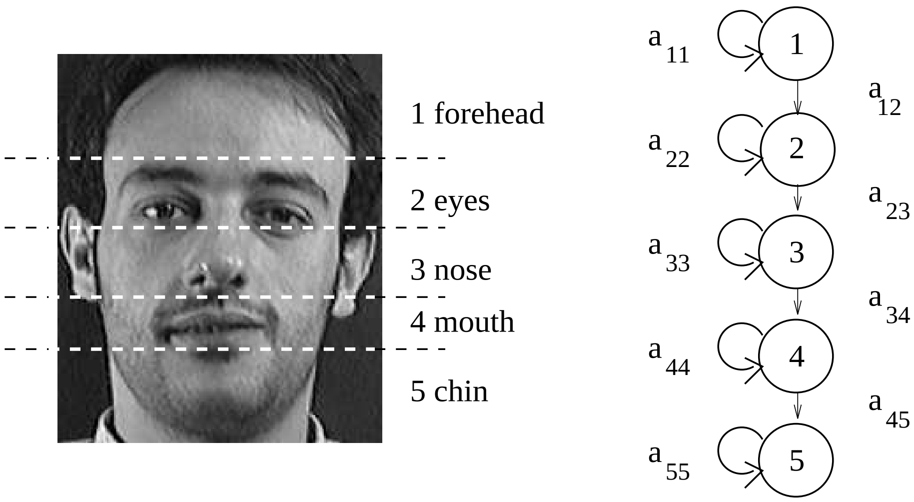


Figure 4.6: Top-bottom 5-state HMM

are used to train a top-bottom HMM with parameters $\mathcal{H} = (5, 8, 7)$. Each training image is sampled by a block of 8 lines moving down in steps of 1 line (i.e. with a 7-line overlap). The overlapping allows the features to be captured in a manner which is independent of vertical position, where a disjoint partitioning of the image could result in the truncation of features occurring across block boundaries. The effect of the overlap is discussed in more detail in the next sections. The parameters for these experiments were chosen based on subjective intuition. The feature extraction and training data segmentation obtained

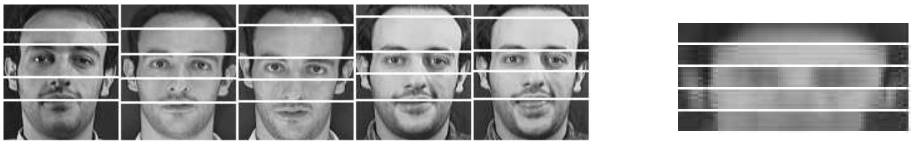


Figure 4.7: Segmented training data and magnified state means for top-bottom HMM

using this model can be observed in figure 4.7. The features correspond approximately to the facial regions which were intuitively predicted in figure 4.6. Due to the overlap, each state leaks into the state following as the first few observations of each state contain pixels already seen in the previous state. It is possible to subjectively associate the features with those understood by humans. For example, the third band appears to contain the eyes, which represent one of the salient features used for identification by humans as reported by Shepherd *et al.* [62].

4.5 Analysis of Top-Bottom Model Parameterisation

This section presents experiments carried out with different models $\mathcal{H}$ using the ORL database of faces. The aim of the experiments was to investigate the effect of different parameterisations on the recognition rates. A HMM was trained for each of the 40 subjects using five training images. The remaining five images for each subject were used for testing, giving a total of 200 test images.

The parameterisation of the model can determine how successful the model is. In the preliminary experiments presented so far, only subjective attempts were made to justify the choice of specific values of $\mathcal{H}$. In the sections that follow experimental results are presented for different values of N, L and M.

For each model $\mathcal{H} = (N, L, M)$, the results of the 200 identification tests are reported as an error rate. Each error rate is calculated as the percentage of the images which are misclassified, where a lower error rate obviously indicates a better model. Trying all possible combinations of N,L and M would require a large number of experiments. It was therefore assumed that, to a certain extent, parameters could be varied independently and only a subset of all possible $\mathcal{H}$ was tested. It is evident that the parameters are not independent. The size of the window L directly constrains the possible values of the overlap M ($0 \leq M \leq L-1$). Moreover, both L and M determine the length of the observation sequence T as can be seen from equation 4.1 and this affects the choice of number of states N. Given the image size, it was decided to experiment with parameters in the following range:

$$\begin{aligned} 2 \leq & \quad N & \leq 112 \\ 1 \leq & \quad L & \leq 10 \\ 0 \leq & \quad M & \leq L-1 \end{aligned}$$

Initially it is assumed that $N = 5$ is a reasonable number of states based on the intuitive argument that five subjective face features appear when traversing the face top-bottom as illustrated in figure 4.6.

4.5.1 Varying the overlap

A model with no overlap implies that training and test faces are partitioned into rigid, arbitrary regions with the risk of cutting across potentially discriminating features. In a top-bottom model with no overlap, features require accurate alignment for successful results. Alignment in images of the same subject is preserved either if the features occupy the exact same position in all the images or if the features are vertically displaced by a number of pixels which is a multiple of L. Unless the images are preprocessed, the features will normally not be in the same position. Therefore in most cases alignment is preserved only if the vertical displacement is a multiple of L. Overlap during the sampling process has the following main functions:

1. The overlap determines how likely feature alignment is and it is expected that a large overlap would increase the likelihood of preserving the alignment.

2. Given a fixed image size and window height, the overlap determines the length of the observation sequence T as can be seen from equation 4.1.

A larger value of M produces a larger T because the face regions are oversampled hence increasing the length of the training and test data observations. The accuracy of the model estimates depends on the number of observations T in the training data. If T is small, the accuracy will be limited, as there are not enough occurrences of model events.

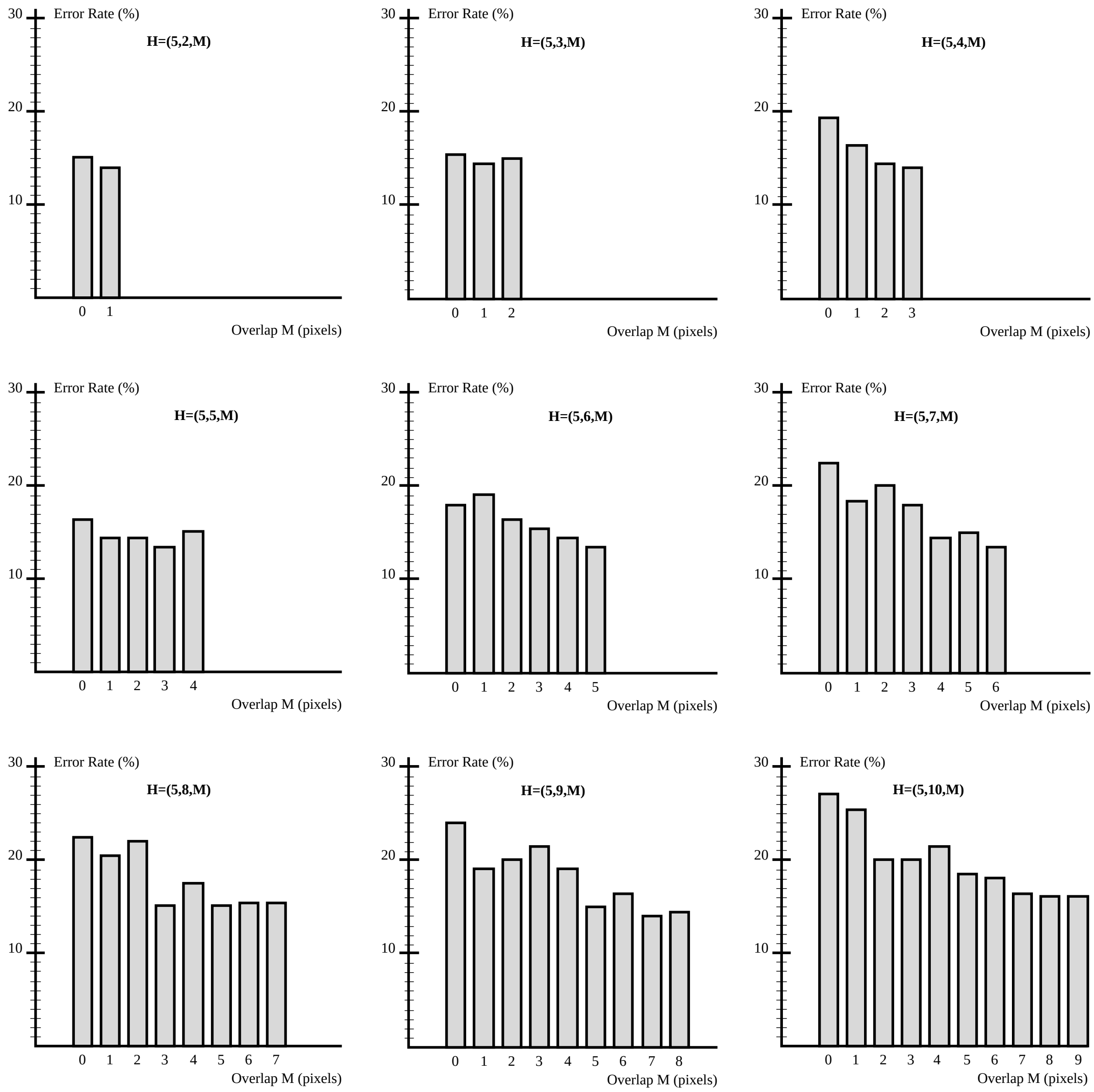


Figure 4.8: Results obtained for varying M

Following the above analysis, it is reasonable to expect better recognition results when a larger value of M is used. In order to determine the effect of M on the recognition performance, a comprehensive set of experiments were run with the number of states fixed to $N = 5$ as discussed in section 4.4, window height in the range $2 \leq L \leq 10$ and every possible overlap $0 \leq M \leq L - 1$. The results are summarised in figure 4.8, where the error rate is expressed as a percentage, and the overlap is in units of pixels. The recognition performance appears to improve as the overlap increases, which is in accordance with expectations. A greater overlap, however, implies a larger value of T and the number of calculations required in the identification process varies linearly with T.

4.5.2 Varying the window height

The window height L has the following functions:

1. It determines the size of the features that the model extracts.

2. For a fixed image size and overlap, L determines the length of the vector series as can be seen from equation 4.1.

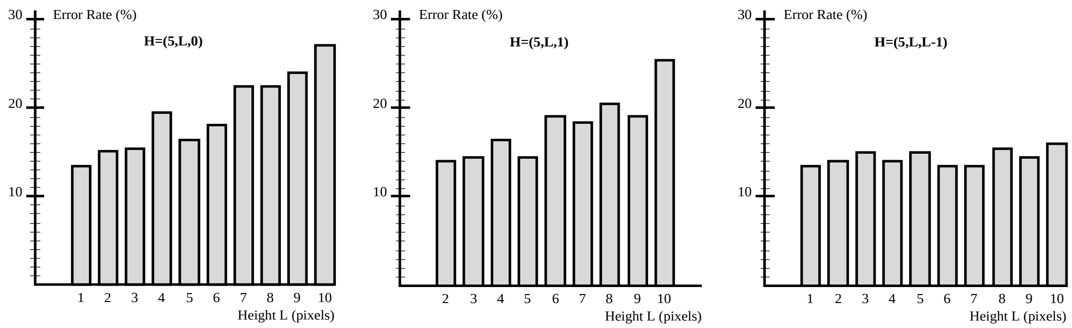


Figure 4.9: Results obtained for varying L

The experiments reported in this section consider the cases with no overlap, one line overlap, and maximum overlap $L-1$. The number of states was still kept to $N=5$. If the sampling window height is sufficiently smaller than the image height Y and there is substantial overlap, then the length of the observation sequence will be large. In this case the value of L is expected to have a limited effect on the identification performance since the overlap guarantees that features are aligned. The histograms of figure 4.9, with the window height L expressed in units of pixels, show the results obtained for the experiments mentioned above. From the results it appears that, for sufficiently large overlap, the window height has a marginal effect on the recognition performance. The effect of the window height becomes more noticeable when there is little or no overlap. In both cases, as the window size increases the error rate also increases. These results are in accordance with expectations. For small overlap, a larger window height implies that there is a smaller chance of features being aligned. It also implies that the model sees less training data, since a fixed small or no overlap implies a smaller T and the value of T also decreases as L increases for a fixed M. If T is too small, it becomes difficult to model the data as not enough training is provided to the HMM.

4.5.3 Varying the number of states

The number of states N in a top-bottom HMM determines the number of features used to characterise the face. If the number of observations in a sequence is very large, a large

N can be chosen to capture more features. However, the computational complexity of the identification algorithm is order N^2 and therefore the smaller the value of N the faster the identification.

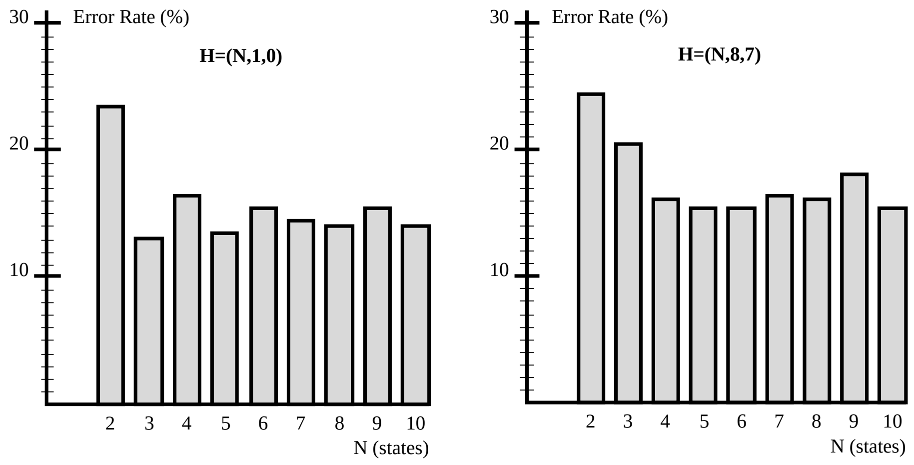


Figure 4.10: Results obtained for varying N

In the experiments presented so far, it has been assumed that five was a reasonable number of states to use. The experiments presented in this section analyse the variation of recognition performance as the number of states varies. Two cases are investigated: the smallest possible window with $L = 1$ and a medium size window $L = 8$ with maximum overlap $M = 7$. The results are presented in figure 4.10. The performance is fairly uniform for the values $4 \leq N \leq 10$, while the error rate increases for values of N smaller than 4, with the exception of the model $\mathcal{H} = (3, 1, 0)$ which recorded an overall error rate of 13%.

4.5.4 Summary of results

The experimental results presented in the previous sections offer an insight into the way top-bottom parameters affect the recognition performance of a HMM. By varying the parameters independently, it was found that some models performed better than others and the results indicated that:

- Large overlap in the sampling resulted in better recognition performances.
- As the overlap became significant, the effect of the window height decreased.
- With the exception of one case, best results were obtained when using at least 3 states.

The various models have so far been assessed on the basis of their recognition performance. In the next sections, some of these experiments are revisited from the point of view of data segmentation and storage requirements.

4.6 Training Data Segmentation

Each subject in the database is stored as a HMM with a mean and standard deviation vector for each state and a transition probability matrix. The internal parameters used by the HMM to characterise a face can be visualised, in order to gain a better understanding of the way a model works. It is possible, for example, to visualise the values of each of the state distribution means by displaying them as bit-maps. Using the Viterbi algorithm, it is also possible to obtain the single best state sequence through a model for the training data. For a top-bottom model the index of each state in the sequence will either remain unchanged or increase by 1. By applying the Viterbi algorithm to the data used to train the model, it is possible to visualise how the data is segmented into states. This sometimes offers a measure of how well the model represents the data and can show how accurate the model estimates are.

Some experimental results with different overlap, window height and number of states are presented in this section and analysed in the context of image segmentation. In all the pictorial figures that follow, the state distribution means are displayed as magnified bit-maps for easier visualisation and they are not in scale with the images of the segmented training data.

4.6.1 Overlap experiments

First, the effect on the segmentation of varying the overlap was investigated. Three experiments with different overlap were carried out using models from the experiments with $\mathcal{H} = (5, 8, M)$. The training data segmentation and the state distribution means were generated for the three cases with maximum, medium and zero overlap ($M = 7, 4, 0$). These three cases were assumed to be representative of the full 0-7 M-range. The results are shown in figure 4.11.

A visual inspection of the results obtained with the three overlaps does not reveal any significant difference. The training images were segmented in a consistent way by all three models, even though the choice of state features varied. The eye band is clearly

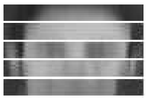
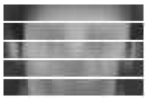
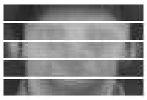
Model	Training Data Segmentation	State Means	Error Rate
H=(5,8,7)			15.5%
H=(5,8,4)			17.5%
H=(5,8,0)			22.5%

Figure 4.11: Segmentation and feature results for varying M

picked up only by the model with the largest overlap. For the model with no overlap, the segmentation depended mostly on the initial vertical location of the feature, as this model has to rely on accurate feature alignment in the vertical direction. For this model, the total number of observations in the sequence is smaller than in the case of the other two models, and therefore each state models, on average, a smaller number of events.

4.6.2 Window height experiments

In order to investigate the effect of the window height L on the training data segmentation, experiments were run with models of the form $\mathcal{H} = (5, L, L-1)$. Two cases were analysed; experiments were carried out with $L = 1$ and $L = 10$, and the results are shown in figure 4.12.

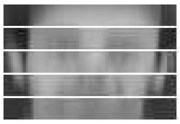
Model	Training Data Segmentation	State Means	Error Rate
H=(5,1,0)			13.5%
H=(5,10,9)			16%

Figure 4.12: Segmentation and feature results for varying L

The segmentation results for the case with $L = 1$ are only consistent for the top three bands. The boundary between the fourth and the fifth state moves irregularly and this is partly due to the small size of the sampling window. For the case with $L = 10$, the image segmentation is consistent across the training images and the five states are similar to those found by the models shown in figure 4.11.

4.6.3 State experiments

The effect of N was investigated through experiments with models of the form $\mathcal{H} = (N, 8, 7)$. Four different cases were analysed, with the value of N set to 2,4,8,16. Figure 4.13 shows the results obtained for these cases.

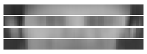
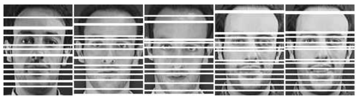
Model	Training Data Segmentation	State Means	Error Rate
H=(2,8,7)			24.5%
H=(4,8,7)			16%
H=(8,8,7)			16%
H=(16,8,7)			18%

Figure 4.13: Segmentation and feature results for varying N

For the case with $N = 2$, both the segmentation and the state images convey little information, as the number of states is too small compared with the number of observations and the number of features that are intuitively visible in the images. Conversely, for case with $N = 16$, it appears that most state bands contain only one or two observations. For the given observation length, this model has too many states. The error rate was higher for the models with $N = 2$ and $N = 16$, compared with the error rate of the models with $N = 4$ and $N = 8$, where the number of states seems more appropriate for the given observation sequence length.

4.7 Storage Requirements

Two quantities are defined in this section. The first quantity is an approximation to the memory requirements to store a trained model representing a certain subject. The other quantity describes the number of parameters used to represent a face image, given a model. Both parameters contribute to the order of calculations needed by the Viterbi recogniser, and therefore can be used to get an impression of how fast images are recognised. These quantities are used in chapter 7, where different spatial resolution experiments are assessed.

4.7.1 Model size

The approximate number of internal parameters used to represent a subject are estimated first. Each subject in the database is stored as a HMM, where the number of parameters required to specify the model depends on the image width X, the number of states N and the window height L. Each model stores the following information:

1. The mean of each state distribution, represented as a XL-row vector.

2. The covariance matrix of each state distribution. Diagonal covariance matrices are used throughout this work, therefore only the XL-row vector representing the standard deviation of each state distribution is required.

3. The transition probability matrix. For a top-bottom model it is sufficient to store for each state s only two parameters, namely the probability of remaining in the same state a_{ss} and the probability of moving to the next one $a_{s\,s+1}$. A total of $2N$ parameters is therefore required.

Using the above assumptions, the quantity p_M^t, defined as the total number of parameters required by a model to represent an individual subject, can be calculated as:

$$p_M^t = 2(XL + N) \tag{4.2}$$

4.7.2 Image size

The number of parameters required to represent a face image depends on the image width X, the window height L and the overlap M. Each image is stored as a sequence of T

observations of size XL. For a top-bottom model, the quantity p_I^t, defined as the total number of parameters required to represent a face image, can be calculated as:

$$p_I^t = XLT \tag{4.3}$$

The value of T can be expressed as a function of L and M using equation 4.1 to yield:

$$p_I^t = XL\left[\phi\left(\frac{Y-L}{L-M}\right)+1\right] \tag{4.4}$$

where ϕ is the round-down function as previously defined.

Chapter 5

Pseudo Two-Dimensional HMMs

The top-bottom technique detailed in chapter 4 gave successful recognition performances of around 85% using the ORL database of images. However, one limitation of the technique was that each image was sampled using blocks of lines. While this allowed for vertical shifting, accurate horizontal alignment was required. Horizontal alignment can only be guaranteed by constraining the training and test data. A more flexible model that allows for shifts in both the horizontal and vertical directions can be obtained using *pseudo 2-dimensional* (P2D) HMMs which are discussed and experimented with in this chapter.

5.1 Introducing P2D-HMMs

Feature alignment in images is useful for applications in the area of image recognition. Levin and Pieraccini [48] formulated the image alignment problem as a dynamic plane warping (DPW) problem, in a way analogous to the dynamic time warping method used in automatic speech recognition. The goal was to align a reference image to a distorted test image. The solution was found to be exponential in the dimensions of the image, hence making it impractical for real images. However, the computational complexity could be reduced to polynomial time by simplifying the original DPW problem. The admissible warping sequences could be limited by assuming that vertical distortion was independent of horizontal position. A statistical interpretation of the DPW approach was realised through planar HMMs, also used by Agazzi *et al.* [1] for degraded text recognition and renamed as P2D-HMMs by Kuo and Agazzi [46] in their keyword spotting work. Images were scanned left-right, top-bottom (using a technique like the one discussed for ergodic 1D HMMs) and the scanned samples were associated with the states of a P2D-HMM arranged in a 2D lattice.

5.1.1 Model definition

P2D-HMM structures are obtained by linking 1D left-right HMMs to form vertical superstates as shown in figure 5.1. The network is not fully connected in two dimensions,

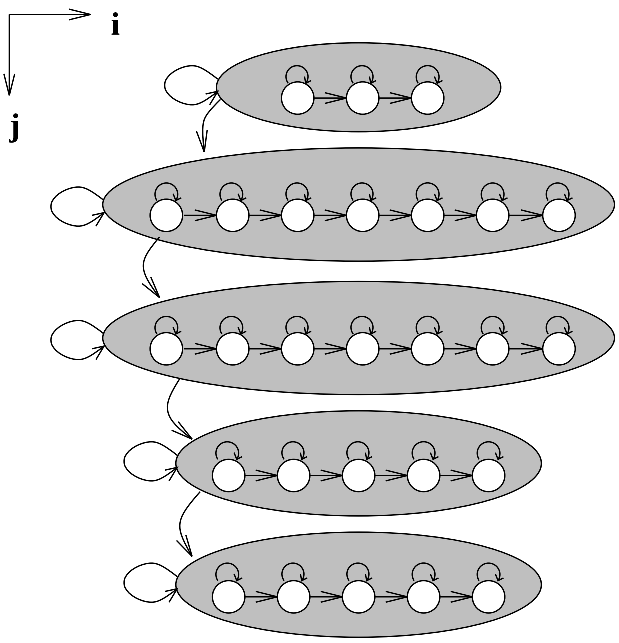


Figure 5.1: Structure of a P2D-HMM

hence it is only *pseudo* 2D. In the horizontal direction, transitions are only allowed among the states of a superstate. In the vertical direction, transitions occur among different superstates. A formal definition of the parameters of a P2D-HMM follows:

1. $\mathbf{N}$ is the number of superstates in the vertical direction.

2. $\mathbf{A} = \{\mathbf{a}_{kj} : 1 \leq k, j \leq \mathbf{N}\}$ is the superstate transition probability matrix.

3. $\mathbf{\Pi} = \{\pi_j : 1 \leq j \leq \mathbf{N}\}$ is the initial superstate probability distribution.

4. $\mathbf{\Lambda} = \{\lambda^j : 1 \leq j \leq \mathbf{N}\}$ is the set of left-right 1D HMMs in each superstate. Each λ^j is specified by the standard 1D HMM parameters:

   - N^j is the number of states.
   - $A^j = \{a^j_{ki} : 1 \leq k, i \leq N^j\}$ is the state transition matrix.
   - $B^j = \{b^j_i(\cdot) : 1 \leq i \leq N^j\}$ is the output probability function.
   - $\Pi^j = \{\pi^j_i : 1 \leq i \leq N^j\}$ is the initial state probability distribution.

A P2D-HMM can be specified using the shorthand notation: $\eta = (\mathbf{N}, \mathbf{A}, \mathbf{\Pi}, \mathbf{\Lambda})$. The model parameters $\mathbf{A}$, $\mathbf{\Pi}$ and $\mathbf{\Lambda}$ can be estimated during the training procedure using the segmental k-means algorithm described by Rabiner [56]. The recognition is achieved by running a double execution of the Viterbi algorithm as described by Kuo and Agazzi [46] and summarised in the next section.

5.1.2 The matching algorithm

Recognition is accomplished by matching a test observation sequence against each stored P2D-HMM in a way similar to the 1D HMM. The sampling technique used for ergodic 1D HMMs illustrated in figure 4.3 can be used to generate a sequence of T observations. The observations can be arranged in a ExF lattice, where, using the parameters shown, E and F can be calculated as follows:

$$E = \phi\left(\frac{X-P}{P-Q}\right)+1 \quad (5.1)$$

$$F = \phi\left(\frac{Y-L}{L-M}\right)+1 \quad (5.2)$$

An observation vector $\mathbf{O}$ therefore consists of a sequence of rectangularly arranged vectors $\mathbf{o}_{ef} : 1 \leq e \leq E,\ 1 \leq f \leq F$ as follows:

$$\mathbf{O} = \mathbf{o}_{11}, \mathbf{o}_{21}, \ldots, \mathbf{o}_{E1}, \mathbf{o}_{21}, \ldots, \mathbf{o}_{EF} \quad (5.3)$$

The state sequence Q associated with the observation sequence $\mathbf{O}$ for the kind of topology shown in figure 5.1 can then be expressed as:

$$Q = \{\mathbf{q}_f, q_{1f}, q_{2f}, \ldots, q_{Ef} : 1 \leq f \leq F\} \quad (5.4)$$

where $\mathbf{q}_f$ is the superstate occupied by the f-th row of observations and q_{ef} is the state occupied by the observation $\mathbf{o}_{ef}$ in the corresponding $\lambda^{\mathbf{q}_f}$. For a model η the goal is to find the best single state sequence that maximises $P(Q \mid \mathbf{O}, \eta)$ or equivalently $P(Q, \mathbf{O} \mid \eta)$.

Defining $\mathbf{O}_f = \mathbf{o}_{1f}, \ldots, \mathbf{o}_{Ef}$ as the sequence observed in the f-th row, the highest probability of ending in superstate j and accounting for the observations in the first f rows of observations is defined as $D_f(j)$:

$$D_f(j) = \max_{\mathbf{q}_1,\ldots,\mathbf{q}_{f-1}} \left\{ P(\mathbf{q}_1 \ldots \mathbf{q}_f = j, \mathbf{O}_1 \ldots \mathbf{O}_f \mid \eta) \right\} \quad (5.5)$$

The probability of row f in superstate j defined as $P_j(f) = P(\mathbf{O}_f \mid \mathbf{q}_f = j)$ is needed to maximise 5.5. Calculating $P_j(f)$ is equivalent to calculating $P(\mathbf{O}_f \mid \lambda^j)$ and can be obtained by running the Viterbi algorithm on the observation sequence in row f and the

1D left-right HMM λ^j. The quantity $\delta^j_{ef}(i)$ is defined as the highest probability of ending in state i and accounting for the first e observations in row f as follows:

$$\delta^j_{ef}(i) = \max_{q_{1f},\ldots,q_{e-1,f}} \left\{ P(q_{1f},\ldots,q_{ef}=i,\mathbf{o}_{1f},\ldots,\mathbf{o}_{ef} \mid \lambda^j) \right\} \tag{5.6}$$

The standard inductive procedure of the Viterbi algorithm as used for 1D HMMs can be used to compute $P_j(f)$ as follows:

1. Initialisation

$$\begin{aligned} \delta^j_{1f}(i) &= P(q_{1f}=i,\mathbf{o}_{1f} \mid \lambda^j) \\ &= P(q_{1f}=i \mid \lambda^j)P(\mathbf{o}_{1f} \mid q_{1f}=i,\lambda^j) \\ &= \pi^j_i b^j_i(\mathbf{o}_{1f}), \\ & \qquad\qquad\qquad 1 \le i \le N^j \end{aligned} \tag{5.7}$$

2. Induction

$$\begin{aligned} \delta^j_{e+1,f}(i) = \max_{i-1\le k\le i} \left[\delta^j_{ef}(k)a^j_{ki}\right] b^j_i(\mathbf{o}_{e+1,f}), \\ 1 \le e \le E-1, \\ 1 \le i \le N^j \end{aligned} \tag{5.8}$$

3. Termination

$$P_j(f) = \max_{1\le i\le N^j} \left[\delta^j_{Ef}(i)\right] \tag{5.9}$$

In the induction step outlined above it is assumed that the states of the 1D HMM occur in a strictly left-right order and hence the maximisation is only computed for $i-1 \le k \le i$. An array $\Psi^j_{ef}(i)$ can be set up to track the optimum states that maximise 5.6. The quantity $D_f(j)$ can then be calculated inductively as follows:

1. Initialisation

$$\begin{aligned} D_1(j) &= P(\mathbf{q}_1=j,\mathbf{O}_1 \mid \eta) \\ &= P(\mathbf{q}_1=j \mid \eta)P(\mathbf{O}_1 \mid \mathbf{q}_1=j,\eta) \\ &= \pi_j P_j(1), \\ & \qquad\qquad\qquad 1 \le j \le \mathbf{N} \end{aligned} \tag{5.10}$$

2. Induction

$$D_{f+1}(j) = \max_{j-1\le k\le j} \left[D_f(k)\mathbf{a}_{kj}\right] P_j(f+1), \tag{5.11}$$

$$1 \le f \le F-1,$$

$$1 \le j \le \mathbf{N} \tag{5.12}$$

3. Termination

$$P^* = \max_{1 \le j \le \mathbf{N}} \left[D_F(j) \right] \tag{5.13}$$

In the induction step outlined above it is assumed that superstates occur in a strictly top-bottom order and hence the maximisation is only computed for $j-1 \le k \le j$. An array $\gamma_f(j)$ is used to store the superstates that maximise equation 5.5. An array $\chi_j(f)$ is defined to store the last state of the optimum path traced by the f-th row of observations $\mathbf{O}_f$ in superstate j as follows:

$$\chi_j(f) = \arg \max_{1 \le i \le N^j} \left[\delta^j_{Ef}(i) \right] \tag{5.14}$$

Finally, by backtracking through $\gamma_f(j)$ and $\Psi^j_{ef}(i)$ it is possible to find the maximum likelihood state sequence $\mathbf{Q}$ as follows:

$$\mathbf{q}_F = \arg \max_{1 \le j \le \mathbf{N}} \left[D_F(j) \right]$$

For $f = F-1$ to 1

$$\mathbf{q}_f = \gamma_{f+1}(\mathbf{q}_{f+1})$$

$$q_{Ef} = \chi_{\mathbf{q}_f}(f)$$

For $e = E-1$ to 1

$$q_{ef} = \Psi^{\mathbf{q}_f}_{e+1,f}(q_{e+1,f})$$

The value of P^* obtained from 5.13 is a measure of how well the P2D-HMM η models the data $\mathbf{O}$. A different model η is generated for each subject in the database. For a test image generating the observation $\mathbf{O}_{test}$, the values of P^* corresponding to the different models are computed and compared. The test image is identified as the subject represented by the highest scoring model.

5.2 Implementation of P2D-HMMs

5.2.1 Equivalent 1D topology

In this section it is shown how a P2D-HMM can be transformed into an equivalent standard 1D HMM. The equivalent model is made of rows of states, one for each superstate of the P2D-HMM. A P2D-HMM and its equivalent are shown in figure 5.2. The shaded

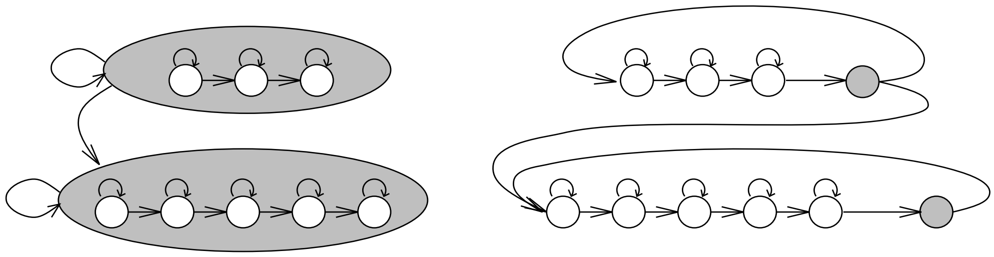


Figure 5.2: P2D-HMM and its equivalent 1D HMM

states in the 1D model are end-of-line states which consume one observation only. The end-of-line states are allowed two transitions: one to the same row of states, which is equivalent to a superstate self-transition; one to the next row of states, which is equivalent to a superstate-to-superstate transition. The sampling technique of figure 4.3 is slightly modified by adding a white frame at the end of each line of sampling. The white frame was chosen as it is considered to be a frame that would be unlikely to occur normally in the sequence. The white frame is an end-of-line marker and is modelled by the end-of-line states. When an end of-line-state is reached, the model can either stay in the same row of states or jump to the next row of states. When the end-of-line state of the last row of states is reached, the model can either repeat the last row of states or terminate when the last observation is reached. The illustrated topology with end-of-line states has the effect of modelling the observation sequence as a two-dimensionally arranged grid of data. Each line of observations is modelled by a row of states. In order to maximise the probability of the end-of-line state modelling the added white frame, the mean of each end-of-line state is set to white and the variance is set to be very small, as detailed in the next section. This makes it very unlikely for an end-of-line state to model any observation other than the white frame. However, it is possible that a row of states could model more than one line of observations, as there is no way to enforce that the white frame should not be generated by a state other than the end-of-line state. In order to reduce the probability of the white frame being generated by a state other than an end-of-line state, the end-of-line marker was chosen to be the white frame. This frame is unlikely to generate a high probability in any other state, as it is an unlikely frame to occur naturally. In the experiments presented in this chapter, the assumption that the suggested topology would model the images as a P2D-HMM was found to be satisfied.

5.2.2 Training procedure for P2D-HMMs

The training procedure detailed in section 4.1.2 needed to be modified for the P2D-HMM experiments. The initial uniform segmentation of the observation sequence was an appropriate estimate for top-bottom models. Uniform segmentation, however, is inadequate for P2D-HMMs, as it fails to preserve the 2D structure of the data. A simple alternative was used to generate a model estimate that served the purpose of specifying the state topology. The Gaussian state distribution parameters were set to neutral values for all the states except the end-of-line states. The neutral parameters were chosen as mid-intensity values for the mean and large standard deviations. For the end-of-line states, the means were set to the end-of-line marker frame, with very small standard deviations. These parameters were re-estimated using the standard Baum-Welch procedure. It was found that by setting the standard deviation of the end-of-line states to be small, the state topology was preserved and the parameters of the end-of-line states were unaltered after re-estimation. The training procedure is illustrated in figure 5.3 and consists of the following steps:

1. J training images are collected for the kth subject in the database and are sampled generating J distinct observation sequences, with an added white frame at the end of each line of observations.

2. A simple common prototype model λ_0 specifying which states are end-of-line states is constructed. The prototype model has the further purpose of specifying the number of states in the HMM, the state transitions allowed and the size of the observation sequence vectors. The mean values of the end-of-line states are set to white (intensity level 255) and the standard deviations are set to a small value (the value used was 10^{-4}). The mean values of all the other states were set to the mid-intensity level value (128 for 8-bit images) with standard deviations of 2x10^2.

3. The HMM parameters of the simple prototype model are re-estimated using the Baum-Welch method. The model parameters are adjusted so as to locally maximise the probability of observing the training data, given each corresponding model. The outcome of this process is the HMM $\lambda^{(k)}$ which is used to represent subject k in the database. It was found that the mean and the standard deviation vectors of the end-of-line states were unaltered after re-estimation, thus preserving the model topology.

5.3 Experimental Results

There are five topology and sampling parameters that characterise a P2D-HMM. The topology information is summarised in the total number of states N, which is the sum of

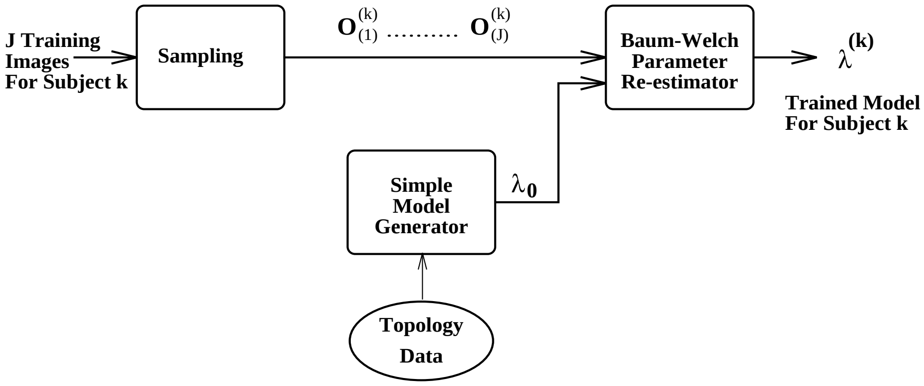


Figure 5.3: Block diagram of P2D-HMM training technique

all the states within the superstates, as follows:

$$N = \sum_{j=1}^{\mathbf{N}} N^j \tag{5.15}$$

The sampling information is represented by the width and height of the sampling window, the horizontal overlap and the vertical overlap. With reference to the parameters of figure 4.3, these are P, L, Q and M respectively. Using short-hand notation, a parameterised P2D-HMM $\mathcal{P}$ is denoted by:

$$\mathcal{P} = (N, P, L, Q, M)$$

The number of states N is expressed as i-j-..., where i is the number of states in superstate 1, j is the number of states in superstate 2, etc., and where the number of states in each superstate includes the end-of-line states.

A set of experiments were carried out for different topology and sampling parameters with the ORL database of faces. Some of the insights obtained from experimenting with top-bottom models were taken into consideration when choosing the P2D-HMM parameters. It was decided that the P2D-HMM should have at least four superstates and the experiments reported here were carried out using four and five superstates. The results are summarised in table 5.1, where the experiments are grouped according to the number of states.

The error rates for the best performing P2D-HMMs were approximately 5%, with many models scoring success rates above 90%. Better results were obtained from the experiments where the overlap was large in both the horizontal and vertical direction, confirming the findings from the top-bottom experiments. The state topology was chosen

N	**Sup.states**	**Topology**	PxL	QxM	**Err. Rate**
16	4	3-5-5-3	10x8	8x6	18%
16	4	3-5-5-3	12x8	8x6	10.5%
16	4	3-5-5-3	24x22	20x13	14%
16	4	3-5-5-3	8x8	6x6	10.5%
18	4	4-5-5-4	2x2	1x1	9%
18	4	4-5-5-4	2x2	0x0	8%
18	4	4-5-5-4	4x4	2x2	8.5%
20	4	4-6-6-4	4x4	0x0	13%
20	5	4-4-4-4-4	7x4	2x1	15%
24	5	3-6-6-6-3	10x8	8x6	5.5%
24	5	3-6-6-6-3	12x8	9x6	5.5%
30	5	4-8-8-6-4	12x8	8x6	6.5%
30	5	4-8-8-6-4	12x8	4x6	14%
30	5	4-8-8-6-4	24x22	20x13	10%
30	5	4-8-8-6-4	2x2	1x1	6.5%
30	5	4-8-8-6-4	2x2	0x0	7%

Table 5.1: Results for P2D-HMMs

based on intuition. The first and last superstate are generally assigned the smallest number of states, as it is believed that they will model regions of less importance for recognition, i.e. the top of the head and the chin. Most of the useful information is assumed to be inside the face and therefore the other superstates are assigned a larger number of states for more accurate modelling.

5.4 Image Segmentation with P2D-HMMs

As for top-bottom models, it is possible to visualise the mean vector of the Gaussian distributions associated with each state. For example, the magnified state means of the P2D-HMM with parameters $\mathcal{P} = (3\text{-}5\text{-}5\text{-}3, 24, 22, 20, 13)$ are displayed in figure 5.4. These are the magnified state means obtained by the P2D-HMM after training on the data shown on the left of figure 4.4. This particular model was chosen because the size of the sampling window (and hence of the mean vector) is sufficiently large to be displayed.

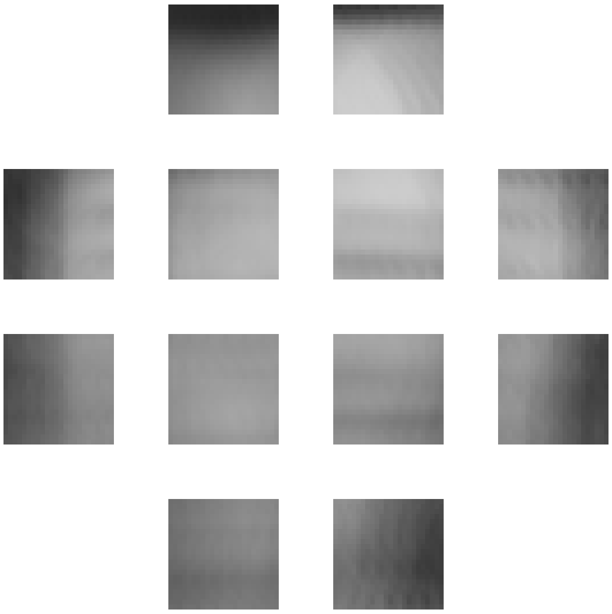


Figure 5.4: P2D-HMM states for $\mathcal{P} = (3\text{-}5\text{-}5\text{-}3, 24, 22, 20, 13)$

Given an image and a P2D-HMM, the optimal state sequence obtained after running the Viterbi algorithm on the image can be used to segment the image into regions. The images are split into a number of horizontal bands equal to the number of superstates. Each horizontal band is further divided vertically into the number of states within that superstate. The results obtained on one of the training images using the same P2D-HMM as above are shown in figure 5.5, where the end of line states are not displayed.

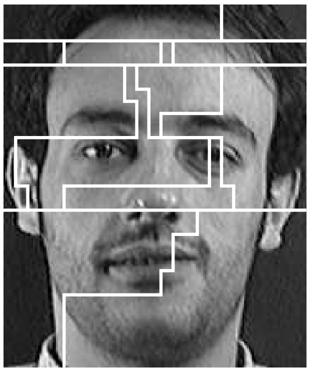


Figure 5.5: P2D-HMM segmentation for $\mathcal{P} = (3\text{-}5\text{-}5\text{-}3, 24, 22, 20, 13)$

Other segmentation experiments were carried out to try to isolate the background from the face image. Three 4-state, 1-superstate P2D-HMM were trained. The four states were used to model the image left edge, the face, the right edge and the end-of-line white frame, as shown in figure 5.6.

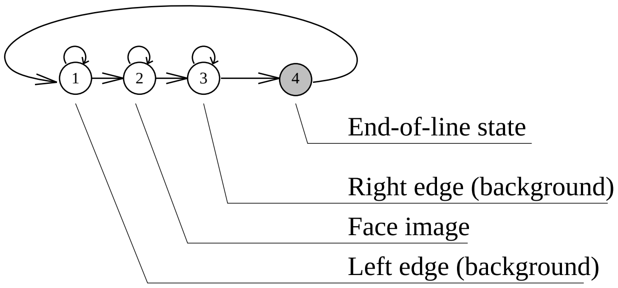


Figure 5.6: The 4-state P2D-HMM

The three 4-state P2D-HMMs were trained on images sampled with a 4x4, 2x2 and 1x1 sampling window and maximum overlap in both directions. The segmentation results are shown in figure 5.7, where three different shades of grey are used to represent each of the P2D-HMM states (hence 3 colours, without counting the end-of-line state). The segmentation results for the 4x4, 2x2 and 1x1 models are shown in the figure from left to right, respectively. The results for the three models are very similar to each other. The left edge is modelled very accurately, with the left ear clearly visible. The right edge, on the other hand, spills into the face at the eyes and mouth height, which are of a dark colour similar to the background. The 4-state P2D-HMMs were also trained and tested with images rotated by 180 degrees (showing the subjects upside-down). The segmentation results obtained were the same as those of figure 5.7.

Figure 5.7: Segmentation results for the 4-state P2D-HMMs

5.5 P2D-HMM Storage Requirements

The number of P2D-HMM parameters needed to store a face model and a face image are defined in this section, similarly to those defined for top-bottom models in section 4.7.

5.5.1 P2D-HMM model size

The number of parameters needed to specify a P2D-HMM is defined as the quantity p_M^p. It depends on the size of the sampling window PxL and the total number of states N. Each HMM stores the following information:

1. The mean of each state distribution, stored as a PxL-row vector.
2. The standard deviation vector of each state distribution, stored as a PxL-row vector.
3. The transition probability matrix, which, for the 1D equivalent of a P2D-HMM illustrated in figure 5.2, consists of two transitions probabilities for each state.

Using these assumptions, the number of parameters p_M^p needed to store a P2D-HMM is:

$$p_M^p = 2(PL + N) \tag{5.16}$$

5.5.2 P2D-HMM image size

The number of parameters needed to represent an image is defined as the quantity p_I^p. It depends on the size of the sampling window PxL and the number of observations in the sequence, defined as T_p, obtained after sampling the image. By applying the same concept used to calculate T in 4.1 for the top-bottom case, extended to two dimensions, T_p for the P2D-HMM is calculated as:

$$T_p = \left[\phi\left(\frac{Y-L}{L-M}\right)+1\right]\left[\phi\left(\frac{X-P}{P-Q}\right)+1\right] \tag{5.17}$$

Using this equation, a value for p_I^p can then be estimated as:

$$p_I^p = PL\left[\phi\left(\frac{Y-L}{L-M}\right)+1\right]\left[\phi\left(\frac{X-P}{P-Q}\right)+1\right] \tag{5.18}$$

5.6 Unconstrained P2D-HMMs

In the last section of this chapter, experiments are presented to investigate the performance of *unconstrained* P2D-HMMs. An unconstrained P2D-HMM is a model with no end-of-line state. During sampling, this model does not require the addition of an end-of-line white frame. As for the P2D-HMM, the states are arranged in a 2-dimensional grid. However, no attempt is made to enforce the fact that the last frame of a line of observations should be generated by the last state of a superstate. It is also possible that a transition to a new superstate could occur from a frame that is in the middle of a line of observations. In practice, the unconstrained P2D-HMM is a standard 1D HMM with left-right transitions and a number of loop-back transitions that simulate the superstate structure. Figure 5.8 shows the topology of a 25-state, 5-superstate unconstrained P2D-HMM.

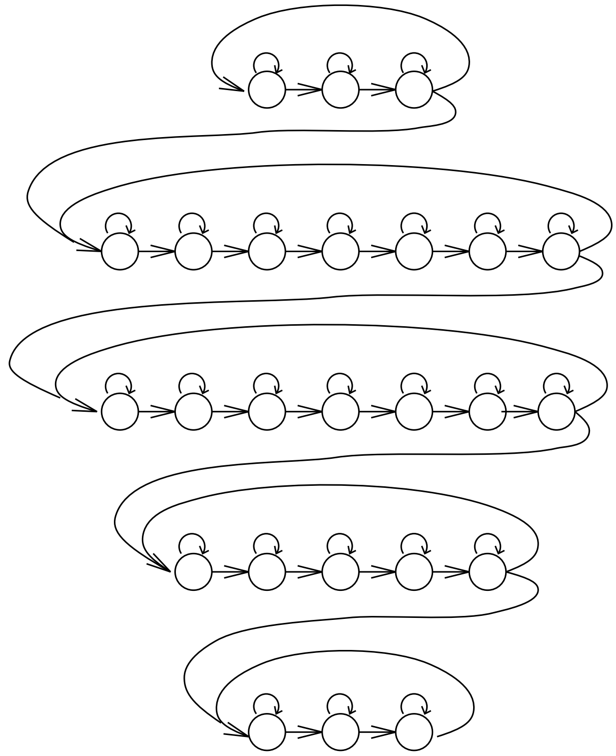


Figure 5.8: Unconstrained 25-state P2D-HMM

Some experiments were carried out using simple topologies. The size of the sampling window was, with one exception, 12x8 pixels. During the training phase, the model parameters were initialised uniformly, by setting the means and the standard deviations to the same values for all the states. The results that were obtained are summarised in table 5.2.

The recognition results for the unconstrained P2D-HMMs are similar to the results obtained with standard P2D-HMMs. The model with error rate 2% scores in absolute terms the best recognition result obtained with the ORL database. However, by considering the performance of the other two 25-state models with slightly different overlap, the 2% error rate appears to be a fortuitous result.

The state segmentation obtained with unconstrained P2D-HMMs was analysed for the various models of table 5.2. In general, the last state of a superstate is associated with

N	**Sup.states**	**Topology**	PxL	QxM	**Err. Rate**
12	4	3-3-3-3	12x8	4x6	18%
12	4	3-3-3-3	15x17	8x12	10%
21	5	3-5-5-5-3	12x8	4x6	12.5%
25	5	3-7-7-5-3	12x8	8x6	6%
25	5	3-7-7-5-3	12x8	4x6	2%
25	5	3-7-7-5-3	12x8	4x4	8.5%

Table 5.2: Results for unconstrained P2D-HMMs

frames either at the end of a line of observations, hence modelling the right edge of the image; or with frames at the beginning of a line of observations, modelling the left edge of the image. This is not surprising, as the observations, sampled around the edges (which, for the most part, contain background information), are very similar. Sometimes the last observation on a line is not modelled by the last state of a superstate and, in a number of cases, the last state of some superstates occurs in the middle of a line of observations. The segmentation that is obtained from unconstrained P2D-HMMs is therefore more difficult to interpret and often appears to be random.

Chapter 6

Comparison With The Eigenface Method

Different face recognition methods were described in chapter 2 and their success rates reported. However, since the experiments were carried out with different data, it was not possible to decide which methods performed better or to make a direct comparison. The constraints applied to the data varied from case to case and often little emphasis was put on presenting a statistical analysis of the results.

In order to gain an insight into how the HMM approach proposed in this dissertation compares with other established techniques, a set of experimental results based on Eigenfaces is presented in this chapter. The Eigenface experiments were carried out using the software implemented by Cham [13]. The Eigenface method is first summarised. A set of recognition results using this method on the ORL database is presented. Since the Eigenface approach is tested on the same data as the HMMs, a comparison between the two methods is drawn. The best top-bottom and P2D-HMM models are selected from the results described in previous chapters and a statistical comparison with the Eigenface method is presented.

6.1 The Eigenface Approach

In many pattern classification problems, data is represented in a n-dimensional space, usually chosen prior to observing any data. For such cases, it is often possible to reduce the dimensionality of the representation space without losing information, hence encoding the data in a more efficiently. One way of achieving this is through Principal Component Analysis (PCA). The principal components of the data distribution can be found by computing the eigenvalues of the covariance matrix of the data. A more detailed description of this method is presented by Thierren [68].

PCA-based methods have been recently applied to face recognition as detailed by Kirby and Sirovich [42] and Turk and Pentland [69]. Each face image is converted into a column-vector by scanning each pixel left-right, top-bottom. Therefore a rectangular image of dimensions X by Y is expressed as a vector in XY dimensions. For a training set of s images, each image is represented as a point in the XY-space. However, as s is in practice much smaller than XY, each image-point will lie on a hyperplane (called feature space) within the full XY-space. By chosing an alternative set of axes with the origin in the feature space, the number of dimensions used to represent the training data can be reduced to $s-1$. The goal of PCA is to determine an appropriate set of axes that represent the s training images in the space with reduced dimensions. The eigenvectors of the data covariance matrix are used as axes and their importance is ranked according to the value of their corresponding eigenvalue. The number of dimensions can be reduced further by considering only the eigenvectors which have larger corresponding eigenvalues. The eigenvectors that best account for the distribution of face images within the entire image space are termed *Eigenfaces* by Turk and Pentland [69].

6.1.1 Calculating the Eigenfaces

Each training image is represented by a column-vector $\mathbf{x}_i$ where $1 \leq i \leq s$. The mean $\mathbf{m}$ of the training set is calculated as:

$$\mathbf{m} = \frac{1}{s}\sum_{i=1}^{s}\mathbf{x}_i \tag{6.1}$$

The mean of equation 6.1 lies in the feature space spanned by the s images and can therefore be chosen as a convenient point to which the origin can be shifted. A vector $\mathbf{y}_i$ is defined as the shifted version of $\mathbf{x}_i$:

$$\mathbf{y}_i = \mathbf{x}_i - \mathbf{m} \tag{6.2}$$

The shifted training set is represented as a XY by s matrix A as follows:

$$A = [\mathbf{y}_1\mathbf{y}_2\ldots\mathbf{y}_s] \tag{6.3}$$

The data covariance matrix is then simply given by:

$$C = AA' \tag{6.4}$$

where A' denotes the transpose of A. The covariance matrix is of dimensions XY by XY, which makes the computation of its eigenvectors unfeasible for typical image sizes. Considering that only $s-1$ eigenvectors will have non-zero corresponding eigenvalues, there is an alternative way to determine those eigenvectors, as illustrated by Turk and Pentland [69]. Denote the eigenvectors of the matrix $A'A$ (which is of size s by s) as $\mathbf{v}_k$ with corresponding eigenvalues λ_k. Then the eigenvector/eigenvalue equation can be written as:

$$A'A\mathbf{v}_k = \lambda_k \mathbf{v}_k \tag{6.5}$$

Premultiplying both sides by A yields:

$$AA' \; A\mathbf{v}_k = \lambda_k \; A\mathbf{v}_k \tag{6.6}$$

Letting $\mathbf{u}_k = A\mathbf{v}_k$ and substituting for C from equation 6.4 yields:

$$C\mathbf{u}_k = \lambda_k \mathbf{u}_k \tag{6.7}$$

which means that the eigenvectors with non-zero eigenvalues of C can be found by pre-multiplying the eigenvectors of $A'A$ by A. The order of calculations is therefore reduced from the resolution of the images (X^2Y^2) to the size of the training set (s^2).

6.1.2 Using Eigenfaces for recognition

Face images are encoded by projecting them onto the axes spanning the feature space. An unknown test image $\mathbf{t}$ is projected to $\mathbf{t}^*$, its image in the feature space, as follows:

$$\mathbf{t}^* = U'(\mathbf{t} - \mathbf{m}) \tag{6.8}$$

where $U = [\mathbf{u}_1\mathbf{u}_2 \ldots \mathbf{u}_E]$ and $1 \leq E \leq s$, depending on how many Eigenfaces are used, is the s by E eigenvector matrix. In order to determine which face best matches the test image, the Euclidean distance ϵ_i between each training image and the projected test image is calculated as:

$$\epsilon_i = ||\mathbf{x}_i^* - \mathbf{t}^*|| \qquad\qquad 1 \leq i \leq s \tag{6.9}$$

where $\mathbf{x}_i^* = U'(\mathbf{x}_i - \mathbf{m})$ is the projection of $\mathbf{x}_i$ onto the feature space.

6.1.3 Experimental results with Eigenfaces

The Eigenface approach described in the previous sections was tested on the ORL database. The Eigenfaces for the 200 training images were calculated and experiments using various number of Eigenfaces were carried out. Since most of the data information is packed in the principal components (the Eigenfaces with largest eigenvalues), it was possible to experiment with a smaller number of Eigenfaces. The five Eigenfaces with largest and smallest eigenvalues are shown in figure 6.1. Reducing the number of Eigenfaces could

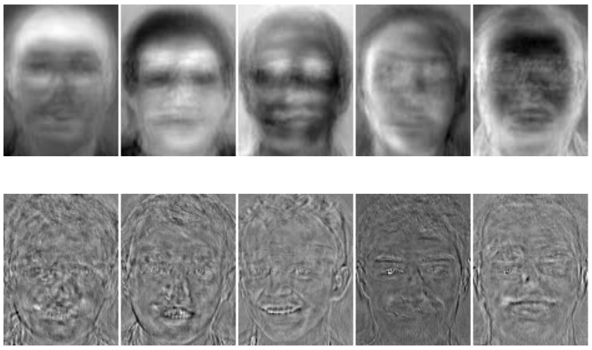


Figure 6.1: Five most and least significant Eigenfaces

have the effect of removing some data noise, while speeding up the recognition process. Experiments where carried out with the number of Eigenfaces varying from 5 to 199 in increments of 5. Each test face was identified as the nearest neighbour training image. The results are shown in figure 6.2. The best results were obtained using between 175 and 199 Eigenfaces, with an error rate of 10%. The performance worsened when using less than 10 eigenvalues and was quite uniform with 10 or more eigenvalues.

6.2 McNemar's Statistical Test

The results obtained with HMMs are compared with the Eigenface method. Both approaches were tested using the same database; however, it was decided that a direct comparison of the results would not be sufficient. A more sophisticated statistical approach, based on McNemar's test as described by Gillick and Cox [27], is used to test the statistical significance of the results. The work presented by Gillick and Cox referred to speech recognition algorithms, but it is equally applicable to face recognition.

Two algorithms, A_1 and A_2, are presented with the same n test images $f = \{f_1, f_2, \ldots, f_n\}$. Assuming that the n test images are a representative sample of a larger population of images, the goal is to establish if the true (but unknown) error rate p_1 of A_1 is larger, equal or

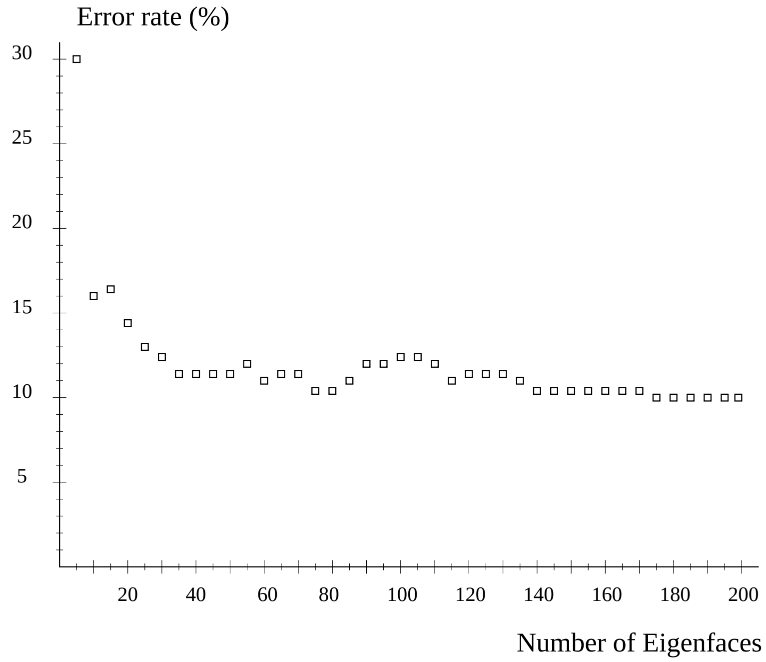


Figure 6.2: Results using the Eigenface approach

smaller than the true (but unknown) error rate p_2 of A_2. The following random variables are defined:

$$\begin{aligned} N_{00} &= \text{Images that } A_1 \text{ and } A_2 \text{ classify correctly} \\ N_{01} &= \text{Images that } A_1 \text{ classifies correctly and } A_2 \text{ misclassifies} \\ N_{10} &= \text{Images that } A_1 \text{ misclassifies and } A_2 \text{ classifies correctly} \\ N_{11} &= \text{Images that } A_1 \text{ and } A_2 \text{ misclassify} \end{aligned}$$

From the definition, it follows that:

$$n = N_{00} + N_{01} + N_{10} + N_{11} \tag{6.10}$$

The joint performance of two algorithms A_1 and A_2 is summarised using a table like the one shown in table 6.1. Probability quantities corresponding to the those defined above are:

$$\begin{aligned} q_{00} &= P(A_1 \text{ and } A_2 \text{ classify } f_i \text{ correctly}) \\ q_{01} &= P(A_1 \text{ classifies } f_i \text{ correctly and } A_2 \text{ misclassifies it}) \\ q_{10} &= P(A_1 \text{ misclassifies } f_i \text{ and } A_2 \text{ classifies it correctly}) \\ q_{11} &= P(A_1 \text{ and } A_2 \text{ misclassify } f_i) \end{aligned}$$

The error rates for A_1 and A_2 can then be calculated as:

$$p_1 = q_{10} + q_{11} \tag{6.11}$$

$$p_2 = q_{01} + q_{11} \tag{6.12}$$

The null hypothesis $\mathbf{H}_0$ is:

$$\mathbf{H}_0 : \qquad p_1 = p_2 \tag{6.13}$$

Substituting for p_1 and p_2 in 6.13, the equivalent null hypothesis $\mathbf{H}_0^1$ is obtained:

$$\mathbf{H}_0^1 : \qquad q_{01} = q_{10} \tag{6.14}$$

Finally, by defining q as the conditional probability that A_1 will make an error, given that only one of the two algorithms makes an error:

$$\begin{aligned} q &= P(A_1 \text{ makes an error} \mid \text{only one makes an error}) \\ &= \frac{P(A_1 \text{ makes an error , only one makes an error})}{P(\text{only one makes an error})} \\ &= \frac{q_{10}}{q_{01} + q_{10}} \end{aligned} \tag{6.15}$$

a further equivalent null hypothesis $\mathbf{H}_0^2$ is obtained:

$$\mathbf{H}_0^2 : \qquad q = \frac{1}{2} \tag{6.16}$$

The fact that the null hypothesis now depends only on q_{01} and q_{10} was expected, as no measure of the relative performance of the two algorithms is obtained when either both fail or both succeed. Let the random variable K represent the number of images for which only one algorithm fails:

$$K = N_{10} + N_{01} \tag{6.17}$$

For $K = k$ and the null hypothesis, the random variable N_{10}, representing the number of images that A_1 misclassified and A_2 classified correctly, follows a binomial distribution

		A_2	
		Correct	**Incorrect**
A_1	**Correct**	n_{00}	n_{01}
	Incorrect	n_{10}	n_{11}

Table 6.1: Results summary for A_1 and A_2

$\mathcal{B} = (k, \frac{1}{2})$. The mean m of this distribution is $m = \frac{k}{2}$. The null hypothesis is tested by computing the probability P of a random variable V from $\mathcal{B} = (k, \frac{1}{2})$ using a two-tailed test as described in Kreyszig [44]:

$$P = \begin{cases} 2P(n_{10} \leq V \leq k) = 2\sum_{v=n_{10}}^{k} \begin{pmatrix} k \\ v \end{pmatrix} \left(\frac{1}{2}\right)^k & \text{if } n_{10} > k/2 \\ \\ 2P(0 \leq V \leq n_{10}) = 2\sum_{v=0}^{n_{10}} \begin{pmatrix} k \\ v \end{pmatrix} \left(\frac{1}{2}\right)^k & \text{if } n_{10} < k/2 \\ \\ 1.0 & \text{if } n_{10} = k/2 \end{cases} \tag{6.18}$$

where n_{10} is a specific value of the random variable N_{10}. A statistical significance level α is chosen and if it is found that $P < \alpha$, then the null hypothesis is rejected. Typical values of α are 0.05,0.01 or 0.001. Throughout this analysis, it is assumed that the errors made by an algorithm are independent.

6.3 HMM and Eigenface Comparison

In the following sections, the best top-bottom and P2D-HMMs are selected and compared with the best Eigenface results using McNemar's statistical test.

6.3.1 Top-bottom HMM vs Eigenfaces

The best performing top-bottom model was found to be $\mathcal{H} = (3, 1, 0)$, which scored an error rate of 13% with 26 misclassified images out of 200. The top-bottom HMM-based algorithm is denoted by A_1. The best Eigenface results were obtained using 175 or more eigenvectors, scoring an error rate of 10% with 20 misclassified images out of 200. The Eigenface algorithm is denoted by A_2. The results obtained with the two methods were analysed and the values of the N_{ij} random variables were computed. Table 6.2 summarises the results.

Using equation 6.18 with $n_{10} = 20$ and $k = 34$, the value of P can be calculated as $P = 0.3915$. The value of P indicates that the observed difference in performance between the top-bottom HMM and the Eigenface approach would arise by chance on about 40% of the occasions and that therefore the null hypothesis can not be rejected for any typical values of α.

		Eigenfaces	
		Correct	**Incorrect**
Top-bottom HMM	**Correct**	160	14
	Incorrect	20	6

Table 6.2: Top-bottom HMM vs Eigenfaces results

6.3.2 P2D-HMM vs Eigenfaces

The best performing P2D-HMM was found to be $\mathcal{P} = (3\text{-}6\text{-}6\text{-}6\text{-}3, 12, 8, 9, 6)$, which scored an error rate of 5.5% with 11 misclassified images out of 200. The P2D-HMM algorithm is denoted by A_1. The Eigenface algorithm is again denoted by A_2. The results obtained with the two methods were analysed and the values of the N_{ij} random variables were computed. Table 6.3 summarises the results.

Using equation 6.18 with $n_{10} = 8$ and $k = 25$, the value of P can be calculated as $P = 0.1078$. With this value of P, the null hypothesis cannot be rejected for typical values of α. The value of P indicates that the observed difference in performance between the P2D-HMM and the Eigenface approach would arise by chance on about 10% of the occasions. However, even though there is not strong statistical evidence, the results suggest that the P2D-HMM performs better than the Eigenface method when tested with the given database.

		Eigenfaces	
		Correct	**Incorrect**
P2D-HMM	**Correct**	172	17
	Incorrect	8	3

Table 6.3: P2D-HMM vs Eigenfaces results

Chapter 7

Domain And Resolution Experiments

This chapter investigates the effect of changing the image domain of representation and the image spatial resolution on the recognition performance of top-bottom and P2D-HMMs. The HMMs with best recognition results were selected and experiments were carried out on them using edge-detected images and images at lower spatial resolution. The results are summarised in the following sections.

7.1 Representation Domain Experiments

The choice of representation domain of image data often determines the degree of success of pattern classification applications. The challenge is to represent the content and the salient features of an image in a compact way, that can be efficiently and robustly used for search and recognition tasks. For example, frequency and frequency/space representation may yield better data separation, hence facilitating the recognition task, as reported by Wechsler [71]. Representation in the frequency domain can be obtained by taking the Fourier transform of the image. Figure 7.1 shows two face images and their compressed Fourier spectra as intensity images. The spectra were compressed[1] using a logarithmic function and scaled to 8 bits to facilitate visual analysis. Details of this enhancement technique are explained by Gonzalez and Woods [30]. It is evident that the structure arguments that were used to choose the HMM topology for the top-bottom and P2D models are no longer valid in the frequency domain. In the spatial domain, the facial features are clearly visible and they can be used to choose appropriate model parameters

[1] If $F(u,v)$, with u and v being frequency variables, denotes the discrete Fourier transform of the image, the compressed spectrum is obtained as $D(u,v) = c\log[\,1 + |F(u,v)|\,]$, where c is a scaling constant.

for the HMM. In the frequency domain, the spatial configuration is lost and no obvious alternative is available to constrain the model parameters.

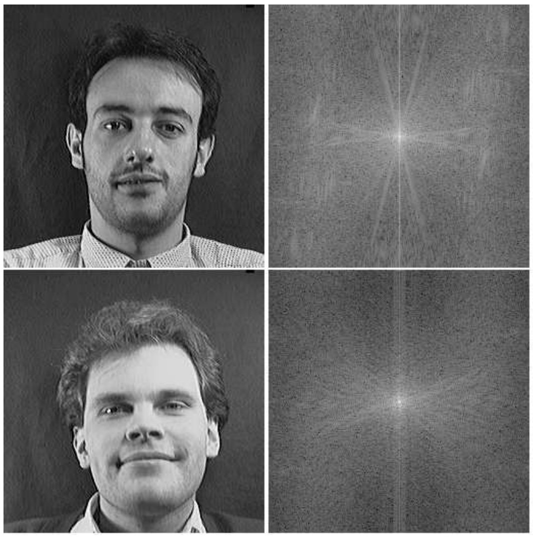


Figure 7.1: Face images and their compressed Fourier spectra

In order to retain some spatial information, experiments were carried out using a joint frequency/space representation. The images in the ORL database were converted to edge-detected images and used to train and test various HMMs. The edge detection technique was based on taking the Pythagorean sum of two perpendicular Sobel gradient operators, as detailed by Gonzalez and Woods [30]. Through edge detection, much of the texture information is lost, however the features can still be located with ease. Figure 7.2 shows the training data segmentation and the magnified states for the top-bottom model with parameters $\mathcal{H} = (5, 8, 7)$. The relevant facial features, such as the eyes, are isolated successfully, even in the absence of the full texture of the face.

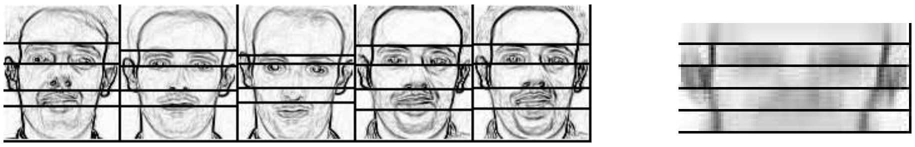


Figure 7.2: Segmentation and states for edge images using a top-bottom HMM

All the images in the ORL database of faces were passed through an edge-detector filter, thus generating a new database of 400 edge-detected images. As for previous experiments, 5 images for each subject were used to train a HMM and the remaining 5 were used for testing. In this section, the results obtained testing three HMMs with the edge-detected database are presented. The HMMs were the two best top-bottom model $\mathcal{H} = (3, 1, 0)$, $\mathcal{H} = (5, 1, 0)$ and the best P2D-HMM $\mathcal{P} = (3\text{-}6\text{-}6\text{-}6\text{-}3, 12, 8, 9, 6)$. Two top-bottom models were used, since the results obtained with the apparently best top-bottom HMM (which had only 3 states and an error rate of 13%) could have been accidental. The top-bottom model with 5 states and an error rate of 13.5% was also used.

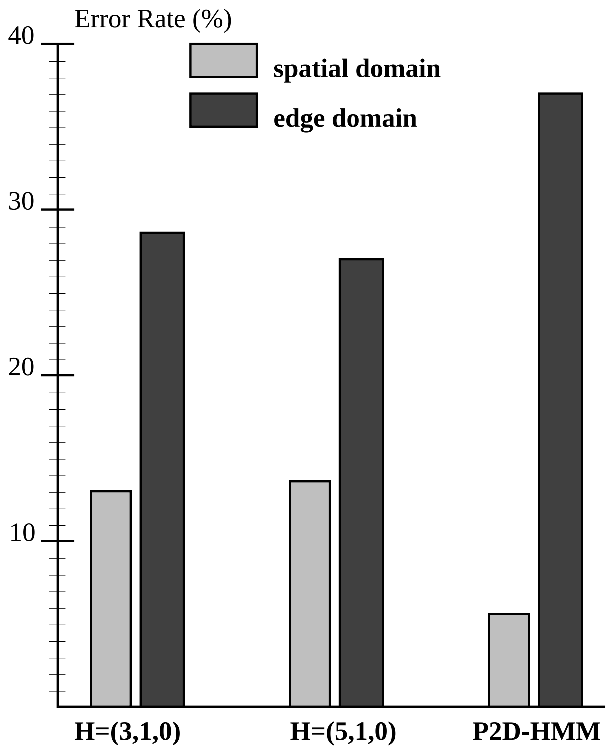


Figure 7.3: Space vs edge domain recognition results

The recognition results are shown in figure 7.3. The recognition performance is worse in the edge domain for both the top-bottom and the P2D models. The performance of the top-bottom models has approximately halved in the edge domain, while the P2D-HMM results worsen by a factor of about 7. The loss of texture information, which does not dramatically affect the segmentation process (as shown also in figure 7.4, where a segmented edge image for the P2D-HMM is shown), causes the recognition performance to drop.

7.2 Spatial Resolution Experiments

In this section, the three models used for the edge domain experiments above are re-adapted to deal with images at different spatial resolutions. It is of interest to investigate

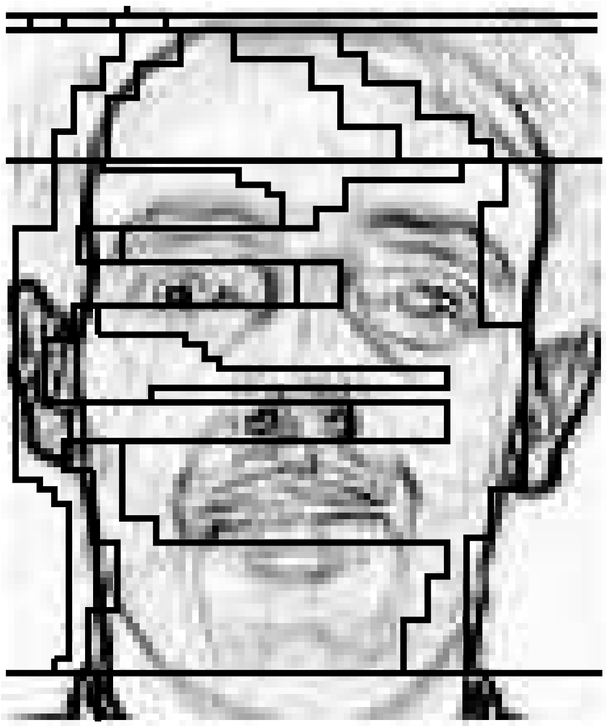


Figure 7.4: Segmentation of edge image using a P2D-HMM

how the recognition performance varies, as the image resolution is decreased, because images at lower resolutions occupy less storage space and run faster through the Viterbi recogniser. For humans, it is reasonable to hypothesise that when moving from coarse to fine spatial quantisation, processing efficiency should increase, since both local feature adequacy and the adequacy of global configurational measures will increase, as indicated by Bachmann [3]. Also, it is expected that the recognition performance will only improve up to a certain level, beyond which recognition efficiency will not improve with any further increase in spatial resolution.

The experiments presented here were obtained with images at 4 different resolutions. Table 7.1 shows the results obtained using the two best top-bottom HMMs. The values p_M^t and p_I^t are those defined in equations 4.2 and 4.4 respectively. The figures reported in the table are rounded to two significant figures.

Resolution (width x height)	**Model**	p_M^t	p_I^t	**Err.Rate**
92x112	$\mathcal{H} = (3,1,0)$	190	10,000	13%
46x56	$\mathcal{H} = (3,1,0)$	100	2,500	17%
23x28	$\mathcal{H} = (3,1,0)$	50	650	19.5%
12x14	$\mathcal{H} = (3,1,0)$	30	160	18.5%
92x112	$\mathcal{H} = (5,1,0)$	190	10,000	13.5%
46x56	$\mathcal{H} = (5,1,0)$	100	2,500	13.5%
23x28	$\mathcal{H} = (5,1,0)$	60	650	14%
12x14	$\mathcal{H} = (5,1,0)$	30	160	17%

Table 7.1: Resolution experiment results for top-bottom HMMs

The recognition results are summarised in figure 7.5. As the resolution halves in each dimension, the storage requirement for each model halves and the storage requirement for each image decreases by a factor of 4. The recognition performance, however, decreases only slightly and the error rate is below 20% even at lower resolutions. With the 5-state model, the difference in performance between the full resolution images and the images at 23x28 is 0.5%. However, the storage requirements of the models and images with images at a resolution of 23x28 are approximately 4 and 16 times smaller respectively.

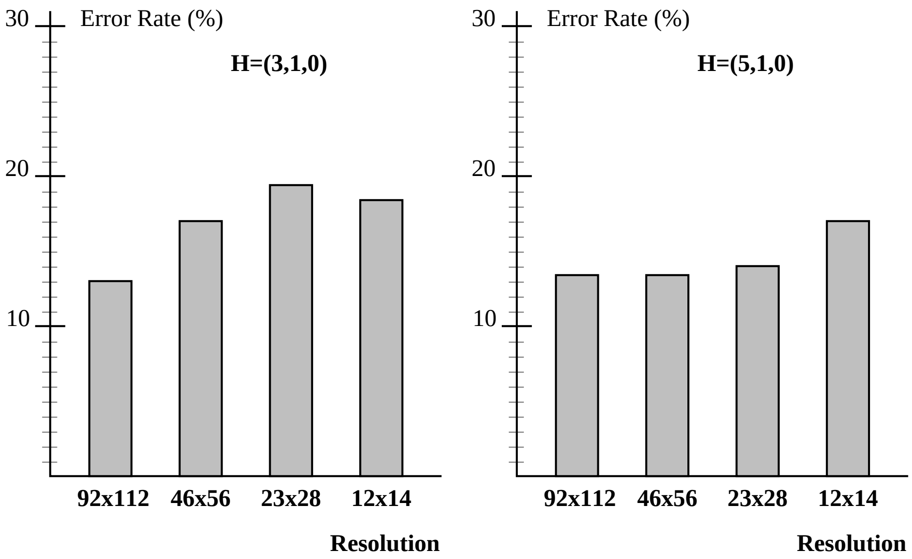


Figure 7.5: Performance vs resolution for top-bottom HMMs

Resolution (width x height)	**Model**	p_M^p	p_I^p	**Err.Rate**
92x112	$\mathcal{P} = (3\text{-}6\text{-}6\text{-}6\text{-}3, 12, 8, 9, 6)$	240	130,000	5.5%
46x56	$\mathcal{P} = (3\text{-}6\text{-}6\text{-}6\text{-}3, 6, 4, 4, 3)$	90	26,000	6,5%
23x28	$\mathcal{P} = (4\text{-}5\text{-}5\text{-}4, 3, 3, 2, 2)$	50	5,000	6%
12x14	$\mathcal{P} = (4\text{-}5\text{-}5\text{-}4, 2, 2, 1, 1)$	40	600	12%

Table 7.2: Resolution experiment results for the P2D-HMM

The same lower spatial resolution images were tested with the best performing P2D-HMM. At the full 92x112 resolution, the model with parameters $\mathcal{P} = (3\text{-}6\text{-}6\text{-}6\text{-}3, 12, 8, 9, 6)$ was initially chosen. As the resolution was decreased, the model parameters were adjusted, trying to approximately preserve the initial ratio of sampling window size to image size, and changing the number of states according to intuition. The results are summarised in table 7.2.

Figure 7.6 graphically summarises the recognition results for the various P2D-HMMs at different resolutions. There is very little difference in performance for image sizes down to 23x28 and at 12x14 the error rate is 12%. The difference in performance between the full resolution case and the case with images at 23x28 is 0.5%, but the storage space required by the images is approximately 25 times smaller with the 23x28 images.

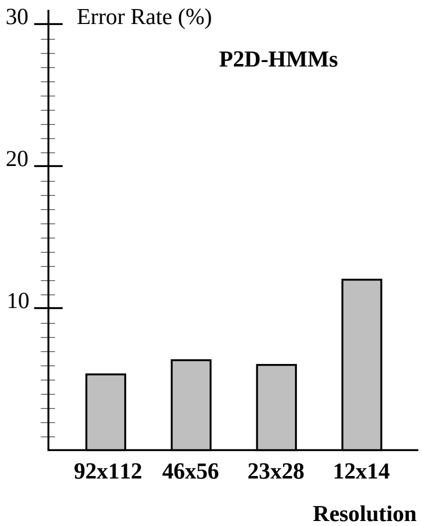


Figure 7.6: Performance vs resolution for P2D-HMMs

In summary, the resolution experiments presented in this section indicate that considerable storage savings can be obtained at little performance cost by reducing the spatial resolution of the database images. As the size of the models and images also affects the speed of training and recognition, the HMMs with lower resolution images train and run more quickly.

Chapter 8

Automatic Face Location

The recognition experiments presented in the previous chapters were obtained using the ORL database of faces, which contained a collection of face images that had been manually located and cropped. In this chapter, the HMM based approach is tested on images that are cropped automatically using a model-based method. The face model is defined first, and a technique based on genetic algorithms is used to find the location of the face in a 256 pixel-wide by 256 pixel-high image. This technique is employed to locate faces in 400 uncropped images, which are then cropped and used to train and test a P2D-HMM. The automatic face location ideas presented in this chapter were implemented by Heap [34] in his final year undergraduate project[1] on human hand tracking.

8.1 Building the Face Model

Automatic face location has a number of useful applications. Ponticos [55] described a system based on motion detection to enhance the quality of video-phone images around the user's face. In the context of this dissertation, still images are used and it is therefore not possible to make use of motion information. In order to locate the face in the image, a face model is defined as a collection of 42 control points of a 2D wire frame structure, as shown in figure 8.1.

This model will be referred to as a Shape Model (SM), which is a set of labelled points joined together, in this case, by straight lines. In order to create a generic face model, a number of examples were collected and used to train the face model.

[1] The work was carried out under the author's supervision

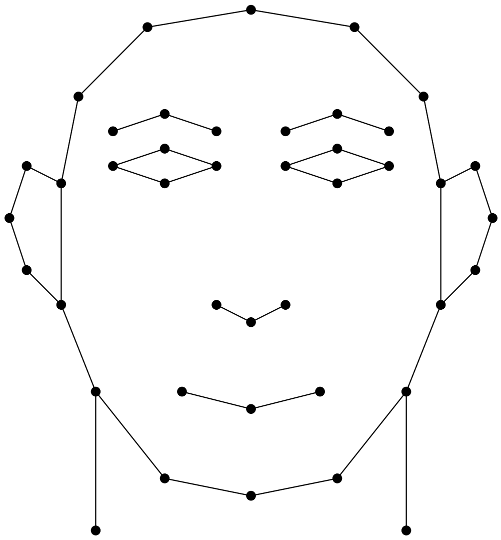


Figure 8.1: Wire frame model of the face

8.1.1 Training the face model

A technique for training the SM using a set of examples was suggested by Cootes *et al.* [15] and is briefly described here. Given a training set containing examples of faces, a vector is formed for each example by manually placing the control points of the wire frame model on the images. In the case of the experiments presented in this chapter, ten 256x256 images (one image each of ten different people) were used for training and for each image, the co-ordinates of the 42 control points were recorded as a vector:

$$\mathbf{m}_i = (x_0, y_0, \ldots, x_{41}, y_{41})' \tag{8.1}$$

where $1 \leq i \leq 10$ and prime indicates transpose. In order to derive any useful statistics from the 10 examples, the shapes have to be aligned as accurately as possible. The alignment technique described here is based on defining an alignment operator A, that transforms each pair of points in a model $\mathbf{m}_i$, by scaling them by s, rotating them by θ and translating them by t_x horizontally and t_y vertically. After applying A, each pair of points (x, y) of $\mathbf{m}_i$ is transformed to a new pair of points (x_A, y_A) defined as:

$$x_A = sx\cos\theta - sy\sin\theta + t_x \tag{8.2}$$

$$y_A = sx\sin\theta + sy\cos\theta + t_y \tag{8.3}$$

Shape $\mathbf{m}_1$ is taken as reference and the remaining 9 shapes are aligned to it. In order to align $\mathbf{m}_2$ to $\mathbf{m}_1$, the operation A is applied to $\mathbf{m}_2$ and the values of s, θ, t_x and t_y that best align the two shapes are determined using a least square approach that minimises the expression:

$$E = (\mathbf{m}_1 - A(\mathbf{m}_2))' \; D \; (\mathbf{m}_1 - A(\mathbf{m}_2)) \tag{8.4}$$

where D is a diagonal matrix of weights, which gives emphasis to points that are more stable over the training set. One way to provide a measure of the stability of a point is to find how much that point moves over the training set relative to the other points. This is achieved by calculating the variance across the training set of the normalised distance of the point from all the other points in the model and taking the inverse of the sum of all such variances. If the distance between point k and point l in the ith training example is defined as R^i_{kl}, then the normalised distance N^i_{kl} between the two points is defined as:

$$N^i_{kl} = \frac{R^i_{kl}}{S_i} \tag{8.5}$$

where the normalisation factor S_i is the sum of the distances of all pairs of points for that training example, i.e.:

$$S_i = \sum_{k=0}^{41}\sum_{l=0}^{41} R^i_{kl} \tag{8.6}$$

By defining V_{kl} as the variance across the training set of the normalised distance between point k and l, the weight d_k for the kth point in the model can then be found as:

$$d_k = \left(\sum_{l=0}^{41} V_{kl}\right)^{-1} \tag{8.7}$$

The diagonal matrix D is thus constructed as:

$$D = \begin{pmatrix} d_0 & 0 & 0 & \cdots & 0 \\ 0 & d_0 & 0 & & \vdots \\ 0 & 0 & \ddots & 0 & 0 \\ \vdots & & 0 & d_{41} & 0 \\ 0 & \cdots & 0 & 0 & d_{41} \end{pmatrix} \tag{8.8}$$

After the 9 shapes are aligned to the reference, the mean of all ten aligned shapes is found and is used as a new reference. The ten training shapes are aligned to the mean, and this process is repeated iteratively, until some form of convergence is achieved. The final mean vector is denoted as $\mathbf{m}_0$ and is the model that will be used to locate the head in other images.

8.1.2 Scoring the face model

In order to determine the position of the face in an image using the SM, a scoring function is needed to assess the suitability of the model in any given position. For a trained model

$\mathbf{m}_0$ and a position A defined by the four variables s, θ, t_x and t_y, the model $\mathbf{m}_A$ is defined as the model $\mathbf{m}_0$ transformed through A, i.e:

$$\mathbf{m}_A = A(\mathbf{m}_0) \tag{8.9}$$

At regular intervals along the boundaries of $\mathbf{m}_A$, a line of pixels normal to the boundary is extracted and the edge strength along that line is calculated by differencing the grey-level values of adjacent pixels. The total number of normals along the model boundary is fixed. The number of pixels along each normal depends on the size of the model and is changed dynamically.

Figure 8.2 shows an example of normals along the model boundary. By summing up the scores[2] of all the normals, an overall score is obtained to represent the fitness of $\mathbf{m}_A$.

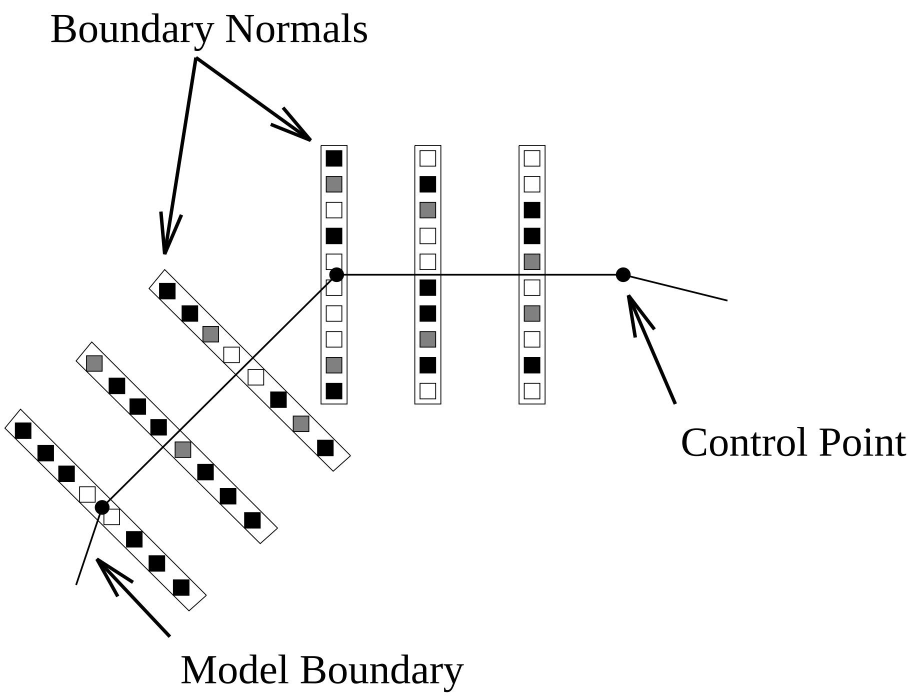


Figure 8.2: Normals along model boundaries

Given a test image and a face model $\mathbf{m}_0$, the goal is to locate the face in the test image as accurately as possible, by finding the position for the model which gives the highest score. An exhaustive search through all the possible values of the four parameters of A would require a very large number of operations. In order to reduce the search space,

[2]The scores were obtained by multiplying the edge values along the normals by a weight vector, the coefficents of which decreased with distance from the boundary

a method based on genetic algorithms is used, an account of which is given in the next section.

8.2 The Genetic Algorithm Approach

Genetic Algorithms (GAs) perform search tasks based on mechanisms similar to those of natural selection and genetics. The detailed presentation of the principles involved in the use of GAs is beyond the scope of this dissertation, and a full account on GAs can be found in Goldberg [28]. The use of GAs in model-based image interpretation was introduced by Hill and Taylor [36] and the implementation described here is based on their work.

Given a test image and $\mathbf{m}_0$, the goal was to find the values of the four parameters of A that maximised the scoring function. The scale and rotation parameters of A were coded into two orthogonal parameters p and q for convenience as follows:

$$p = s\cos\theta \tag{8.10}$$

$$q = s\sin\theta \tag{8.11}$$

Therefore, the four parameters defining the transformation A were p, q, t_x and t_y, and were encoded in a 32-bit code as illustrated in figure 8.3.

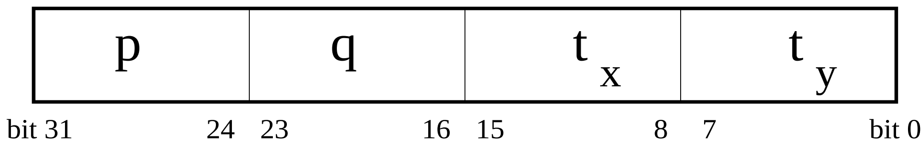


Figure 8.3: Genetic code representing the transformation A

The basic principle used in the GA experiments was that if a 32-bit code was found to generate a high score, then it was assumed that other good codes existed nearby in the search space. In the first instance, a population of 50 random 32-bit codes was generated and each code was given a score. A new sample of 50 codes was chosen with replacement from the original random population, with probabilities proportional to the code's scores. This ensured that the codes with higher scores were more likely to be selected. The processes of crossover and mutation were applied to the new sample. Crossover consisted of taking two codes and swapping bit portions between them at random places. Mutation consisted of introducing random bits in the code. After crossover and mutation, a new population of 50 codes was generated. This process was repeated for 100 iterations. At the end, the code with the highest score from the last population was selected as the overall winner, revealing the position, scale and rotation of the face in the test image.

8.3 Automatic Location Results

The GA-based face location system[3] was tested using a collection of 400 uncropped 256x256 images. The images were heads against a dark, homogeneous background. Training was carried out using the SM of figure 8.1 and 10 images, for which the control points were located manually. Initial experiments with a SM describing only the outer face boundary had not produced good results, and it was decided to add internal face features, such as the eyes, the eyebrows, the nose and the mouth. After aligning the training set, the mean shape was calculated and used to locate the face in the 400 uncropped images. The GA technique is not deterministic and its output is likely to change at each run. The method is also prone to be attracted to local maxima and converge to regions of little interest. For these reasons, it was decided that the 100-iteration GA procedure should be repeated 20 times. The highest overall score was then chosen as best match. By repeating the procedure 20 times, the probability of the overall winner being a true good match was increased.

The results obtained by the 20 runs of the GA for one of the images in the database are shown in figure 8.4, where an uncropped image is shown, followed by the same image with the best model match superimposed on it and finally the cropped face. The cropped image was obtained by drawing a rectangle around the model edges.

Figure 8.4: Face location using GAs

The location results were inspected visually and were found to be accurate for most

[3]The GA experiments were carried out with a mutation rate of 0.004 and a crossover rate of 0.5.

images in the database. In current work at the Olivetti Research Laboratory in Cambridge, the GA approach is being tested with colour, cluttered background images. The work is carried out within the framework of Medusa, a system for orchestrating networked multimedia devices, such as cameras, and prototyping media processing functions and applications, described by Wray *et al.* [74]. The initial results obtained indicate an equally successful performance, using images captured in a standard office environment. In other work by Hill *et al.* [35], the GA search was used in combination with active shape model refinement to improve the results. In the context of this dissertation, the cropped images obtained from the GA results are sufficient to be used by a P2D-HMM, which does not require accurate alignment of features, and therefore active shape model refinement was not investigated. The results obtained with a P2D-HMM trained and tested with the automatically cropped images generated from the GA are presented in the next section.

8.4 Recognition of Automatically Cropped Images

The GA routine was used to crop 200 full resolution (256x256), 8-bit grey-level images of 40 different subjects (5 images for each subject). The 200 cropped images were used to train a HMM. The block diagram shown in figure 8.5 illustrates the training process.

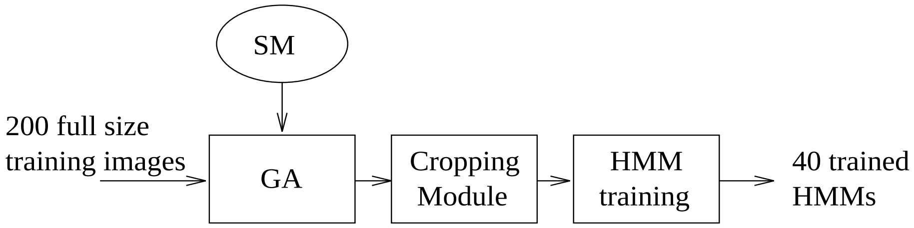


Figure 8.5: HMM training with automatically cropped images

Similarly, the GA routine was used to crop 200 other full resolution images, which were used to test the HMM. The process is illustrated in figure 8.6.

A 24-state P2D-HMM was chosen for the experiments presented in this section. P2D-HMMs generally require little horizontal and vertical alignment, and their performance with the automatically cropped images is expected to be similar to that obtained with the manually cropped images of the ORL database. The resolution results presented in chapter 7 indicated that the recognition performance was little affected by reducing the image resolution. It was therefore decided to re-scale the automatically cropped

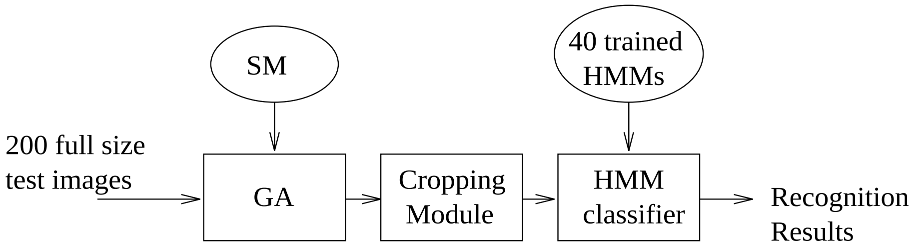


Figure 8.6: HMM testing with automatically cropped images

images to an image resolution of 46x56. The P2D-HMM used for these experiments was parameterised as:

$$\mathcal{P} = (24, 3\text{-}6\text{-}6\text{-}6\text{-}3, 6, 4, 4, 3)$$

Using the ORL database of faces, this model had an error rate of 6.5%. Experiments were carried out with the automatically cropped images a number of times, as the GA process produces a different output every time it is run. The error rates obtained varied between 4% and 5% (i.e. from 8 errors out of 200 images, to 10 errors out of 200 images). These figures are very similar to the error rates obtained with the manually cropped images. These results appear to indicate that the HMM's performance is unaffected when using images cropped automatically. The system described contains all the components of a fully automated system.

Chapter 9

Conclusions

This dissertation has presented a novel approach for modelling human faces, for the purpose of recognition, based on HMMs. The potential benefits of this approach were investigated and this chapter concludes the investigation by summarising the experimental results and indicating directions of future work.

9.1 Summary of Results

9.1.1 Model assessment experiments

It was first shown how the face images could be represented using top-bottom models. These models made use of the structural information contained in the face and successful recognition rates of around 85% were obtained. It was shown how the face could be segmented into horizontal bands and how these bands were similar to the bands a human would have drawn based on subjective judgement. The model parameterisation was investigated through an extensive set of experiments: by varying the model and the sampling technique, the parameters resulting in the lowest error rates were determined. Top-bottom models, however, took advantage only of the structural information of the face along the vertical direction, by segmenting the face into horizontal bands. The facial features related to each other in a top-bottom sense and the 2D structure of the image was little exploited.

In order to make use of the 2D information contained in the face, P2D-HMMs were introduced. By representing the data using a combination of left-right models arranged in an ordered sequence, a pseudo-2-dimensional representation of the face was obtained. It was shown how an equivalent standard 1D HMM could be constructed to simulate the behaviour of the P2D-HMM, and recognition experiments were carried out with the ORL database. Experiments were set up to investigate the effect of the various parameters on the performance of the model and successful recognition rates of around 95% were obtained.

9.1.2 Performance assessment experiments

In order to assess the performance of the HMM-based approach, a comparison was presented with one of the best known face recognition algorithms. The Eigenface algorithm, based on principal component analysis, was tested using the ORL database of faces. By varying the number of Eigenfaces used to define the feature sub-space, various experiments were carried out and the model with the highest success rate (around 90%) was selected. A statistical comparison with the best performing HMMs, both top-bottom and P2D-HMMs, was carried out. From the analysis, it was found that the observed difference in performance between Eigenfaces and top-bottom HMMs in favour of Eigenfaces would arise by chance on about 40% of the occasions. On the other hand, the observed difference between P2D-HMMs and Eigenfaces in favour of P2D-HMMs, would arise by chance on approximately 10% of the occasions. Thus, although the statistical evidence is not strong, there is evidence to suggest that the P2D-HMM system is superior to the Eigenface method for the given database.

The performance of the HMMs was analysed using edge-detected images and images at lower spatial resolution. It was found that, in the edge domain, the performance dropped approximately by a factor of 2 for top-bottom HMMs and by more than 6 times for the P2D-HMM with the best performance. The segmentation results in the edge domain, however, were similar to those obtained in the intensity domain. The results obtained for images at lower spatial resolutions indicated that the performance was almost unaffected for images as small as 23 pixels wide and 28 pixels high. At this resolution, however, the storage space required for the models and images in the database was significantly lower than for the full resolution case. Naturally, with the smaller resolution images, the model training and the recognition of the test images could be carried out more quickly.

Finally, a shape model and a genetic algorithm technique were used to locate the face in an image. Based on this technique, 400 faces were automatically cropped and used to train and test a P2D-HMM. It was found that the performance of the P2D-HMM was practically the same as for the case with the manually cropped database, indicating that the model could be used in a fully automated system.

9.2 Limits and Shortcomings

The HMM approach has been shown to yield satisfactory recognition rates. By exploiting the inherent ability of HMMs to segment data automatically, successful recognition results were obtained with limited initial guidance. However, while experimenting with HMMs, some shortcomings became evident.

HMMs are processor intensive models. Depending on the parameterisation used, the Viterbi algorithm can require a large number of calculations. This implies that sometimes the algorithm runs slowly. For example, using a Sun Sparc II workstation, the P2D-HMM with parameters $\mathcal{P} = (3\text{-}6\text{-}6\text{-}6\text{-}3, 12, 8, 9, 6)$ classified full resolution images at a speed of approximately 4 minutes per image.

The HMM analysis is based on grey-level templates and therefore lighting plays an important role. The success of the model partly depends on the training data used. Initial informal experiments with equal lighting conditions for all training images revealed that the HMM would have low success rates in classifying images taken under different lighting conditions. In order to compensate for this, it was decided that the ORL database should have training images captured under a variety of lighting conditions.

Finally, the P2D-HMM segmentation results were unsatisfactory, despite the improvement in recognition rates obtained with these models. While the top-bottom HMM segmentation matched expectations, the segmentation obtained with P2D-HMMs appeared quite erratic.

9.3 Future Work

9.3.1 Face recognition work

Computer facilities are becoming widespread. Information super-highways, optical fibre networks such as those of the Granta Project in Cambridge, are being laid out to connect cities, universities, companies and the home. Multimedia applications, in which computers exchange video and audio data, are becoming commonly available. Computers in the office and in the home will be equipped with video cameras and face recognition in this context will represent a valuable tool. New applications will emerge: it will be possible to index video documents dynamically, using the knowledge of who appears in them. Computers will not switch on unless the user is recognised as an authorised one. By knowing which user is in a certain room through face recognition, it will be possible to personalise dynamically the equipment in that room according to the preferences of the user. The rapid expansion of the home entertainment industry will bring more applications: a user may want to fast-forward through a movie, until a scene featuring a specific actor is reached.

In the light of this scenario, it is planned to develop the ideas introduced in this dissertation into a large scale system capable of recognising faces in real-time. The framework for this system is the ubiquitous deployment of digitally networked video cameras at the Olivetti Research laboratory in Cambridge. Inside the laboratory, offices are equipped with several camera sources. The Medusa application environment makes streams of live video images available to software processing elements, which can be distributed elsewhere on the network. Instances of a processor bank, a network module with a number of processors, will provide a computation resource, which can be allocated dynamically to the various tasks involved in face recognition. Initially, the system is going to be employed to discriminate between approximately 40 people (Olivetti and University staff). Further experiments will be required to establish if the HMM-based technique can scale up to model a larger database of subjects.

In order to make the system more tolerant to orientation changes, individual models will be trained for views of the same subject at different orientations to the camera. Test images will be matched against the models of different subjects and head orientations. Experiments will be carried out to investigate how the recognition performance is affected by introducing different orientations.

9.3.2 Broader work

The HMM approach presented in this dissertation was applied to face images, but it can in principle be used for any image. Work with HMMs on generic images has already been carried out, as detailed in chapter 3. In most cases the observation vectors represented a set of features extracted from the images. In this dissertation, the observation vectors are image templates, used as input to a continuous density HMM. This approach has the advantage that structural information can be integrated easily in the model. The approach therefore can be extended to any image, the structure of which can be modelled by a set of interconnected states, each representing features in the image.

9.4 Summary

This dissertation has presented a novel approach to face recognition based on continuous density HMMs. Experiments were presented to assess the plausibility of the approach and to investigate its performance, as model parameters, model structure, image resolution and image domain were varied. Through the integration of structural and statistical information, image segmentation and feature extraction were carried out automatically. The HMM approach gave satisfactory recognition rates of up to around 95% and the method is currently being implemented as a real-time system, using the insights obtained from the experimental results presented in this dissertation.

Bibliography

[1] O.E. Agazzi, S. Kuo, E. Levin, and R. Pieraccini. Connected and degraded text recognition using planar hidden markov models. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, V:113–116, 1993.

[2] N.M. Allinson, A.W. Ellis, B.M. Flude, and A.J. Luckman. A connectionist model of familiar face recognition. *IEE Colloquium on "Machine Storage and Recognition of Faces"*, 5:1-10(Digest No: 1992/017), 1992.

[3] T. Bachmann. Identification of spatially quantised tachistoscopic images of faces: how many pixels does it take to carry identity? In V. Bruce, editor, *Face Recognition*, pages 87–103. Laurence Erlbaum Associates, 1991.

[4] R.J. Baron. Mechanisms of human facial recognition. *International Journal on Man-Machine Studies*, 15:137–178, 1981.

[5] L.E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities*, III:1–8, 1972.

[6] L.E. Baum and G.R. Sell. Growth transformations for functions on manifolds. *Pacific Journal of Mathematics*, 27(2):211–227, 1968.

[7] P. Benson and D. Perrett. Face to face with the perfect image. *New Scientist*, pages 32–35, 22 February 1992.

[8] P.J. Benson and D.I. Perrett. Perception and recognition of photographic quality facial caricatures:implications for the recognition of natural images. In V. Bruce, editor, *Face Recognition*, pages 105–135. Laurence Erlbaum Associates, 1991.

[9] V. Bruce. *Recognising Faces.* Laurence Erlbaum Associates, 1988.

[10] V. Bruce and A.W. Young. Understanding face recognition. *British Journal of Psychology*, 77:305–327, 1986.

[11] R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, 1993.

[12] N. Cantor and W. Mischel. Prototypes in person perception. In L. Berkowitz, editor, *Advances in Experimental Social Psychology*, volume 12, pages 3–52. Academic Press, 1979.

[13] T.J. Cham. Persons verification (face recognition). Cambridge University Engineering Department, Final Year Undergraduate Project, 1993.

[14] J. Chen and A. Kundu. Rotation and gray scale transform invariant texture identification using wavelet decomposition and Hidden Markov Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):208–214, 1994.

[15] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Training models of shape from sets of examples. In *British Machine Vision Conference*, pages 9–18. Springer-Verlag, 1992.

[16] G.W. Cottrell and M. Fleming. Face recognition using unsupervised feature extraction. *International Neural Network Conference*, 1:322–325, 1990.

[17] I. Craw. Recognising face features and faces. *IEE Colloquium on "Machine Storage and Recognition of Faces"*, 7:1-4(Digest No: 1992/017), 1992.

[18] I. Craw and P. Cameron. Face recognition by computer. In *British Machine Vision Conference*, pages 488–507. Springer-Verlag, 1992.

[19] J. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America A*, 2(7):1160–1169, July 1985.

[20] J.G. Daugman. High confidence visual recognition of persons by a test of statistical independence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11), 1993.

[21] R. Diamond and S. Carey. Why faces are and are not special: an effect of expertise. *Journal of Experimental Psychology: General*, 115:107–117, 1986.

[22] P. Ekman and W.V. Friesen. Measuring facial movement with the Facial Action Coding System. In P. Ekman and K.R. Scherer, editors, *Emotion in the human face*, pages 178–211. Cambridge University Press, 1982.

[23] G.D. Forney. The Viterbi Algorithm. *Proceedings of the IEEE*, 61(3):268–278, March 1973.

[24] R.C. Fowler. FINGERPRINT: an old touchstone decriminalised. *IEEE Spectrum*, page 26, February 1994.

[25] R. Gallery and T.I.P. Trew. An architecture for face classification. *IEE Colloquium on 'Machine Storage and Recognition of Faces'*, 2:1-5(Digest No: 1992/017), 1992.

[26] F. Galton. Personal identification and description 1. *Nature*, pages 173–177, 21 June 1888.

[27] L. Gillick and S.J. Cox. Some statistical issues in the comparison of speech recognition algorithms. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 532–535, 1989.

[28] D.E. Goldberg. *Genetic algorithms in search, optimisation, and machine learning.* Addison-Wesley, 1989.

[29] A.J. Goldstein, L.D. Harmon, and A.B. Lesk. Identification of human faces. *Proceedings of the IEEE*, 59(5):748–760, 1971.

[30] R.C. Gonzalez and R.E. Woods. *Digital Image Processing.* Addison-Wesley, 1992.

[31] L.D. Harmon, M.K. Khan, R. Lasch, and P.F. Ramig. Machine identification of human faces. *Pattern Recognition*, 13(2):97–110, 1981.

[32] D.C. Hay and A.W. Young. The human face. In A.W. Ellis, editor, *Normality and pathology in cognitive functions*, pages 173–202. Academic Press, 1982.

[33] Y. He and A. Kundu. 2-D shape classification using Hidden Markov Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(11):1172–1184, 1991.

[34] A.J. Heap. Object location using snakes. Cambridge University Computer Laboratory, Final Year Undergraduate Project, 1994.

[35] A. Hill, T.F. Cootes, and C.J. Taylor. A generic system for image interpretation using flexible templates. In *British Machine Vision Conference*, pages 276–285. Springer-Verlag, 1992.

[36] A. Hill and C.J. Taylor. Model-based image interpretation using genetic algorithms. In *British Machine Vision Conference*, pages 266–274. Springer-Verlag, 1991.

[37] C.-L. Huang and C.-W. Chen. Human facial feature extraction for face interpretation and recognition. *Pattern Recognition*, 25(12):1435–1444, 1992.

[38] T. Hutcheson. FACE: smile,you're on candid camera. *IEEE Spectrum*, pages 28–29, February 1994.

[39] R.A. Hutchinson and W.J. Welsh. Comparison of neural networks and conventional techniques for feature location in facial images. *IEE International Conference on Artificial Neural Networks*, Conf. Publication Number 313:201–205, 1989.

[40] T. Kanade. Computer recognition of human faces. In *Interdisciplinary Systems Research*. Birkhäuser Verlag, 1977.

[41] M. Kass, A. Witkin, and D. Terzopoulos. SNAKES: Active Contour Models. *Proceedings of the International Conference on Computer Vision*, pages 259–268, 1987.

[42] M. Kirby and L. Sirovich. Application of the Karhunen-Loéve procedure for the characterisation of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):103–108, 1990.

[43] T. Kohonen. *Self-organization and associative memory*. Springer-Verlag, 2nd edition, 1988.

[44] E. Kreyszig. *Advanced Engineering Mathematics*. Wiley, sixth edition, 1988.

[45] A. Kundu, Y. He, and P. Bahl. Recognition of handwritten word: first and second order Hidden Markov Model based approach. *Pattern Recognition*, 22(3):283–297, 1989.

[46] S. Kuo and O.E. Agazzi. Machine vision for keyword spotting using pseudo 2D Hidden Markov Models. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, V:81–84, 1993.

[47] A. Lanitis, C.J. Taylor, and T.F. Cootes. An automatic face identification system using flexible appearance models. In *British Machine Vision Conference*, volume 1, pages 65–74. BMVA Press, 1994.

[48] E. Levin and R. Pieraccini. Dynamic planar warping for optical character recognition. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, III:149–152, 1992.

[49] L.L. Light, F. Kayra-Stuart, and S. Hollander. Recognition memory for typical and unusual faces. *Journal of Experimental Psychology: Human Learning and Memory*, 5:212–228, 1979.

[50] R. Mandelbaum. SPEECH: just say the word. *IEEE Spectrum*, page 30, February 1994.

[51] B. Miller. Vital signs of identity. *IEEE Spectrum*, pages 22–30, February 1994.

[52] L. Najman, R. Vaillan, and E. Pernot. Face from sideview to identification. In G. Vernazza, A.N. Venetsanopoulos, and C. Braccini, editors, *Image Processing: Theory and Applications*. Elsevier Science Publishers, 1993.

[53] O. Nakamura, S. Mathur, and T. Minami. Identification of human faces based on isodensity maps. *Pattern Recognition*, 24(3):263–272, 1991.

[54] D.I. Perrett, P.A.J. Smith, D.D. Potter, A.J. Mistlin, A.S. Head, A.D. Milner, and M.A. Jeeves. Visual cells in the temporal cortex sensitive to face view and gaze direction. *Proceedings of the Royal Society of London*, B223:293–317, 1985.

[55] C. Ponticos. A robust real time face location algorithm for videophones. In *British Machine Vision Conference*, volume 2, pages 449–458. BMVA Press, 1993.

[56] L.R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[57] L.R. Rabiner and B-H. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, Englewood Cliffs, NJ, 1993.

[58] C.S. Ramsay, K. Sutherland, D. Renshaw, and P.B. Denyer. A comparison of vector quantization codebook generation algorithms applied to face recognition. In *British Machine Vision Conference*, pages 508–517. Springer-Verlag, 1992.

[59] G. Robertson and I. Craw. Testing face recognition systems. In *British Machine Vision Conference*, volume 1, pages 25–34. BMVA Press, 1993.

[60] M. Rydfalk. Candide, a parameterised face. Technical Report LiTH-ISY-I-0866, Department of Electrical Engineering, Linköping University, Sweden, 1987.

[61] A. Samal and P.A. Iyengar. Automatic recognition and analysis of human faces and facial expressions: A survey. *Pattern Recognition*, 25(1):65–77, 1992.

[62] J. Shepherd, G. Davies, and H. Ellis. Studies of cue saliency. In G. Davies, H. Ellis, and J. Shepherd, editors, *Perceiving and remembering faces*, pages 105–131. Academic Press, 1981.

[63] J.W. Shepherd, F. Gibling, and H.D. Ellis. The effect of distinctiveness, presentation time and delay on face recognition. In V. Bruce, editor, *Face Recognition*, pages 137–145. Laurence Erlbaum Associates, 1991.

[64] D. Sidlauskas. HAND: give me five. *IEEE Spectrum*, pages 24–25, February 1994.

[65] J.E. Siedlarz. IRIS: more detailed than a fingerprint. *IEEE Spectrum*, page 27, February 1994.

[66] T.J. Stonham. Practical face recognition and verification with WISARD. In H.D. Ellis, M.A. Jeeves, F. Newcombe, and A. Young, editors, *Aspects of face processing*, pages 426–441. Martinus Nijhoff Publishers, 1986.

[67] K. Sutherland, D. Renshaw, and P.B. Denyer. A novel automatic face recognition algorithm employing vector quantisation. *IEE Colloquium on "Machine Storage and Recognition of Faces"*, 4:1-4(Digest No: 1992/017), 1992.

[68] C.W. Therrien. *Decision estimation and classification*. Wiley, 1989.

[69] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[70] R. Want and A. Hopper. Active badges and personal interactive computing objects. *IEEE Trans. on Consumer Electronics*, February 1992.

[71] H. Wechsler. *Computational Vision*. Academic Press, San Diego CA, 1990.

[72] E. Winograd. Elaboration and distinctiveness in memory for faces. *Journal of Experimental Psychology: Human Learning and Memory*, 7:181–190, 1981.

[73] K.H. Wong, H.H.M. Law, and P.W.M. Tsang. A system for recognising human faces. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 1638–1642, 1989.

[74] S. Wray, T. Glauert, and A. Hopper. The Medusa applications environment. *Proceedings of the International Conference on Multimedia Computing and Systems*, May 1994. Boston MA.

[75] C.J. Wu and J.S. Huang. Human face profile recognition by computer. *Pattern Recognition*, 23(3/4):255–259, 1990.

[76] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using Hidden Markov Model. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 379–385, 1992.

[77] R.K. Yin. Face recognition by brain-injured patients: a dissociable ability? *Neuropsychologia*, 8:395–402, 1970.

[78] A.W. Young and V. Bruce. Perceptual categories and the computation of "grandmother". In V. Bruce, editor, *Face Recognition*, pages 5–49. Laurence Erlbaum Associates, 1991.

[79] S.J. Young. The HTK Hidden Markov Model Toolkit: Design and philosophy. Technical Report TR.153, Department of Engineering, Cambridge University, UK, 1993.

[80] A.L. Yuille, P.W. Hallinan, and D.S. Cohen. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8(2):99–111, 1992.