

# Learning to Identify Faces in Images and Video Sequences

A thesis submitted to the University of Manchester

for the degree of Doctor of Philosophy

in the Faculty of Medicine, Dentistry and Nursing

1999

Gareth J. Edwards

Division of Imaging Science and Biomedical Engineering

# Contents

<b>1</b>	<b>Introduction</b>	<b>18</b>
1.1	Interpreting face images . . . . .	18
1.2	Motivation . . . . .	19
1.3	Approach . . . . .	20
1.4	Outline of Thesis . . . . .	22
<b>2</b>	<b>Machine Vision for Face Interpretation</b>	<b>24</b>
2.1	Location . . . . .	24
2.2	Identification . . . . .	25
2.3	Further interpretation . . . . .	25
2.4	Tracking . . . . .	26
2.5	Synthesis . . . . .	26
2.6	Types of approach . . . . .	27
2.7	Model-based methods . . . . .	28
2.7.1	Properties of models . . . . .	29
2.8	Shape-based methods . . . . .	30
2.8.1	Active Contours . . . . .	30
2.8.2	Deformable Shape Models . . . . .	32

2.8.3	Active Shape Models . . . . .	33
2.9	2D appearance-based models . . . . .	34
2.10	Combined shape and texture models . . . . .	37
2.11	3D models . . . . .	38
2.12	Anatomical models . . . . .	39
2.13	Discussion of model-based approaches . . . . .	40
2.14	Measuring face interpretation performance . . . . .	42
2.14.1	Recognition tasks . . . . .	43
2.14.2	Measuring performance . . . . .	44
2.14.3	The FERET programme . . . . .	47
2.14.4	Advanced interpretation . . . . .	48
2.15	Summary . . . . .	49
<b>3</b>	<b>Shape and grey-level Appearance Models</b>	<b>51</b>
3.1	Modelling shapes . . . . .	52
3.1.1	Labelling the training shapes . . . . .	52
3.1.2	Aligning the training shapes . . . . .	53
3.1.3	Principal Component Analysis of training set . . . . .	54
3.2	Searching images for plausible shapes . . . . .	56
3.2.1	Matching local grey-level models . . . . .	57
3.3	Modelling shape-free texture . . . . .	61
3.4	Interpreting faces using ASMs . . . . .	63
3.4.1	Classification . . . . .	64
3.4.2	Identification using shape and texture . . . . .	65

3.5	Discussion of the ASM-based approach . . . . .	67
3.6	Summary . . . . .	68
<b>4</b>	<b>Appearance Models</b>	<b>70</b>
4.1	Motivation . . . . .	70
4.2	Formulation . . . . .	71
4.2.1	Choice of shape parameter weights . . . . .	75
4.3	Example of a face model . . . . .	75
4.3.1	Visualisation . . . . .	76
4.3.2	Fitting the model by hand . . . . .	78
4.3.3	Limitations of the reconstruction method . . . . .	79
4.3.4	Specificity . . . . .	79
4.4	Summary . . . . .	80
<b>5</b>	<b>Partitioned Models</b>	<b>83</b>
5.1	Motivation . . . . .	83
5.1.1	Interpretation . . . . .	84
5.1.2	Synthesis . . . . .	84
5.1.3	Tracking . . . . .	85
5.1.4	Model building . . . . .	85
5.2	Modelling subspaces . . . . .	86
5.3	Linear Discriminant Analysis . . . . .	87
5.3.1	Formulation . . . . .	88
5.4	Residual subspaces . . . . .	90
5.5	Identity model using LDA . . . . .	92

5.5.1	Non-identity model . . . . .	93
5.5.2	Projecting images onto subspaces . . . . .	93
5.6	Expression model . . . . .	95
5.6.1	Projection onto expression subspaces . . . . .	97
5.7	Face manipulation . . . . .	98
5.7.1	Retaining the integrity of fine texture . . . . .	102
5.8	Alternatives to LDA . . . . .	105
5.9	Summary . . . . .	107
<b>6</b>	<b>Active Appearance Models</b>	<b>108</b>
6.1	Motivation . . . . .	108
6.2	Background . . . . .	110
6.2.1	Global optimisation . . . . .	110
6.2.2	Directed optimisation . . . . .	110
6.2.3	Related work . . . . .	111
6.3	Active Appearance Model search . . . . .	112
6.3.1	Overview of AAM search . . . . .	112
6.3.2	Learning to correct the model parameters . . . . .	113
6.3.3	Regression results for the face model . . . . .	115
6.3.4	Iterative model refinement . . . . .	116
6.4	Examples . . . . .	116
6.5	AAM search versus hand-fitting . . . . .	117
6.6	Comparison with ASM-based recognition . . . . .	118
6.7	Summary . . . . .	120

<b>7</b>	<b>Tracking Faces</b>	<b>121</b>
7.1	Simple tracking using ASMs . . . . .	121
7.2	Kalman filtering . . . . .	122
7.2.1	Basic theory . . . . .	123
7.2.2	Example model . . . . .	124
7.2.3	Kalman update procedure . . . . .	126
7.3	Filtered ASM tracking . . . . .	127
7.3.1	Dynamic models . . . . .	129
7.3.2	Discussion . . . . .	129
7.4	Tracking using a Partitioned AAM . . . . .	130
7.4.1	Motivation . . . . .	131
7.4.2	Overview . . . . .	131
7.4.3	Tracking translation, scale and orientation . . . . .	132
7.4.4	Tracking the model parameters . . . . .	134
7.5	Limitations of decoupled AAM tracking . . . . .	136
7.6	Dynamically updating the partitioned model . . . . .	136
7.6.1	Motivation . . . . .	137
7.6.2	Formulation . . . . .	139
7.6.3	An adaptive tracking scheme . . . . .	140
7.7	Initial evaluation . . . . .	141
7.7.1	Stability of identity measurement . . . . .	142
7.7.2	Reconstruction error . . . . .	145
7.7.3	Linear relationship between parameters . . . . .	148

7.8	Summary . . . . .	149
<b>8</b>	<b>Interpreting Sequences</b>	<b>150</b>
8.1	Interpretation by tracking . . . . .	150
8.2	Experimental framework . . . . .	151
8.2.1	The interpretation task . . . . .	151
8.2.2	Test data . . . . .	152
8.3	Static-Static recognition . . . . .	154
8.4	Dynamic-Static recognition . . . . .	156
8.5	Dynamic-Dynamic recognition . . . . .	158
8.6	Discussion of results . . . . .	160
8.7	Summary . . . . .	161
<b>9</b>	<b>Extensions and Future Work</b>	<b>166</b>
9.1	General applicability of AAMs. . . . .	166
9.2	Automatic landmarking . . . . .	168
9.3	Extending models to colour . . . . .	171
9.4	Recognising expression . . . . .	172
9.5	Extending the representation . . . . .	175
9.6	A half-face model . . . . .	176
9.6.1	Detecting faces . . . . .	178
9.6.2	Dealing with occlusion . . . . .	179
9.6.3	Efficiency of AAMs . . . . .	179
9.6.4	Dynamic models . . . . .	180
9.7	Summary . . . . .	180

<b>10 Conclusions</b>	<b>182</b>
10.1 AAMs in machine vision . . . . .	182
10.1.1 Appearance Models . . . . .	183
10.1.2 Active Appearance Models . . . . .	183
10.1.3 Partitioned Models . . . . .	184
10.1.4 Interpreting sequences . . . . .	185
10.2 Final statement . . . . .	186
<b>A Warping Face Images</b>	<b>187</b>
A.1 Image warping . . . . .	187
A.1.1 Piece-wise affine warping . . . . .	188
<b>B The Training Images</b>	<b>192</b>



# List of Figures

2.1	Typical scheme for model-based image interpretation. . . . .	29
2.2	Typical example of a ‘snake’ attracted to edges in a face image. Initial position shown on left, final solution on right. . . . .	31
2.3	Example of ROC curves. ‘Chance’ curve shown as straight line. Increasing performance as curves move towards top-left. . . . .	46
3.1	Face images with 122 key landmark points placed by hand annotation.	53
3.2	Effect of varying each of first three face shape parameters between $\pm 3$ s.d. . . . .	57
3.3	Grey-Level sample patches aligned along normals to curve. . . . .	58
3.4	ASM Search. At each model point a better location is sought by searching along the normal at the current location. . . . .	59
3.5	Locating a face using the Active Shape Model search algorithm. . . .	60
3.6	Example faces with extracted ‘shape-free’ patches. . . . .	62
3.7	First three modes of variation of a typical shape-free face model. . . .	63
3.8	Illustration of the effect of training-class variability. Unknown example is more likely to be Ann than Brian, even though is lies closer to the centroid of Brian. . . . .	65
3.9	Examples from the three image sets used by Lanitis to evaluate ASM-based recognition. . . . .	66
4.1	Selection of typical face images from the training set. . . . .	76

4.2	The effect of varying the first three parameters of the appearance model between $\pm 3$ s.d's. . . . .	77
4.3	Reconstruction of images from training set and unseen images. . . . .	78
4.4	Reconstruction using unified fitting method. Left - original image, Centre - fitting with shape-dominated scheme, Right - unified fitting scheme. . . . .	80
4.5	A selection of random faces generated by the model. . . . .	82
5.1	Linear Discriminant Analysis in two dimensions. Examples from each class are shown scattered in 2D - each is also shown projected onto the single discriminant axis. This projection yields the optimum group separation. . . . .	90
5.2	Effect of varying the first 3 parameters of the 'identity' subspace model built using LDA. . . . .	92
5.3	Effect of varying the first 3 parameters of the 'non-identity' subspace model built by analysis of data after 'projecting-out' identity variation. . . . .	93
5.4	Original images projected onto identity and non-identity subspaces respectively. . . . .	94
5.5	Training examples marked with expression labels. . . . .	96
5.6	Effect of varying the first 3 parameters of the 'expression' subspace model built using LDA. . . . .	97
5.7	Effect of varying the first 3 parameters of the 'non-expression' subspace model built using LDA. . . . .	98
5.8	Original images projected onto expression and non-expression subspaces respectively. . . . .	99
5.9	Schematic diagram of face manipulation method. . . . .	101
5.10	Images lose texture as they are reconstructed using fewer model parameters. . . . .	102
5.11	Manipulating expression <i>without</i> retaining image texture. . . . .	104
5.12	Manipulating expression whilst retaining fine texture. . . . .	104

5.13	Bias caused by poor training data. . . . .	105
6.1	Overview of AAM search scheme. . . . .	113
6.2	Examples of AAM search. Original image on left. Iterations 1,2,5 shown on right. . . . .	117
6.3	Typical search performance. RMS value of grey-level error per pixel is shown as a function of iteration number. Image grey-levels are in the range 0-255. . . . .	118
7.1	Illustration of Lanitis' simple tracking scheme. . . . .	122
7.2	Example of a 1-dimensional integrated random-walk. . . . .	125
7.3	Schematic diagram of Kalman filter algorithm. . . . .	127
7.4	Schematic diagram of decoupled and filtered tracking algorithm. . . .	132
7.5	First 2 identity and non-identity parameters for a typical sequence. .	137
7.6	Limitation of Linear Discriminant Analysis: Best identification possible for single example, Z, is the projection, A. But if Z is an individual who behaves like X or Y, the optimum projections should be C or B respectively. . . . .	138
7.7	Schematic diagram of full, refined tracking algorithm. . . . .	142
7.8	Typical values of first 6 identity parameters versus frame number, using simple tracking scheme. . . . .	143
7.9	Typical values of first 6 identity parameters versus frame number, using decoupled tracking scheme. . . . .	143
7.10	Typical values of first 6 identity parameters versus frame number, using full, adaptive tracking scheme. . . . .	144
7.11	Reconstruction error during tracking using simple, unfiltered scheme.	146
7.12	Average percentage difference in reconstruction error (compared with the simple tracking scheme) for the simply decoupled and adaptive tracking schemes. . . . .	147

7.13	Average value of R-squared statistic for each identity parameter, indicating a strong linear relationship between the identity and non-identity parameters. . . . .	148
8.1	Some examples frames from sequences of individuals in the ‘probe’ set.	154
8.2	Some examples frames from sequences of individuals in the ‘gallery’ set.	155
8.3	Distance between probe sequence and gallery image is calculated by projecting the image in the same way as the sequence. . . . .	157
8.4	ROC curves for verification system. Dynamic-static system compared with static-static system. . . . .	158
8.5	ROC curves for verification system. Dynamic-dynamic system compared with static-static system. . . . .	159
9.1	Example images of MRI brain slices with landmarks overlayed. . . . .	167
9.2	First 3 modes of variation of brain model. . . . .	168
9.3	AAM search applied to a previously unseen brain image. . . . .	169
9.4	Detail of training image - in this particular case, the landmarks (circles) are badly placed whilst the result of AAM search (triangles) is closer to the desired position. . . . .	170
9.5	First 3 modes of variation of a colour face model. . . . .	173
9.6	Typical examples of face images used to evaluate expression recognition performance. . . . .	174
9.7	Examples of face images used to build a half face model. . . . .	177
9.8	First three modes of variation of half face model. . . . .	178
A.1	Delaunay triangulation applied to the mean shape of the face PDM. . .	189
A.2	Using piece-wise affine warping can lead to kinks in straight lines. . .	190

# List of Tables

3.1	Classification results reported by Lanitis [61]. . . . .	67
6.1	Classification results using Active Appearance Model versus Active Shape Model. . . . .	119
7.1	Measure of ‘stability’ of identity estimates for 3 tracking methods compared with the average percentage difference in reconstruction error. .	145
8.1	Average distance between gallery and probe images using <i>static-static</i> recognition scheme. . . . .	163
8.2	Average distance between gallery and probe images using <i>dynamic-static</i> recognition scheme. . . . .	164
8.3	Average distance between gallery and probe images using <i>dynamic-dynamic</i> recognition scheme. . . . .	165

# Abstract

We present novel methods for locating, tracking and interpreting faces in images and video sequences using 2D Appearance Models of faces. We describe how to construct models that represent both the shape and texture variation in faces and can be used to generate photo-realistic synthetic reconstructions of new faces. We describe how Appearance Models can be combined with an active search algorithm. We show how these Active Appearance Models (AAMs) can be efficiently fitted to image data. AAMs provide the basis for many types of analysis, including face identification and expression recognition. We show how Appearance Models can be partitioned into subspaces describing different types of ‘real-world’ variation such as identity, pose, lighting and expression. This partitioning provides the basis for an adaptive tracking scheme that exploits the fact that identity must remain constant during a sequence. The scheme provides on-line refinement of the subspaces during tracking and improves the stability of measurements of identity. All the methods have been systematically tested using still images and video sequences. We show that AAMs provide an effective means of interpreting faces in images and video and that the adaptive tracking scheme results in improved face recognition compared with equivalent static methods.

# Declaration

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

# Copyright

1. Copyright in text of this thesis rests with the Author. Copies (by any process) either in full, or of extracts, may be made **only** in accordance with instructions given by the Author and lodged in the John Rylands University Library of Manchester. Details may be obtained from the Librarian. This page must form part of any such copies made. Further copies (by any process) of copies made in accordance with such instructions may not be made without permission (in writing) of the Author.
2. The ownership of any intellectual property rights which may be described in this thesis is vested in the University of Manchester, subject to any prior agreement to the contrary, and may not be made available for use by third parties without the written permission of the University, which will prescribe the terms and conditions of any such agreement.

Further information on the conditions under which disclosures and exploitation may take place is available from the Head of the Division of Imaging Science and Biomedical Engineering.



# Acknowledgements

I would like to thank my supervisor, Professor Chris Taylor for his dedicated supervision and insightful input to the research over the past three and half years. Thanks also for agreeing to fund several pleasurable trips to international conferences.

Other members of the Wolfson Unit particularly important to this work are the rest of the ‘faces group’ - Tim Cootes, Nick Costen and Kevin Walker. Further helpful advice and code fragments were provided by Andreas Lanitis, David Cooper, Alan Brett, Paul Smyth, Jim Graham, Stuart Solloway and Neil Thacker.

Computing resources were provided and maintained by Warren Mittoo, David Burke and Jermaine Gilmour. Additional administrative support was provided by Christine Cummings, Angela Castledine and Shelagh Stedman.

Patrick Courtney and Visual Automation Ltd. are to be thanked for their considerable and successful efforts to take this research to the outside world.

I would like to thank Mum and Dad, the rest of the family and most of all Francesca for their continuous support and encouragement.

My research was jointly funded by British Telecom PLC and EPSRC.

# Chapter 1

## Introduction

### 1.1 Interpreting face images

This thesis presents research into automatic interpretation of video (and still) images of human faces. The interpretation of face images can be broken down into a number of distinct tasks. These include:

- face detection (where is it?)
- face identification (who is it?)
- expression recognition (what is their expression?)
- pose recovery (where are they looking?)

In this thesis we adopt an approach that addresses these, and other tasks, as examples of the same general problem - *understanding images*.

Most applications of machine vision require a system to understand images in some way, recovering some or all of the structure of the world represented by the image, and explaining the meaning of this structure. We present a number of techniques

designed to explain the appearance of faces in still and video images. We build upon existing methods that explain face images in terms of either their shape, or their grey-level appearance. The aim of the approach is to devise algorithms that make optimal use of all the available information. We describe a method that combines both shape and grey-level description. This generic approach to the image understanding problem leads to novel methods of image synthesis and manipulation. Further, we seek to make use of the fact that a video sequence contains more information than a static image, and present methods that attempt to make optimal use of the dynamic information present in a sequence. We also demonstrate the wider applicability of these algorithms in other areas of machine vision.

## 1.2 Motivation

*Face recognition* has become one of the most active areas of research in computer vision. There are two main reasons: the commercial potential offered by practical recognition systems, and the challenges that face images provide as a test of machine vision algorithms. In this project we are motivated by both.

The term ‘face recognition’ implies finding the identity of faces in images. This project goes beyond identity recognition and attempts a more detailed understanding of face images and video sequences, including, for example, expression recognition. However, reliable face identification is an important aim of this project. The list below gives a number of potential applications of reliable face identification technology.

- *Access control*
- *Surveillance*
- *Secure transactions*
- *Human-computer interaction*

- *Database retrieval*

## 1.3 Approach

A successful interpretation system should be able to locate and track faces in images, interpreting specific properties of the face regardless of confounding factors. For example, the system should be able to identify a face regardless of variation in pose, expression and lighting. This makes the analysis of face images a difficult machine vision task. As a result of this difficulty, many researchers have concentrated on particular constrained applications, contributing little to overall progress. Others have attempted to tackle the various generic problems (location, identification, and expression recognition) independently. The difficulty with this approach is that the effects of all the sources of variability in face images are compounded, so it is extremely difficult to extract a descriptor for one characteristic of interest (e.g. identity) without taking account of the others (e.g. facial expression, lighting and pose).

Rather than separating face analysis into separate tasks, such as feature location, person identification, expression recognition, lighting normalisation, etc., we have developed a unified approach. The basis for this is a compact, parameterised model of facial appearance that accounts for all the important, systematic sources of variability. Our approach consists of modelling - in which flexible appearance models of facial appearance are generated - and interpretation - in which the models are used to analyse the information content of the face image, such as the expression or identity of the individual.

There are many existing model-based approaches to face interpretation, some of which will be discussed in Chapter 2. The most closely related work, and indeed the precursor to our current research, is the appearance model approach of Lanitis *et al* [67] [66]. This approach uses statistical models of shape, local grey-level appearance, and global grey-level appearance. By combining the shape model with the local grey-

level models, an *Active Shape Model (ASM)* can be created and used for face location [64]. The located face is then interpreted using the parameters of all three models.

The work described in this thesis extends Lanitis' approach by encapsulating all the information about face variation within a single model. We show how this full *Appearance Model* provides a more compact and specific representation of face images. The model captures the features that are common to all faces and contains a description of how their appearance is allowed to vary over a large range of face images.

From this unified description, we show how specific types of variation, such as identity, expression, pose and lighting can be separated and modelled individually. This separation of sources of variation is the basis for a novel approach to tracking; the system is able to exploit known dynamic constraints, such as the fact that an individual's identity must remain fixed over a sequence. Moreover, the parameterised representation allows synthesis of photo-realistic faces and manipulation of their characteristics such as pose and expression.

The unified model provides a complete and specific description of face images, and is used to locate and interpret faces in images. This involves solving the difficult optimisation problem of matching the model automatically to new images. As we will show, a typical model may contain more than 80 parameters; matching methods based on standard optimisation algorithms fail because of this high-dimensionality and the preponderance of local minima. Active Shape Model search [27] provides a partial solution to the problem. Using just the shape and local grey-level information, faces can be located and their shape recovered. Given the shape, it is relatively straightforward to find the 'best-fit' of the full Appearance Model. The shortcoming of this approach is that it does not fully exploit knowledge of *combined* shape and grey-level appearance during search. We describe a novel solution to the problem using *Active Appearance Model Search*, which completely unifies the face location and interpretation tasks.

Although the main aim of this thesis is to present a unified framework for face image understanding, none of the algorithms presented are particular to face images. By adopting a generic, model-based approach, we present methods that are potentially useful in many model-based image interpretation applications.

## 1.4 Outline of Thesis

**Chapter 2** reviews current approaches to automatic face interpretation. We describe several approaches, concentrating on those most closely related to our own.

**Chapter 3** reviews Active Shape Models (ASMs) and their use in face interpretation. We also describe the use of grey-level models in combination with ASMs. Previous results obtained using this approach are presented along with a discussion of the strengths and weaknesses of the method.

**Chapter 4** introduces a combined *Appearance Model*, which encapsulates both shape and texture, discussing the motivation for such a model, and the details of its formulation. We describe a specific appearance model, built to describe faces, and illustrate some of its properties.

**Chapter 5** discusses the motivation and describes a method for partitioning the full appearance model into separate subspaces for pose, expression, lighting and identity. In particular, we focus on a model that isolates identity variation; this is used later in the tracking system. In illustrating the partitioned model we show how it can be used for facial synthesis and manipulation.

**Chapter 6** describes the *Active Appearance Model* algorithm. We explain the novel optimisation method behind the technique, and discuss the properties of the approach. We introduce a model trained to interpret face images and present experimental results.

**Chapter 7** describes our approach to face tracking in video sequences. Active Appearance Model search is combined with a partitioned model, allowing us to impose strong dynamic constraints on the tracking system.

**Chapter 8** shows how the novel tracking method can be used to enhance the interpretation of video sequences. We show how evidence can be integrated over time to give better estimates of identity. We evaluate the system and present test results for a set of video sequences.

**Chapter 9** presents several recent extensions to the work described in this thesis. We discuss these, along with directions for future research.

**Chapter 10** contains a general discussion of the work presented in this thesis.

## Chapter 2

# Machine Vision for Face Interpretation

In this chapter we review existing approaches to the automatic interpretation of face images. We concentrate on methods closely related to those used in our research, focusing primarily on model-based approaches. Face location and face interpretation are often treated as separate problems. Some researchers, including ourselves, prefer a unified approach in which similar techniques are used for both location and identification; we pay particular attention to such work. First, we discuss some of the tasks a face interpretation system might be expected to perform.

### 2.1 Location

A useful face recognition system must have at least some degree of autonomy in locating faces in images. For example, in security applications, one of the main attractions of face recognition is the potential of passive, perhaps even covert, person identification systems. Whilst systems based on the user placing his or her head in a fixed position are conceivable, it is hard to see what benefits this confers over other



active approaches, such as fingerprint [73] [53] or iris recognition [33] [84], or even the humble keypad.

We can define *face detection* as a subtask of face location. This comprises locating any region of the image that contains a face - including the possible detection of multiple faces. We define *face location* to include more detailed description of the location of features within the face region. There is some overlap between this definition of location and *interpretation*, moreover, reliable interpretation is impossible without accurate location. In this thesis we present methods which combine the location and interpretation tasks.

## 2.2 Identification

The primary interpretation task of most systems is to identify the located face(s). There are two main types of task; given a face image, choose a match from a list of possible candidates, or, given a face image, decide whether or not it is a close enough match to a pre-defined candidate. These two tasks are often referred to as *identification* and *verification*. For each task, the same information needs to be extracted from the image, although the classification methods differ.

## 2.3 Further interpretation

Whilst most systems focus on identification, there is increasing interest in other forms of interpretation. Automatic interpretation of expression may prove to be an important part of improved human-computer interaction. Detailed understanding of human emotion almost certainly requires video rather than static images, nevertheless some systems, including ours, display a degree of success with still images.

Whilst perhaps not strictly *face* interpretation, machine vision has been applied to

lip-reading, and in particular, combining lip-reading with speech recognition systems [69] [14]. A complete face interpretation system ought to be able to perform lip-reading as a sub-task. Other sub-tasks of face interpretation include gaze estimation for machine interfaces [80] and blink detection for monitoring driver fatigue [91].

A successful system must be able to deal with many sources of face variation. For example, even if we are not particularly interested in the pose of a face, an identification system must still be able to understand the difference between image variation caused by pose change, and that caused by differences between individuals' appearance.

## 2.4 Tracking

In practical applications the source of face images will often be a video camera. Ideally, we would analyse as much of the video stream as possible. Not only does it inherently make sense to use as much evidence as possible, it may be essential in certain cases to monitor the activity of a person over time. *Tracking* usually implies using knowledge of previous locations to help find the current location - even if just to provide a starting estimate - simply running a global face location algorithm on each frame of a sequence is not strictly tracking. Most current approaches treat face tracking and face interpretation separately - the interpretation is usually performed on static images extracted from the sequence. In this thesis we present a system in which the tracking and interpretation algorithms are unified.

## 2.5 Synthesis

Given that a good system must be able to 'understand' the variation present in face images, it is reasonable to expect that given some parameters, the system should be able to recreate an instance of a face image. This is a characteristic of *generative*

*models* - models that are sufficiently complete to be able to generate realistic synthetic images. In this thesis we describe several such models.

A system that understands not only the totality of variation in faces, but also understands the sources of variation, such as expression, pose, and lighting should be able to manipulate synthetic faces. Later in the thesis we will present examples of synthetic reconstructions in which specific characteristics of a face are altered.

## 2.6 Types of approach

In most face-interpretation scenarios, the position of the face in an image is not accurately known, thus a system must perform *face location* before interpretation is possible. Many researchers treat face location as distinct from the interpretation task. We believe that such approaches are fundamentally flawed; given the detailed knowledge required for face interpretation, it seems natural, that once obtained, this knowledge should be used for face detection.

The variability of faces from one image to the next makes reliable interpretation difficult. One approach to this problem is to measure properties of the images that are as invariant as possible such as edges. An alternative to reducing variability is to build prior models of the variability. We review several model-based approaches in this chapter.

An alternative to *whole face* interpretation is to concentrate on particular features such as eyes or mouths. The motivation for this is the reduced amount of variability in a small feature compared with a whole face; this is also the downside - there is less information to constrain the interpretation task.

This thesis presents 2D view-based models of faces. The computational complexity and storage requirements of 2D algorithms are likely to be smaller. There is an extremely large amount of 2D training data available in the form of face images; real

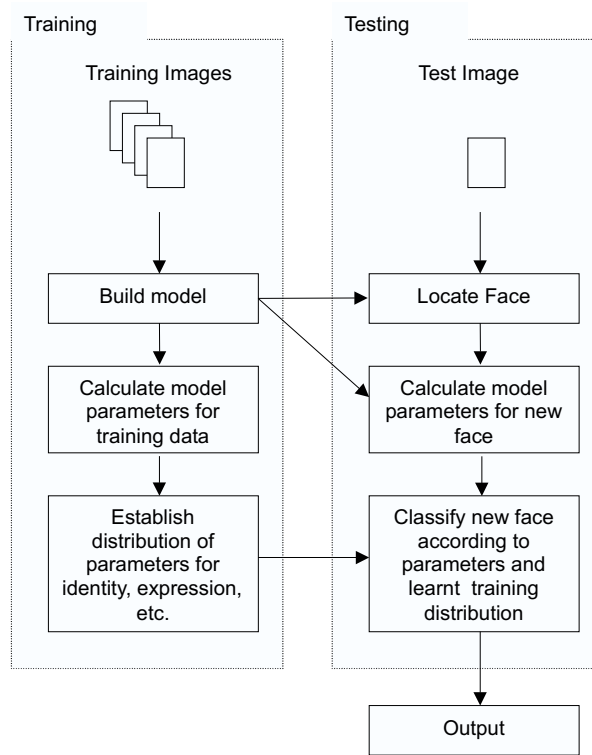
3D data is much more elusive, requiring special equipment. The practical application of 3D-based systems is limited by the difficulty of making 3D measurements in real applications, where the installation of specialist equipment may not be feasible (or affordable). Indeed, we would like to take advantage of existing cameras in such places as shopping centres, cash machines, etc. The human visual system provides a compelling existence proof that 3D analysis is not required for successful interpretation - we can perform the task perfectly well on photographs and films. Despite these reservations about the 3D approach, we provide a brief review. In particular, we are interested in the use of 3D training data for analytically addressing the problem of 2D appearance change due to pose and lighting.

## 2.7 Model-based methods

In this section we outline a variety of model-based methods presented in recent literature. Some are techniques specifically intended for the interpretation of face images, whilst others are more general methods in computer vision which have been applied to face images. It is notable that the latter are usually more successful.

Model-based methods always involve a training stage, where the model itself is configured. This may range from the simple definition of some ad-hoc constraints to full statistical learning methods. For problems such as face interpretation - where a large degree of variability is involved - the aim of training is usually to produce a *parameterised* model. Given an image, the interpretation task is to find the optimal set of model parameters that best ‘fit’ the image data in some sense. These parameters usually become the input to a classifier or other interpretation mechanism. A typical model-based scheme is illustrated in Figure 2.1

Model-based methods address the need in non-trivial applications to ‘understand’ face images. Given some set of image measurements, a model provides a frame of reference in which to interpret those measurements. Model-based methods also



**Figure 2.1:** Typical scheme for model-based image interpretation.

provide the means to deal with extremely complex and variable structures and with noisy and incomplete image data. It is difficult to conceive of a system capable of accurately measuring the positions of various facial features without some prior knowledge of facial structure and variability.

### 2.7.1 Properties of models

A useful model must fulfil two main criteria: *generality* and *specificity*. General models are those that account for all possible sources of appearance variation in the class of objects of interest (in this case faces), and can thus represent any example of the class. Specific models constrain the allowable variability so that only ‘legal’ examples can be generated. Specific models provide powerful image interpretation constraints - the expected shapes of structures, their spatial relationships and their grey-level (or colour) appearance can be used to restrict an automated system to

plausible interpretations.

We are particularly interested in generative models, that is models which are capable of reconstructing realistic images of faces. Such models allow a straightforward statement of the interpretation problem; given an image, adjust the parameters of the model in such a way as to generate a synthetic image as close as possible to the original. This statement only holds if the model is specific - it must not be capable of ‘explaining’ image regions that do not contain faces. At this point we note that the requirement of specificity is much more difficult to attain than that of generality. A completely general image model is the trivial null-model, i.e. no constraints on the image. In this case any image pixel can take any value, and thus produce any image.

The following sections review existing model-based approaches to face interpretation. Some involve models of whole faces, others are feature-based. In each case we discuss the specificity and generality of the approach.

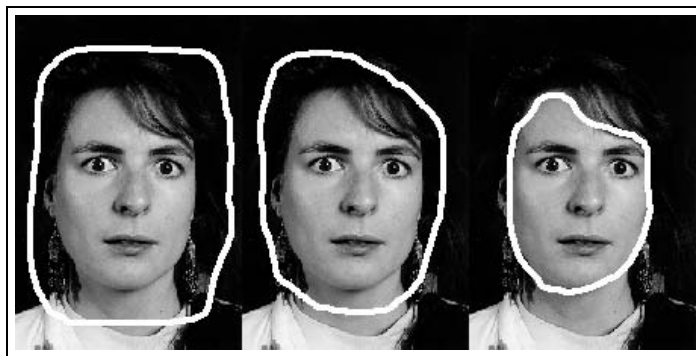
## 2.8 Shape-based methods

Many researchers attempt to locate the outline of the whole face or of individual facial features in face images. Some methods are purely data-driven such as the ‘snake’ of Kass *et al* [57], whilst others rely on prior models, taking advantage of the constrained geometrical relationships between the positions of facial features. These shape models are usually characterised by sets of key points, organised into contours.

### 2.8.1 Active Contours

A popular method of contour detection is the data-driven method of Kass *et al* [57], often referred to in computer vision literature as ‘Active Contour Models’ or more commonly ‘Snakes’. A Snake is a contour, usually represented as a spline curve, which can be placed in an image, and is then ‘attracted’ to image features. For example,

if one wished to locate the boundary of an object, the snake could be configured to be attracted to edges in the image. The snake's movement in the image is not completely unconstrained; the algorithm attempts to minimise an overall 'energy' function which incorporates image data, elasticity and smoothness. Whilst seeking suitable image features, the snake favours configurations in which its elastic energy is minimised (i.e. the snake contracts as much as possible) and in which the snake is as smooth as possible. Kass *et al* [57] used snakes to track the boundaries of lips in face images, showing reasonable results in constrained situations. Waite and Welsh [94] applied the same technique to location of complete head boundaries. Figure 2.2 illustrates the typical behaviour of an edge-attracted snake.



**Figure 2.2:** Typical example of a 'snake' attracted to edges in a face image. Initial position shown on left, final solution on right.

Only the image-data, elasticity, and smoothness constraints affect the solution found by a snake - there is no prior knowledge of the expected configuration. This makes the algorithm broadly applicable, but is also its drawback. For example, an edge-based snake is attracted to *any* edge, regardless of whether it belongs to the object of interest; snakes are thus particularly bad at dealing with background clutter in a scene. The snake can take any energy-minimising configuration, regardless of whether the solution represents a plausible shape. Because the snake itself has no prior knowledge of the scene, it is usually necessary to provide knowledge at run-time - by placing the snake initially close to the desired solution. The behaviour of a snake is highly dependent on the weights given to the data, elasticity and smoothness terms

in the energy function. Practical image interpretation systems may require difficult manual tuning of these parameters.

Despite the drawbacks of the method, some researchers are still attracted to the data-driven nature of the solution - it requires virtually no training data. Okubo *et al* [76] describe the application of snakes for lip-tracking; their success is possible because of the highly constrained conditions of their application. Recently, Yokoyama *et al* [95] describe a snake-based method for locating facial contours, adding an extra energy term for deviation from symmetry. This appears to offer some improvement over completely unconstrained snakes. The problem with this sort of approach is the *ad-hoc* nature of the constraints: how do you set the weight of the symmetry term? Too high and non-symmetric face images will be missed, too low and the constraint offers nothing. This approach might work reasonably well for rigid objects, but it is difficult to guess in advance the allowable variability of complex flexible objects such as faces.

## 2.8.2 Deformable Shape Models

Yokoyama's symmetry-snake approach uses an extra, fairly weak constraint to improve the robustness of image search. As more constraints are introduced, the approach begins to become more *model-based*, incorporating detailed prior knowledge of shapes. A common approach is the use of 'hand-crafted' geometric templates such as in the work of Yuille *et al* [97]. Yuille's models are used to locate eyes and mouths in images. A flexible model of eye shapes is built from primitive geometric shapes such as circles and ellipses. The template is allowed to vary by varying parameters such as scale, rotation, circle radius, etc. In total, Yuille uses 11 parameters. The search procedure is similar to that of snakes - the template is updated to minimise an image cost function. However, rather than allowing the contours to deform arbitrarily, updates are performed on the model parameters. The parameter values are constrained to restrict the solution to 'plausible' shapes. Brunelli and Poggio [16]



introduce a larger model of the complete face, using the constrained geometrical relationships between facial features. This model performs a step-wise search, where an initial feature is located and then the model constraints are used to limit the search space for other features. Yow and Cipolla [96] describe the use of a probabilistic belief network based on grouping hand-crafted models of facial parts such as eyes, nose and mouth.

An alternative to handcrafting shape variability is to base deformations on the physical properties of objects such as stiffness and elasticity. Pentland [77] describes the use of Finite Element Analysis for generating flexible deformations of templates. Similar work has been presented by Terzopoulos and Metaxas [89] and Nastar and Ayache [74]. Terzopoulos and Waters [90] combined Finite-Element based deformations with hand-crafted anatomical constraints.

The difficulty of these approaches is the arbitrary definition of both the model and ‘plausible’ variation. For example, using an ellipse to detect the eye region can be neither general nor specific: No eyes are exact ellipses, and lots of things are elliptical but are not eyes. A model must be handcrafted to achieve the best trade-off. Physically based deformations offer no real solution - the generation of the model parameters is automatic, but is no more likely to be specific than hand-crafted parameters. This difficulty is partly addressed by Craw *et al* [32], who attempt to derive the variability constraints from a large set of training shapes, using a model similar to that of Brunelli and Poggio [16].

### 2.8.3 Active Shape Models

The problem of building general and specific models of object shape is addressed by the *Point Distribution Models (PDMs)* of Cootes *et al* [27]. They model the shapes of variable objects via a statistical analysis of *landmark points* located on training images [12] [35]. New shapes belonging to the general class can be reconstructed/parameterised using a weighted sum of basis functions derived during the

analysis. *Active Shape Models*(ASMs) combine the constrained variability of a PDM with a search method driven by the image data. Lanitis *et al* [63] [62] describe the use of ASMs for both modelling shape variation of faces and for locating facial features in images. Active Shape Models are covered in more detail in Chapter 3.

Active Shape Models use a linear formulation to encapsulate shape variation. Similar approaches have been attempted using non-linear formulations. Bregler *et al* [14] describe a number of local linear shape models to produce a global non-linear model of lips. Edwards *et al* [36]\* describe the extension of PDMs to non-linear shape models using a Multi-Layer Perceptron. When using a non-linear model, it was found that the shape variability of the training images could be explained with as few as half the number of parameters, indicating that the representation was more specific. More recently, Cootes *et al* [21] and Heap and Hogg [49] have described methods which combine multiple linear models of variation to produce a non-linear representation.

Of the contour-based approaches discussed, only ASMs offer a plausible means of achieving both generality and specificity. This comes from the fact that both the mean shape of the model and allowed variability associated with human faces are derived through a statistical analysis of the training set, rather than by generating arbitrary shape variations and imposing arbitrary constraints.

## 2.9 2D appearance-based models

An alternative to modelling shape is to model grey-level appearance. Kirby and Sirovich [58] first proposed a Principal Component Analysis, or Karhunen-Loeve Decomposition of face images. The analysis is performed on a set of roughly aligned training images. Exploiting the fact that there is correlation between pixel values across the training set, they seek a set of basis images that represent the train-

---

\*These experiments were conducted primarily by Andreas Lanitis

ing data as compactly as possible. Each training image can be reconstructed as a weighted sum of basis images. Although this type of analysis can be applied to many types of images, face analysis is the most well-known application - the approach is often referred to as the *eigenface* method - the space in which the basis images lie referred to as the *eigenspace*.

Turk and Pentland [92] first used the eigenface method to design an automatic face identification system; the decomposition weights of a particular face are used for recognition. The formulation of the eigenface decomposition also provides a fast method of performing correlation-based matching. Moghaddam and Pentland [71] also showed how the representation can be used to model and locate individual facial features, such as eyes and mouths.

The strength of the eigenface approach is its use of statistical training; it does not rely on *ad-hoc* models or detailed anatomical knowledge. Unfortunately, as a model-based approach, it suffers from poor specificity. Other than an initial rough alignment of the images, there is no explicit correspondence between pixels across the training set, for example, in a set of face images the positions of the eyes will vary considerably. A specific model must simultaneously represent both shape and intensity changes across the training set; the eigenface approach does not achieve this. As a result, the basis images can be combined to produce illegal examples of faces. Craw *et al* [31] attempted to remedy this problem by first normalising the training images by warping a set of hand-placed control points to an average shape before performing the decomposition. This important step ensures that the eigenface decomposition reflects only the variation in grey-level appearance and not variation in shape. Warping to a reference shape greatly reduces the within-class variation - i.e. the variation between images of the same individual, thus making person identification easier. Lanitis *et al* [67] adopt a similar approach, which forms a starting point for the work in this thesis. The method is described in detail in Chapter 3.

There are other alternatives and extensions to the Karhunen-Loeve expansion. Akamatsu *et al* [2] propose a method using a Karhunen-Loeve decomposition of the power spectrum after applying a Fourier transform to the basis images. They reason that the power spectra may vary in a more linear fashion than the raw images, though there is no obvious reason why this should be so. Cottrel and Fleming [29] train an eigenface-type model using a multi-layer perceptron. Such a non-linear approach could simultaneously represent shape and texture changes using a single set of model parameters. However, with such a high-dimensional input, and large degree of variability, training this model effectively is prohibitively difficult. Belhumeur *et al* [7] begin with a Karhunen-Loeve expansion before performing a further statistical analysis on the expansion coefficients of the training set. By using Linear Discriminant Analysis (LDA), a set of basis functions are generated which best describe the differences between the identity of individuals at the expense of other variation, such as expression and lighting. A similar analysis was performed by Zhao *et al* [98], testing several combinations of Principal Component Analysis and Linear Discriminant Analysis; like Belhumeur *et al* [7] they report that PCA followed by LDA improves classification performance.

Moghaddam *et al* [72] model two mutually exclusive classes of variation: intra-personal (pose, expression, lighting, etc.) and extra-personal (difference between individuals). They use a Bayesian classifier for recognition; the likelihood for each class of variation is learned from the training data by performing density estimation in the eigenspace. They note that in the eigenface representation, intra-personal differences are swamped by extra-personal differences. This is to be expected in a representation that is not shape-normalised.

Gong *et al* [45] use a Gabor Wavelet Transform (GWT) to build a representation for classifying head pose. By using only the magnitude of the frequency responses, the representation is less sensitive to image plane translations.

## 2.10 Combined shape and texture models

Clearly, face images are characterised by variation in shape and grey-level (or colour) texture<sup>†</sup>. A truly general and specific model of appearance must account for both factors. The method of Lanitis *et al* [67] uses a combination of Active Shape Models and shape-normalised texture decomposition. Faces are described and reconstructed using a combination of the shape and texture models; however, since there are no constraints on the models' combination, it is possible to produce implausible shape and texture combinations. A similar representation is described by Vetter [93] and used to generate synthetic views of faces from different viewing angles to that of the input image.

Cootes and Taylor [22] and Nastar *et al* [75] attempt to build a unified model of the grey-level surface by combining the point co-ordinates at key points with the grey-level intensity at those points. Unfortunately, no effective method of matching these models to image data exists.

Lades *et al* [60] describe a combined shape and intensity face model of a different form. During training, they overlay a rectangular grid on a training image from each individual in the database. They measure the responses at each of the grid points for a set of two-dimensional Gabor filters tuned to different orientations and scales. This provides a model-based description of the shape and texture of a given face. When a new image is presented to the system, the grid is overlaid and allowed to deform. A similarity measure between the new image and each training image is computed, based on the responses of the same set of Gabor filters and the grid distortion.

Jones and Poggio [55] describe a combined shape and texture model built from 100 prototype images. In this approach, an image is represented as a sum of warped prototypes. A matching scheme is described but it is both slow ( 1 minute per image)

---

<sup>†</sup>Unless stated otherwise, in this thesis we use the term 'texture' to describe the grey-level appearance remaining after shape normalisation.

and unreliable, working only when accurately hand-initialised. Rikert and Jones [80] have used such models to train a neural-network to estimate gaze direction based on extracted model parameters.

This thesis describes the use of a new type of combined shape and texture model, which can be viewed as an extension of the approach of Lanitis *et al* [67], and is related to the approach of Jones and Poggio [55]. This method, described by Edwards *et al.* [38], begins with a Point Distribution Model and uses this same set of points to build the shape-normalised texture model. By learning about the correlation between the parameters of the two models, a single unified model is built which is both general and specific. The model is also generative - capable of synthesising realistic images of faces. This approach is explained in detail in Chapters 3 and 4.

## 2.11 3D models

Some researchers have attempted to use 3D models for interpretation. The obvious attraction is the prospect of accurate analysis and reconstruction of faces under any viewing angle. Further analytical methods such as ray tracing can be used to deal with lighting variation. The drawback of 3D models is the large increase in complexity and storage, as well as the difficulty of obtaining suitable training data.

DeCarlo and Metaxas [34] describe a 3D mesh representing the surface of a face. The mesh deformations are controlled by a small set of parameters that are hand-crafted, but based on observed face anthropometry. The model is used to track faces in images, using an optical-flow based algorithm. The system must be initialised by accurately fitting the model by hand to the first frame of the sequence. Clark and Kokeur [18] and Li *et al* [68] use variations of a 3D model known as CANDICE [82], a 3D wire frame model derived using a triangulation algorithm. Again, these models are used to track faces, but require very good initialisation.

A common use of 3D models is to interpret pose from 2D images. A example is the work by Shakunaga *et al* [85] who use a standard 3D head model to back-project located 2D features, thus obtaining an estimate of pose. This single, fixed model does not represent the different 3D face shapes of individuals and is thus prone to error - although the system can be calibrated for a particular individual. It is not clear, however, that using such a model offers any benefit over raw, data driven calibration from the 2D features; the model provides only *fixed* geometric projections of the image data. This 2D-3D mapping might only be achieved with proper knowledge of 3D variability. Shimizu *et al.* [86] use a large number<sup>‡</sup> of training images along with 3D range data captured at the same time. This data is used to build a generic model of 2D-3D variability. The 2D data is represented as a number of edges extracted from the original images, thus providing some degree of normalisation against texture variation; at the expense, of course, of the texture information itself. They propose a model-matching scheme based on closest-curve matching.

## 2.12 Anatomical models

Anatomical models are motivated by the need for generality and specificity, acknowledging the fact that appropriate variability constraints are unlikely to be found in an *ad-hoc* manner or by using physics-based deformations such as vibrational modes. In particular, the finite and fixed (assuming healthy individuals) number of facial muscles place constraints on the allowable deformations. However, this type of modelling does not lead to reliable estimates of the variability between different individuals, nor does it automatically deal with pose and lighting variation. Anatomical Models of the face, incorporating tissue and muscles have been described by Terzopoulos and Waters [90] and Essa and Pentland [43] [42]. Aoki and Hashimoto [3] describe a highly detailed physical model based on data obtained from 3D CT scans of heads. In this approach, the model consists of three layers, skull, muscle and skin, as in real

---

<sup>‡</sup>The actual number of training examples is unspecified.

faces. Motion is synthesised by spring-like deformations of muscle and skin, together with rigid movements of the jawbone. The computational expense of this approach is high, and while useful for synthesis of facial expressions, the model is not used for interpretation. Like the 3D models discussed above, anatomical models are useful for tracking a *particular* face once initialised, but are not good for generalising to new individuals.

## 2.13 Discussion of model-based approaches

In the previous sections we have outlined some recent model-based approaches to face interpretation. The papers cited represent a small selection from the extensive literature in the field, but were chosen to illustrate all of the major approaches.

Generally, the more information a model uses, the higher will be its dimensionality (the number of parameters required to control it), and the more difficult it becomes to learn about the allowable variability (and thus achieve specificity). The aim of good modelling is the representation of all the image information in the lowest possible dimensionality. ASMs are an example of a compromise approach using limited image information (typically the shape of boundaries). This naturally compromises the information content of the model, but makes generality and specificity realistic objectives. Strictly though, the fact that the ASM only uses boundary information limits its specificity - as long as the boundary region makes sense, other areas are ignored. Cootes *et al* [27] point out the benefit of this, that ASM are robust to occlusion, since typical occlusions might only cross a few boundary points and thus have little effect. Although a useful property in many circumstances, ASMs do not deal with occlusion directly, it just happens that their lack of specificity with regard to large internal regions allows the model to fit when most of the boundary points are visible.

The eigenface approach is an attempt to model the grey-level information in



face images. The linear schemes proposed by Kirby and Sirovich [58] and Turk and Pentland [92] fall short of the requirement of specificity. However, the method does provide some reduction in dimensionality and provides a fast alternative to correlation-based matching schemes. We do not favour the direct non-linear extension of eigenface methods - why invent complex algorithms when the main source of non-linearity (lack of correspondence) can be easily removed? Both Craw *et al* [31] and Lanitis *et al* [67] have demonstrated that a shape-normalised representation displays superior specificity. Until recently, the application of such models has been limited by the lack of an algorithm for rapid image interpretation.

The most promising approaches involve the use of both shape and texture information. Indeed, these summarise all the useful information available in the image (including the possibility of coloured texture.) The reasoning is straightforward - only by interpreting shape variation correctly can we build a specific model of texture, and only by combining shape and texture variation can we generalise to new faces with different shapes and textures. Such a model must also understand the inevitable correlation between shape and texture, in order to retain specificity.

The use of 3D models and physical models may offer alternative methods for learning about variability in face images. However, Lanitis *et al* [67] showed that, at least for limited pose angles, view-based learning from training images is sufficient. In Chapter 9 we show how view-based learning can deal with full pose range from frontal to profile. The added complexity of 3D and anatomical models makes them difficult to train and configure; as yet there are no successful face interpretation algorithms which use such models. We regard it in the same way as we regard the use of colour images: humans do not require colour images, 3D images, or anatomical knowledge of facial muscles in order to interpret face images. Moreover, most useful applications of face recognition would currently have to take their input from 2D monochrome cameras. Anatomical and 3D models may, however, provide a means of approximating change in 2D appearance due to pose and lighting in the absence of suitable training images.

In addition to the approaches we have discussed here, the Face Recognition literature contains many papers describing rather arbitrary schemes for specific interpretation tasks. Most of these can be regarded as one-off engineering attempts that contribute very little to the overall understanding of the face interpretation problem. It is hoped that the work presented in this thesis is of a more generic nature.

## 2.14 Measuring face interpretation performance

Often the evidence presented concerning the effectiveness of a given computer vision technique consists of a few successful images printed in a scientific paper<sup>§</sup>. Face interpretation is an area of research characterised by lack of comparative evidence. In most cases, this is not so much the result of bad scientific practice, but rather due to the lack of standardisation across task definitions and test data. The range of problems addressed is large - it is hard, for instance, to compare the merits of a head-pose estimation algorithm with those of an expression classifier.

Many research groups use their own test data to evaluate algorithms - this is often a necessity, due to the lack of publicly available data. It would be in the interest of the research community to ensure that local data is made publicly available for other groups to use in algorithmic testing.

In an ideal world, algorithms themselves would be available for public evaluation<sup>¶</sup>. This is hindered by the lack of any common framework for the interchange of code. Within the C++ community, there exists a common environment for machine-vision programming - the Image Understanding Environment (IUE). It remains to be seen whether it will become extensively used.

---

<sup>§</sup>Indeed, in many cases, even this is enough to dismiss a given method!

<sup>¶</sup>Many of the algorithms described in this thesis are available in cross-platform MATLAB code through the web-site of the Wolfson Image Analysis Unit - <http://www.wiau.man.ac.uk/>

### 2.14.1 Recognition tasks

The only area in which there exists enough data to compare face interpretation algorithms is *identity recognition*. Even this can encompass a range of tasks and test conditions. There are two main criteria that characterise the recognition phase:

1. Interpretation requirement - identification or verification?
2. Input constraints - is the head position fixed/given in advance?

In addition to these criteria we could add a list of other constraints concerning image capture, such as lighting, camera calibration, etc. Most current algorithms are tested on static images, although the source of the static image may well be a frame taken from a video camera.

The first type of interpretation, *identification* involves comparing the unseen input image against a database of known individuals and labelling the identity of the unseen image. A suitably advanced system should also recognise when a face is not part of the database and perform an appropriate action. *Verification* is a slightly different task; in this case, the system is told in advance which person to expect, and it must return a yes/no verification for the input image. In most literature, and in this thesis, the phrase ‘Face Recognition’ is often used to refer to either of these tasks.

Some systems work only when given the position of the face in the image. These rely on the assumption that either the user will be constrained or that future development will yield a reliable location algorithm. More advanced systems can perform location, either given a reasonable initial approximation, or completely unprompted. Many systems use completely separate technologies for location and interpretation tasks - the leading model-based techniques tend to use the same technology for each.

Almost every paper concerning face recognition contains some measure of recognition/verification rate. These can only be assessed in the context of the particular

task the experimenters set for themselves. In order to directly compare recognition algorithms they must be tested on the same task. Section 2.14.3 describes a recent attempt to provide a standard test framework.

### 2.14.2 Measuring performance

Most performance measuring algorithms involve splitting a set of images into a *training set* and *test set*. It is essential that the test set plays no part in the training procedure. In face recognition experiments this is a necessary but not sufficient condition that must be satisfied in order to avoid bias. Unfortunately, there often exists other bias in the data - for example, all the images of a particular person, X, might be captured against the same background. Without careful thought, one cannot be sure whether the algorithm is recognising person X or simply the background. Where it is not possible to eliminate bias in the image set, experimental protocol must account for it.

In full recognition experiments the commonly used performance measure is simply the *recognition rate* - the percentage of the inputs correctly identified. Care must be taken when interpreting this statistic; how many possible answers are there? If there are only two people in the test then 50% recognition is the same as chance. In tests on larger databases, *ranked recognition* is often used. In this case, the algorithm will return, say, the 3 closest matches in the database. The reasoning is that security systems would probably be happy to allow passage if the individual was ranked 3rd out of a possibility of thousands. Lanitis *et al* [67] performed this type of recognition experiment on a database of images gathered in-house. The full set of results is given in Table 3.1.

Another measure of performance is the Receiver Operating Characteristic curve (ROC). An ROC curve can be used to predict a system's expected performance on a variety of tasks. Given a test image and a database of  $N$  images, we ask the question, how far is the test image from each database image? In this case we allow

the situation where the test image has no match in the database. Since almost all algorithms return some normalised scalar measure of ‘distance’ between images, classification performance will depend on the choice of ‘matching’ threshold,  $T$ . Let  $d_{dt}$  be the distance between a pair of images,  $I_d$  and  $I_t$  (a database and a test image). We define the identification *decision rule* as:

$$\text{if } d_{dt} < T \quad \text{person is the same} \quad (2.1)$$

$$d_{dt} > T \quad \text{person is not the same} \quad (2.2)$$

Given a test image,  $I_t$ , the decision rule is evaluated for each of the  $N$  database images,  $I_d$ . This will result in a certain number of accepted matches,  $n_a$ , and rejections,  $n_r$ . Clearly, the following relationships are true:

$$n_a + n_r = N \quad (2.3)$$

$$n_a = \begin{cases} 0 & \text{if } T = 0, \\ N & \text{if } T = \infty \end{cases} \quad (2.4)$$

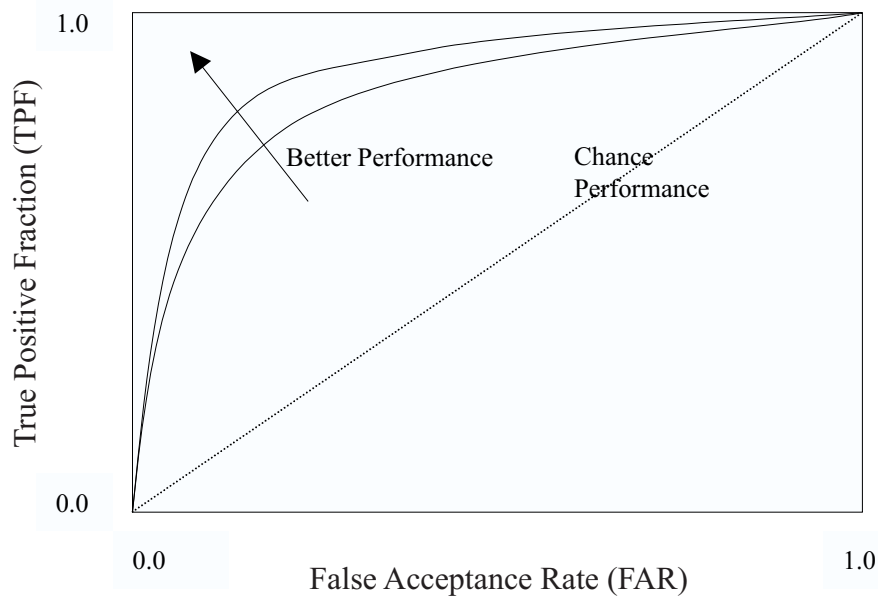
This states the obvious result, that given an threshold of zero, all potential matches will be rejected, whereas with an infinite threshold, all potential matches will be accepted.

The average number of accepted matches and rejections can be calculated over all the images in the test set, thus giving a means of evaluating an algorithm’s performance. As the threshold level,  $T$ , is varied the ratio of  $n_a$  to  $n_r$  will change. At any particular threshold level, we can calculate how many of the test images are *correctly* matched against training images. This figure is known as the *True Positive Fraction (TPF)*. As the threshold becomes very large the TPF will approach a value of unity. We can also calculate the ratio of the number of matches that were returned incorrectly, over the total number of database images. This is often called the *False Alarm Rate (FAR)*. As the threshold becomes large the FAR will also approach unity.

An ROC curve plots the value of TPF against FAR for varying threshold level,

*T.* Some example of ROC curves are shown in Figure 2.3. The key points are these:

- the diagonal line from bottom-left to top-right is the performance to be expected by chance alone;
- curves which pass close to the top-left of the plot generally indicate better performance



**Figure 2.3:** Example of ROC curves. ‘Chance’ curve shown as straight line. Increasing performance as curves move towards top-left.

Whilst a useful measure of a systems performance, the ROC curve itself does not provide a single, definitive statistic indicating a systems usefulness. In some systems the actual cost of a false acceptance might be greater than in others. The ROC curve allows a choice of where the threshold ought to be set for a given level of performance. The ROC approach uses a suitably normalised measure of distance to reject images that do not match any of the database images. However, in the situation where it is known that *all* test images occur somewhere in the database, nearest-neighbour

matching schemes, where a different threshold is effectively used for each image, may give better performance than indicated by the ROC curve.

### 2.14.3 The FERET programme

The Face Recognition Technology (FERET) programme [78] was initiated in September 1993 by the U.S. Department of Defense, Defense Advanced Research Projects Agency (DARPA) and the U.S. Army Research Laboratory. The major aim of the programme was the collection of a large database of face images, and subsequent testing and evaluation of leading face recognition systems. The first test phase took place in August 1994. In this test, the image database was split into a training set (known as the *gallery*) of 316 individuals and test set (known as *probes*). The main test assessed the verification rate of the algorithms on the probe images. In March 1995 the test was extended, with a gallery of 817 individuals. This second phase introduced *duplicate* images in the probe set; these are images of the same person taken on a different date. The significance of duplicate probe images is that the image capture conditions were quite different. By September 1996 the database had been extended to a gallery of 3323 images and probe set of 3816 images. The FERET test procedure requires each entrant to supply a distance measure for each of the probe/gallery pairs. From these figures, the test administrators calculate ROC performance characteristics. Details of the test procedure and latest results can be found in Rizvi *et al* [81].

The 1996 test compared 10 algorithms from 7 different institutions. There was found an enormous difference in performance between the normal probes (captured on the same day) and the duplicate probes (captured on a different day). For the normal probes, given a 10% FAR, the TPF varied from 0.95 to 0.995 across the algorithms. For the duplicate probes, the range was 0.58 to 0.80.

Naturally, most researchers and commercial companies are keen to quote the results achieved on the normal probes. Far more revealing, however, is the huge drop in

performance observed for the duplicate images. To relate this to real numbers, imagine a security system based on the *best* of the algorithms. If the system registered 1000 individuals, it could be set to allow 800 people to correctly enter the building, with the proviso that 100 people would get past the gate anyway, incorrectly verified. Obviously there is significant variability between images of individuals taken at different sittings; the algorithms tested under FERET do not deal well with this variation.

The FERET test data is not publicly available. Groups must apply to take part in the programme, which involves testing under the supervision of a FERET administrator, who personally brings the data to the test site. Unfortunately, we are not yet part of the FERET programme, though we are keen to take part in the planned next phase. The normal probe set is not particularly interesting since it does not provide the sort of realistic data a working system would have to deal with. We regard any claims made by third parties based on the normal probe images with scepticism.

#### **2.14.4 Advanced interpretation**

The system we present in this thesis can be used for a range of interpretation tasks. We present results of both identity and expression recognition experiments. A major part of the work is the extension of the algorithms to perform interpretation of video sequences. This involves making optimal use of dynamic information as well as individual frames. There are currently no standard test databases of such video data - instead we have produced an internal database of training and test sequences captured in sessions separated by a 5-month period. It is intended that this data be made publicly available (see Appendix B) in order that other researchers can compare their results with ours.



## 2.15 Summary

In this chapter we have reviewed some of the leading approaches to face interpretation. The majority of algorithms are concerned primarily with face identification. We have concentrated mainly on model-based techniques, particularly those related to the methods presented in this thesis.

The key requirements of a successful model are generality and specificity. Several models exist which are specific within their own frame of reference. The best example is the Active Shape Model (ASM) approach. ASMs are only capable of generating plausible face shapes and are thus specific in one respect. However, by not using all the image information they remain capable of fitting to image regions that are not faces, but simply satisfy some of the shape requirements of a face. ASMs are therefore not completely specific. Ideally a model should use all the available image information and be generative, that is, capable of reconstructing a synthetic example of a face, only in this way can we be assured of specificity.

A popular approach to using grey-level image information is the eigenface method. Unfortunately the simple linear analysis of training images is not sufficient to build specific models. The main problem is the lack of pixel correspondence - eigenfaces attempt to explain both shape and texture variation with a single texture model. Lanitis' shape-free region models first account for shape variation by warping all the images to common shape, thus producing a more specific representation of texture variation. The shape-free region models can be combined with Active Shape Models to account for both shape and texture variation. The drawback of this approach is that correlation between shape and texture is not accounted for. This thesis presents a new method of combining shape and texture variation.

Despite the large number of researchers engaged in automatic face interpretation, there exist few satisfactory methods of comparing the performance of algorithms. The FERET programme has attempted to provide a unified test framework for face

recognition/verification. The results obtained by the 7 research groups involved in the latest phase of the FERET programme are generally poor; identification proves very difficult on images taken under different conditions. All the algorithms perform extremely well on the images captured on the same day - the suspicion is that this test is biased. We believe that understanding the variation between images of the same individual is the key to accurate recognition.

We describe later a recognition scheme that can use video sequences as well as static images. The dynamic information may help the system understand the variation present in images of the same person. Unfortunately there is no standard test data on which to test such algorithms. The experiments presented in this thesis were performed on especially collected test sequences. We took care to capture the sequences with a 5 month break between sittings, in different conditions. This allows an experiment similar to the *duplicate* problem in the static FERET test, on which existing algorithms performed badly. It is hoped that other research groups will test their algorithms on our video database.

## Chapter 3

# Shape and grey-level Appearance Models

The work described in this thesis builds on existing methods developed in the Wolfson Image Analysis Unit. This chapter describes *Point Distribution Models* (PDMs) of object shape as introduced by Cootes *et al* [23], followed by an overview of *Active Shape Models* (ASMs) [27] which use PDMs in image search. ASMs are a good example of the effective use of prior models in computer vision. They offer a robust solution to the interpretation of shapes in images, and are based on models learnt from training sets of example objects. ASMs have been used successfully in many applications, from medical image analysis [25] [24] [41] [59] [87] [88] to industrial inspection [52]. It is the ability of the ASM approach to adapt to a wide range of tasks which makes them suitable for interpreting images of humans, which are typically extremely variable. ASMs have been used successfully to track whole individuals in scenes [5], and to interpret hand gestures [1]. In this chapter we will describe primarily the work of Lanitis *et al* [66] [67] [65] who describe the use of ASMs for the interpretation of face images.

## 3.1 Modelling shapes

An Active Shape Model is intended to locate and interpret a particular class of shapes in images. In most non-trivial applications, the shapes of interest will exhibit a range of variability. A good model of object shape will encapsulate this variability, but not allow variability that produces non-legal shapes, that is shapes which are not valid examples of the chosen class of objects. Faces are a good example; there are many possible configurations for the outlines of the lips, but none in which the lips appear above the nose. For any complex object, the only practical way of establishing allowable variation is by learning from a set of examples. ASMs contain a statistical model of shape variability, learnt by analysing a training set. This model is known as a Point Distribution Model, or PDM. The following sections outline the construction of a PDM.

### 3.1.1 Labelling the training shapes

The first step in building a PDM is to annotate (usually manually) the structures of interest in each of a set of training images. This involves defining a set of ‘landmark’ points, corresponding to specific image features. Each training shape can be represented by a vector  $\mathbf{x}$ :

$$\mathbf{x} = (x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n)^T \quad (3.1)$$

where  $(x_i, y_i)$  is the position of the  $i^{th}$  landmark point.

The landmarks are usually placed on the boundaries of structures of interest and other points that appear consistently in different images of the same class of objects. It is important that landmarks are placed consistently throughout the set of training images, in order to achieve a standardised representation. To achieve this, certain landmarks are placed at easily identifiable positions, such as the corners of the eyes

and lips, whilst others are evenly distributed along the boundaries between these. In our experiments with shape models, we use 122 landmark points to define the shape of a face. Figure 3.1 shows examples of face images annotated with their landmark points.



**Figure 3.1:** Face images with 122 key landmark points placed by hand annotation.

### 3.1.2 Aligning the training shapes

The aim of the PDM is to capture the variation in shape across a class of objects. In most applications of PDMs there is no desire to model variation due to translation, in-plane rotation or change of scale - indeed, it is necessary to ensure that any such variation is, as far as possible, removed from the model. In order to achieve this, the training shapes are aligned before training begins so as to minimise the total-squared distance between the landmarks and their mean positions over the whole training set.

This is achieved using a Generalised Procrustes Analysis method [46]. Each shape is aligned with the average shape using a weighted least-squares method. The weights are chosen to give more significance to the points that tend to be most ‘stable’ over the training set. Further details of the alignment procedure are given by Cootes *et al* [27].

### 3.1.3 Principal Component Analysis of training set

The Procrustes Analysis results in a set of aligned training shape vectors,  $\mathbf{x}_i$ . The dimensionality,  $2n$ , of  $\mathbf{x}$  is typically larger than the number of independent ways in which the shapes can vary. This is because the points do not move independently of each other. For example, the movement of two nearby points at the tip of the chin will be very highly correlated. The PDM uses a parameterised representation of the shape variation that captures this correlation between points and can thus represent the variation present in the training set by a much smaller number of parameters than  $2n$ . This is achieved by performing Principal Component Analysis [70], as follows:

The mean training shape,  $\bar{\mathbf{x}}$ , is given by:

$$\bar{\mathbf{x}} = \frac{1}{m} \sum_{i=1}^m \mathbf{x}_i \quad (3.2)$$

Each shape’s deviation from the mean,  $\bar{\mathbf{x}}$ , is given by  $\delta\mathbf{x}_i$ :

$$\delta\mathbf{x}_i = \mathbf{x}_i - \bar{\mathbf{x}} \quad (3.3)$$

The  $2n \times 2n$  covariance matrix,  $\mathbf{S}$ , is then calculated:

$$\mathbf{S} = \sum_{i=1}^m \delta \mathbf{x}_i \delta \mathbf{x}_i^T \quad (3.4)$$

By calculating the eigenvectors,  $\mathbf{p}_k$  ( $k = 1, \dots, 2n$ ), of the covariance matrix, the cloud of shape examples in the  $2n$ -dimensional space can be represented by a set of mutually orthogonal axes defined by the eigenvectors of  $\mathbf{S}$ .

$$\mathbf{S} \mathbf{p}_k = \lambda_k \mathbf{p}_k \quad (3.5)$$

where  $\lambda_k$  is the  $k^{th}$  eigenvalue of  $\mathbf{S}$ ,  $\lambda_k \geq \lambda_{k+1}$ .

The largest eigenvalues correspond to the axes that describe the most significant modes of variation of the shapes. Most of the variation can usually be described by a relatively small number,  $t$  ( $< 2n$ ), of these axes.

Any shape,  $\mathbf{x}$ , in the training set can then be approximated by a weighted sum of the first  $t$  eigenvectors and the mean shape,

$$\mathbf{x} \approx \bar{\mathbf{x}} + \mathbf{P} \mathbf{b} \quad (3.6)$$

where  $\mathbf{P} = (\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_t)$  is the matrix of the first  $t$  eigenvectors, and  $\mathbf{b}$  is a vector of weights, normally referred to as *shape parameters*.

The value of  $t$  is usually chosen so that the sum of the variances of the first  $t$  modes, describes a given proportion of the total variance,  $\lambda_T$ , where,

$$\lambda_T = \sum_{k=1}^{2n} \lambda_k \quad (3.7)$$

The number of modes in the model can also be chosen in such a way as to ensure the model is able to reconstruct its training examples with a given level of accuracy.

By varying the values of the shape parameters,  $\mathbf{b}$ , in equation 3.6, representations of new examples can be constructed. It is important to define limits on the range of values that the shape parameters can take. Assuming a uni-modal distribution of each parameter,  $b_k$ , each one is chosen to be within the limits,

$$-r\sqrt{\lambda_k} \leq b_k \leq r\sqrt{\lambda_k} \quad (3.8)$$

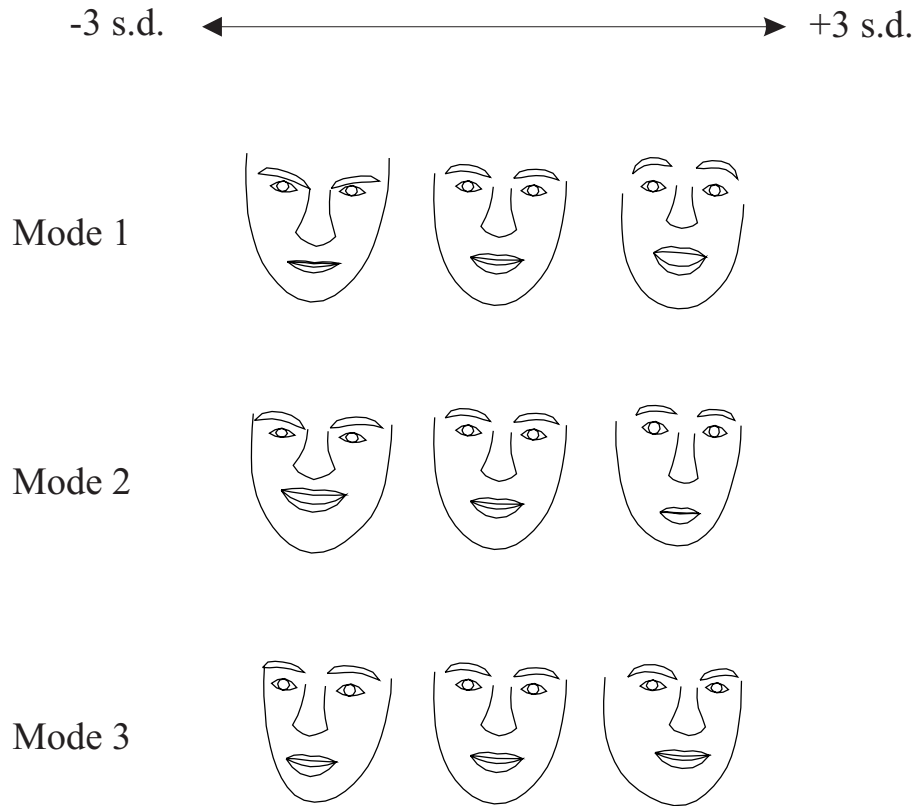
where  $r$  is chosen such that any shapes generated are plausible instances of the class of objects to be represented.

Each element of the vector  $\mathbf{b}$  controls a mode of variation of the shape model. If we vary a single element of  $\mathbf{b}$  and fix all the others, equation 3.6 can be used to reconstruct a set of example shapes corresponding to the variation encapsulated by that element of the shape vector. Figure 3.2 illustrates the three most significant modes of variation for a PDM of the human face - that is the effect of varying the first three model parameters. The model shown was built from a training set of 768 images and is represented by 30 eigenvectors. These eigenvectors represent 98% of the variance observed in the set of training examples.

## 3.2 Searching images for plausible shapes

A PDM is specific, that is, capable of generating only ‘legal’ shapes. This property is critical for robust image interpretation, in which we seek image shapes that can be represented by the model. Although there exist various methods of searching for plausible shapes (see for example, Hill and Taylor [51]), the most successful algorithm is the Active Shape Model (ASM). A detailed description of this method can be found





**Figure 3.2:** Effect of varying each of first three face shape parameters between  $\pm 3$  s.d.

in Cootes *et al* [26]; here we give a brief overview.

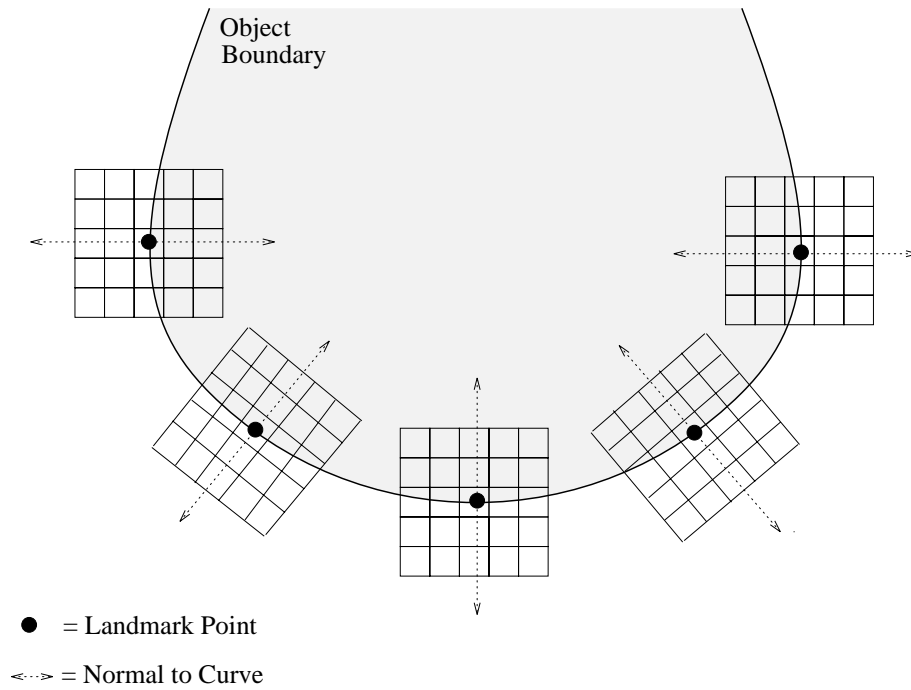
### 3.2.1 Matching local grey-level models

The approach taken in the Active Shape Model algorithm is to combine PDM model constraints with a local search for each of the landmark points. In order to do this, factor models [54] are built of the local grey-level appearance around each point [26]. Each model is usually of a region aligned normally to the curve on which the landmark point lies. This is illustrated in figure 3.3.\*. A statistically defined ‘fit’ function allows the similarity between a local model and any image patch of the same dimensions to be assessed. Fitting the factor model to an image patch, yields a set of local model

---

\*This illustration was kindly provided by Dr. Stuart Solloway

parameters - these are sometimes used for further analysis, such as recognition.

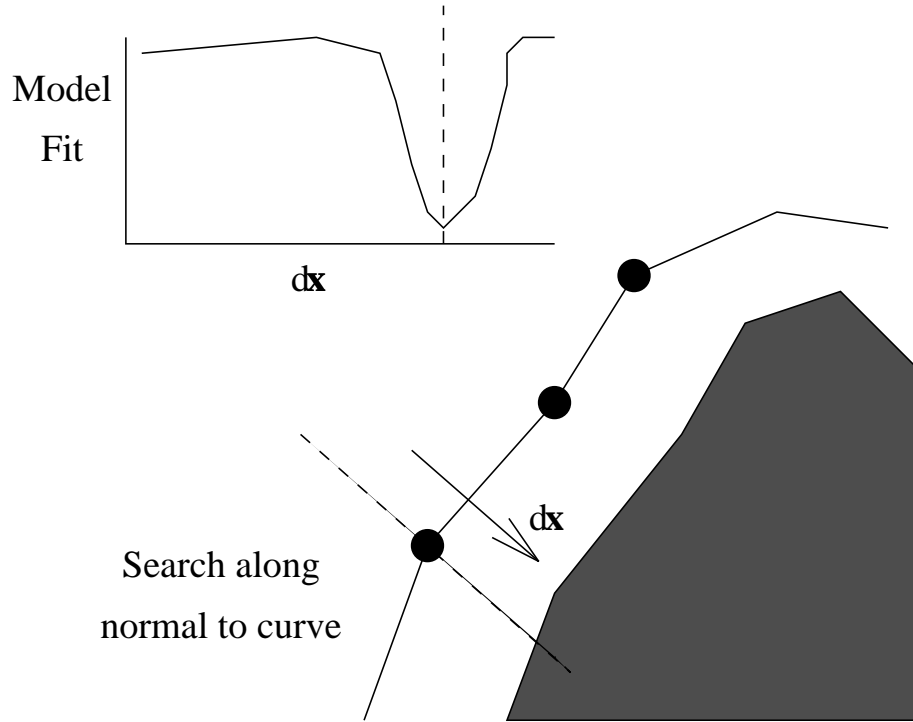


**Figure 3.3:** Grey-Level sample patches aligned along normals to curve.

The ASM algorithm combines search for matches to local grey-level models with the global shape constraints provided by the PDM. The algorithm is similar to the ‘snakes’ of Kass *et al* [56] in that image data is used to ‘attract’ control points; the crucial difference is the application of global *a priori* shape constraints, in fact ASMs are sometimes referred to as ‘smart snakes’.

To begin ASM search, an instance of a PDM is initialised in an image. The aim is to then iteratively refine the PDM. This is achieved by searching the image around the current location of each landmark point, seeking better matches for its local grey-level model. Usually the search is along normals to the curve on which the point lies. This is illustrated in figure 3.4

This results in a set of suggested adjustments to the landmark points, given by a



**Figure 3.4:** ASM Search. At each model point a better location is sought by searching along the normal at the current location.

displacement vector,  $d\mathbf{x}$ :

$$d\mathbf{x} = (dx_1, dx_2, \dots, dx_n, dy_1, dy_2, \dots, dy_n) \quad (3.9)$$

Given the vector of required adjustments,  $d\mathbf{x}$ , the model is updated in two stages:

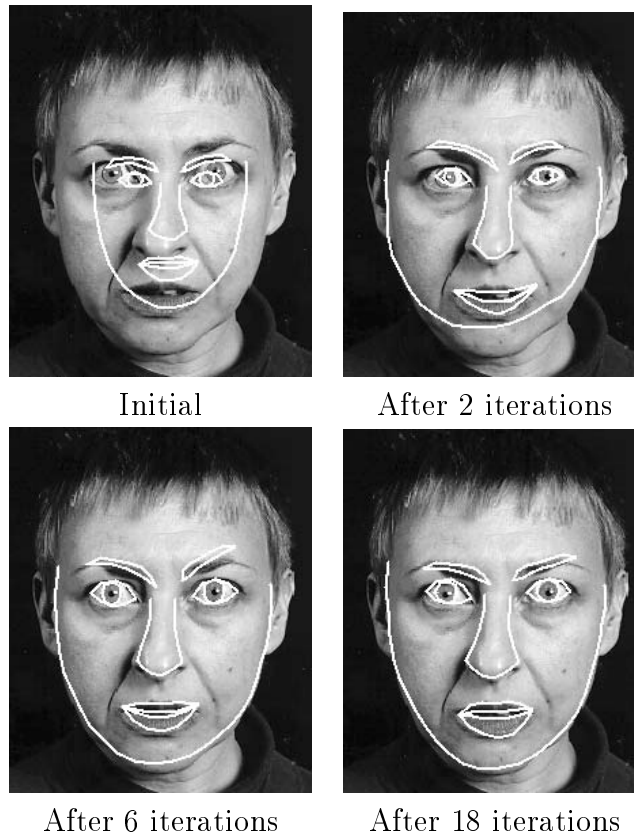
1. the pose, scale, and orientation of the model are updated.
2. the shape parameters,  $\mathbf{b}$  are updated.

The pose, scale and orientation of the model are adjusted in such a way as to make the points move as near as possible to their desired locations in a least-squares sense. The remaining differences between the model points and their desired locations are known as the residual displacements,  $d\mathbf{x}'$ . The residual displacements are reduced by updating the shape parameters of the model. The model parameters,  $\mathbf{b}$ , are allowed to vary within limits learnt from the training set. Cootes *et al* [27] show that by applying

a least squares approach, the optimum adjustments,  $d\mathbf{b}$ , to the shape parameters,  $\mathbf{b}$ , are given by:

$$d\mathbf{b} = \mathbf{P}^T d\mathbf{x}' \quad (3.10)$$

By ensuring that the model points are only moved by changing the model parameters within limits learnt during training, the new shape will always represent a legal example. The search procedure is repeated until further iterations do not result in any change in the model. At this point, the search is said to have converged. Figure 3.5 shows an example of ASM search.



**Figure 3.5:** Locating a face using the Active Shape Model search algorithm.

### 3.3 Modelling shape-free texture

Modelling shape alone only encapsulates a limited amount of information. Models of the grey-level appearance attempt to capture information from all the pixels in the region of the image containing the face. A well-known grey-level modelling technique is the ‘eigenface’ method used by Turk and Pentland [92]. In the eigenface approach, a set of training images containing faces are represented as vectors of pixels, and Principal Component Analysis is performed, yielding a low-dimensional representation of the image data.

The main drawback of the eigenface method is the lack of pixel correspondence across the training set. Even if the training images are normalised for the position, scale and in-plane rotation of the face, the natural variability in face shape means that corresponding facial features occur at different image locations. This occurs in images of the same person (due to factors such as pose change and expression) and images of different individuals (due to variation in face shape). We follow the approach of Lanitis *et al* [64] and address this problem by deforming face images to a standard shape. In this procedure, the training images are deformed so that key landmark points are made to coincide. Details of the warping algorithm are given in Appendix A. The landmark points are those defined in the PDM, and each image is deformed to the average shape of the training set. Figure 3.6 shows some example faces and extracted ‘shape-free’ patches.

Each shape-free patch can be represented by a vector of grey-level values,  $\mathbf{g}$ :

$$\mathbf{g} = (g_1, g_2, \dots, g_n) \quad (3.11)$$

where  $g_i$  is the intensity of the  $i^{th}$  pixel in the shape-free patch. Principal Component Analysis of the shape-free grey-level vectors for the training images produces a model of the form:



**Figure 3.6:** Example faces with extracted ‘shape-free’ patches.

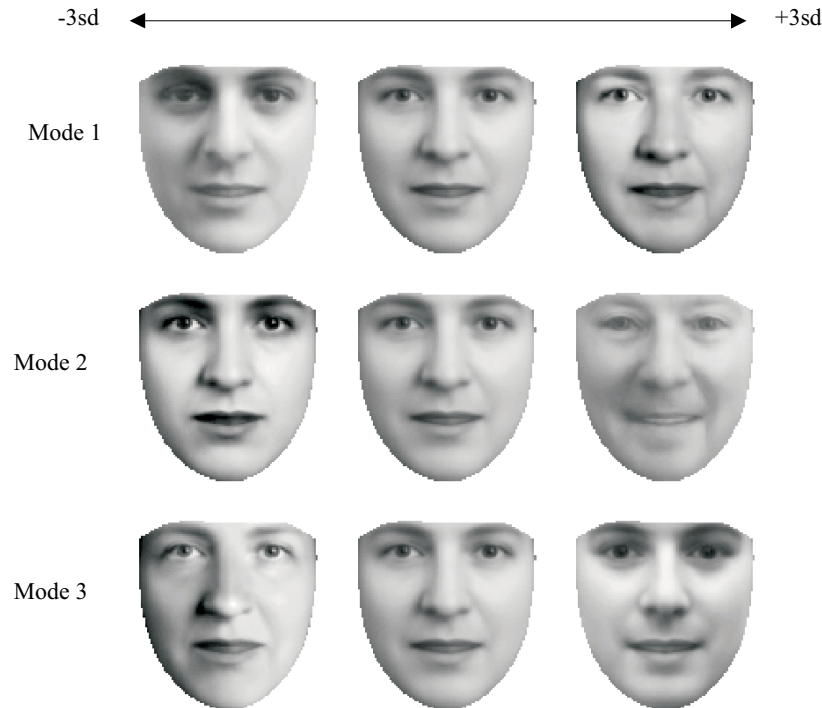
$$\mathbf{g} \approx \bar{\mathbf{g}} + \mathbf{P}_g \mathbf{b}_g \quad (3.12)$$

where  $\bar{\mathbf{g}}$  is the mean grey-level vector,  $\mathbf{P}_g$  is a set of orthogonal *modes of variation* and  $\mathbf{b}_g$  is a vector of grey-level parameters. Since the columns of  $\mathbf{P}_g$  are orthogonal, the set of model parameters for a given vector,  $\mathbf{g}$ , can be calculated by:

$$\mathbf{b}_g = \mathbf{P}_g^T (\mathbf{g} - \bar{\mathbf{g}}) \quad (3.13)$$

Variation encapsulated by this model is representative of ‘real’ grey-level variation in face images, rather than variation caused by lack of alignment in the training set.

Such a grey-level model is an important component of the combined appearance models we present in Chapter 4. Figure 3.7 shows the first three modes of variation of a typical shape-free face model.



**Figure 3.7:** First three modes of variation of a typical shape-free face model.

### 3.4 Interpreting faces using ASMs

A PDM provides a parameterised description of object shape, given by the shape vector,  $\mathbf{b}$ . Once an ASM search has located a plausible shape in an image, the shape vector can be used to interpret the meaning of the shape found. As has been pointed out by several authors [16] [31], using shape alone limits the accuracy of interpretation for face images; better interpretation must use grey-level information. Both local

grey-level models, and ‘shape-free’ grey-level models can be used for interpretation [64].

### 3.4.1 Classification

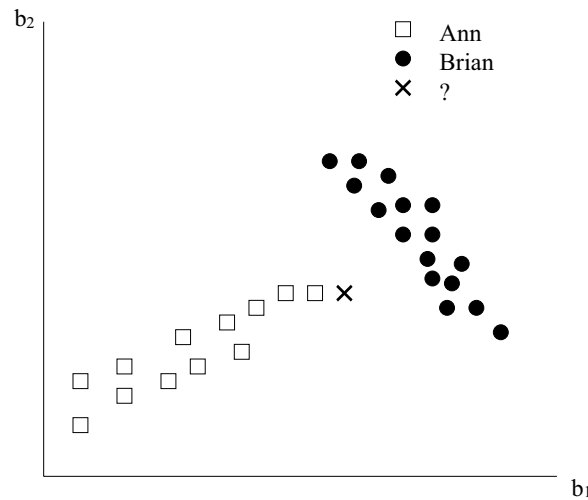
A simple and important type of face interpretation is classification, most often to decide the identity of the face. A typical task is to assign a previously unseen image to one of several possible identities. For each identity the system is previously presented with several examples of that person. If, for example, the classifier uses shape information, the vector of shape parameters,  $\mathbf{b}$ , extracted from the unseen images is used to measure the ‘distance’ between the image and each training group. The simplest measure of identity is given by the Euclidean distance between the model parameters of the located face and the centre of each class in the training set. This measure is, however, not satisfactory due to the confounding effect of other types of variation on face shape. It is essential to take account of the variation present in the training examples. A simple 2D illustration of this is shown in Figure 3.8. The squares and circles represent training examples plotted in shape parameter space. The unseen face lies closer to the centre of ‘Brian’ than ‘Ann’, yet an observation of the variability in the two classes shows that it is far more likely to be ‘Ann’ than ‘Brian’.

This problem can be overcome by using the *Mahalanobis Distance* [70]. This is a measure of the distance to class centroids, but which takes into account the spread of the individual classes and correlation between variables. Let  $\bar{\mathbf{b}}_i$  be the mean model parameter vector of class  $i$ . The Mahalanobis Distance,  $D_i$  between an observation  $\mathbf{b}$  and the class  $i$  is given by:

$$D_i^2 = (\mathbf{b} - \bar{\mathbf{b}}_i)^T \mathbf{C}_i^{-1} (\mathbf{b} - \bar{\mathbf{b}}_i) \quad (3.14)$$

where  $\mathbf{C}_i$  is the covariance matrix for the training examples of class  $i$ . In order to





**Figure 3.8:** Illustration of the effect of training-class variability. Unknown example is more likely to be Ann than Brian, even though it lies closer to the centroid of Brian.

classify a new example, the Mahalanobis Distance between the observation and each class centre is calculated. The observation is assigned to the ‘nearest’ class.

### 3.4.2 Identification using shape and texture

Lanitis [61] [65] used an ASM algorithm to locate faces in images. Several classification experiments were performed using shape, local grey-level models, global shape-free grey-level models, and combinations of the three.

The experiments used a training set consisting of 160 images, 8 each for 20 individuals. This training set was used to build a PDM, a set of local grey-level models, and a shape-free grey-level model. Identification trials were performed on 2 test sets, a so-called ‘normal’ set containing 200 images without occlusions, and a ‘difficult’ test set of 60 images in which occlusions were present. Figure 3.9 shows some example images from the three sets.

The system was shown a test image and ASM search was performed, given a



**Figure 3.9:** Examples from the three image sets used by Lanitis to evaluate ASM-based recognition.

reasonable starting approximation<sup>†</sup>. ASM search produced shape and local grey-level model parameters. The resulting shape was used to warp the face into the shape-free reference frame, and the shape-free grey-level model parameters computed using equation 3.13. The face was then classified using the smallest Mahalanobis distance from the centre of each training class. Lanitis [61] recorded correct matches and occasions when the correct match was returned as one of the three most likely faces. Table 3.1 gives the classification results reported. It should be noted that these classification results do not allow for rejection, that is, when the test face is not recognised at all. Since the entire set of test faces are known to be in the gallery, this forced-choice method is not an entirely fair test, and can produce over optimistic results.

---

<sup>†</sup>Since all the faces occur in roughly the centre of the image, the ASM could simply be started from a central position each time.

	Normal test set		Difficult test set	
	Correct	Within 3	Correct	Within 3
Shape model	50.3%	66.6%	15.6%	31.1%
Shape-free grey model	78.7%	87.3%	31.1%	53.3%
Local grey-level models	77.3%	89.7%	28.9%	57.8%
Shape + shape-free models	85.3%	93.3%	34.4%	56.7%
Shape + local models	80.0%	90.3%	34.4%	66.7%
All methods	92.0%	97.0%	48.9%	77.4%

**Table 3.1:** Classification results reported by Lanitis [61].

### 3.5 Discussion of the ASM-based approach

Table 3.1 shows encouraging results using the ASM approach. In particular, combining all the shape and grey-level information appears to give better performance than using any of the models alone.

Unfortunately, the combination of the models occurs only at the final classification stage; until then, the system uses 3 completely separate models. This has several drawbacks. Firstly, there is considerable overlap between the information encapsulated by the models. Local grey-level models contain some variation explained in the shape-free grey-level model, and some of the change in shape-free grey-level variation is correlated with variation in the shape model. Not only does this mean the representation is redundant, but also, since the models are treated as independent, the representation is not specific - it is possible to generate illegal combinations of shape

and grey-level appearance, such as a shape which represents a closed mouth, but a shape-free patch showing teeth. This lack of specificity can only serve to degrade both location and classification performance.

The ASM search algorithm does not make full use of the information modelled, the shape-free grey-level model is sometimes used to give an overall final fit measure, but otherwise plays no part in image search. Moreover, the information used in ASM search is not used optimally - again because the algorithm does not account for correlation in the data. Not only are local grey-level models correlated with the shape model, but individual local models are correlated with each other; this is particularly obvious for say, two nearby edge points. Haslam [48] describes a method in which the local models are concatenated to produce a single model, accounting for the correlations between individual models, however, the ASM search algorithm cannot be used with this model. Alternative search strategies such as Genetic Algorithms must be used, which can be successful as Hill and Taylor [50] have shown, but cannot approach the speed of ASMs.

In this thesis we develop a unified approach to modelling which addresses all these problems. We describe the construction of a complete model of appearance, controlled by a single set of uncorrelated parameters. Further, we describe a search algorithm that makes use of all the information in the model, both shape and global grey-level appearance. The new method uses a more specific representation, and all the image data, whilst achieving the speed of the ASM method.

## 3.6 Summary

This chapter has introduced Point Distribution Models (PDMs), which are used as part of the unified appearance model we describe later in the thesis. A PDM is built from a training set, using Principal Component Analysis (PCA), a technique we have described and will use again. By using a training set, the PDM comes close to

achieving the twin goals of generality and specificity; however only a limited amount of the information present in face images is represented. We have also described the construction of shape-free grey-level models of the full-face region, which capture additional information that is assumed to be independent of shape.

We have given a brief overview of the Active Shape Model algorithm, which uses a PDM in conjunction with local grey-level models to drive image search. ASM search was used by Lanitis [61] as the basis for several face recognition experiments. In summary, the results showed that, having located the face with an ASM, the best results were achieved by using a combination of shape, local grey-level, and shape-free grey-level models for classification.

Finally, we have discussed the shortcomings of the ASM approach. In particular the combination of shape and grey-level models introduces a lack of specificity because the assumption of independence is invalid.

# Chapter 4

## Appearance Models

In this chapter we describe a new approach to face modeling using *Appearance Models*<sup>\*</sup>. These are statistical models of the appearance of faces in images, learnt from a set of training images. Unlike Point Distribution Models and Shape-Free Region Models, the approach encapsulates both shape and grey-level variation in a single model. The method of constructing an Appearance Model is described and illustrative results are given.

### 4.1 Motivation

The model-based methods presented in the previous chapter do not unify the full shape and texture information in face images. In order to encapsulate the information required for both location and interpretation, three separate models are used. Since the models are not completely independent there is inevitably some redundancy and lack of specificity in the representation. The aim is to produce a single representation

---

<sup>\*</sup>This type of model was introduced by Edwards *et al* [38] and referred to as a *Combined Appearance Model*; since then we have adopted the shorter version of the name. This should not be confused with the earlier work of Lanitis [64] who used the term *Appearance Model* to describe the coupling of a PDM and Shape Free Grey-Level Model. Unless explicitly stated otherwise, the term is used to refer to the new type of combined model.

that encapsulates all the variation of face images in a single model.

Particularly desirable is a generative model - a model capable of reconstructing synthetic examples of face images. The ability to synthesise complete, photo-realistic faces is itself a desirable property, but it is in the analysis of face images that a complete representation is important. In fact, this requirement can be taken as a necessary (but not sufficient) condition that must be fulfilled by a specific and general model. If the system claims to understand face images then it ought to be able to reproduce them.

In many machine vision applications, images are analysed by making a limited set of measurements - edges for example. The Active Shape Model method relies upon a relatively small number of measurements around landmark points. Although a reduced number of measurements makes for efficient processing, it inevitably reduces the power of further analysis. In any measurement system, a reduction in measurement dimensionality risks losing discriminatory power. If we begin by choosing arbitrary features such as edges at the start of the analysis it is very difficult to quantify how much interpretation power we lose by ignoring other features. By starting with a complete representation of face appearance, we have the opportunity to reduce the dimensionality at a later stage if required, but with analysis of and control over the loss of discriminatory power.

## 4.2 Formulation

An Appearance Model is generated by combining a Point Distribution Model with a Shape-Free Region Model. The PDM explains variation in face shape, whilst the region model explains intensity variation, but with the important step of shape normalisation - the warping of the training images to an average shape is important in establishing correspondence.

We begin, as previously, with a training set of labelled images, where key landmarks are marked on each face. Given such a training set, we can generate a Point Distribution Model as described previously. Recall that all the training shapes are aligned into a common frame before applying Principal Component Analysis (PCA) to the data as described in section 3.1. Any example can then be approximated using:

$$\mathbf{x} \approx \bar{\mathbf{x}} + \mathbf{P}_s \mathbf{b}_s \quad (4.1)$$

where  $\bar{\mathbf{x}}$  is the mean shape,  $\mathbf{P}_s$  is a set of orthogonal *modes of shape variation* and  $\mathbf{b}_s$  is a set of shape parameters.

To build a statistical model of the grey-level appearance we use the method described in section 3.3, warping each example image so that its control points match the mean shape (using the triangulation algorithm given in Appendix A). We then sample the grey-level information  $\mathbf{g}_{im}$  from the shape-free image patch. The warping stage ensures that there is correspondence between grey-level values of pixels over the training set.

Most naturally acquired training sets will contain considerable variation due to large-scale properties of the lighting and camera configuration. We account for this by extending the approach given in section 3.3. The effect of variation in image brightness and contrast across the training set can be removed in advance by applying normalisation to the shape-free patches. This prevents the model encapsulating such variation as a natural part of face variation. This is desirable from a both reconstruction and analysis point of view; the effects of global lighting change can be extracted or synthesised separately. To minimise the effect of global lighting variation, we normalise each training sample by applying a scaling,  $\alpha$ , and offset,  $\beta$ ,

$$\mathbf{g} = (\mathbf{g}_{im} - \beta \mathbf{1}) / \alpha \quad (4.2)$$



The values of  $\alpha$  and  $\beta$  are chosen to best match the vector to the normalised mean sample of the whole training set. Let  $\bar{\mathbf{g}}$  be the mean of the normalised data, scaled and offset so that the sum of the elements is zero and the variance of the elements is unity. The values of  $\alpha$  and  $\beta$  required to normalise  $\mathbf{g}_{im}$  are then given by:

$$\alpha = \mathbf{g}_{im} \cdot \bar{\mathbf{g}} \quad , \quad \beta = (\mathbf{g}_{im} \cdot \mathbf{1})/n \quad (4.3)$$

where  $n$  is the number of elements in the vectors.

Obtaining the mean of the normalised data is then a recursive process, as the normalisation is defined in terms of the mean. A stable solution can be found by using one of the examples as the first estimate of the mean, aligning the others to it (using 4.2 and 4.3), re-estimating the mean and iterating.

By applying PCA to the normalised data we obtain a linear model of the form:

$$\mathbf{g} \approx \bar{\mathbf{g}} + \mathbf{P}_g \mathbf{b}_g \quad (4.4)$$

where  $\bar{\mathbf{g}}$  is the mean normalised grey-level vector,  $\mathbf{P}_g$  is a set of orthogonal *modes of grey-level variation* and  $\mathbf{b}_g = (b_{g1}, b_{g2}, \dots, b_{gt})$  is a set of grey-level appearance parameters<sup>†</sup>.

The shape and appearance of any example can thus be summarised by the vectors  $\mathbf{b}_s$  and  $\mathbf{b}_g$ . Since we expect some correlation between the shape and grey-level variation, we apply a further PCA to the data as follows. For each example we generate the concatenated vector  $\mathbf{b}$ ,

$$\mathbf{b} = \begin{pmatrix} \mathbf{W}_s \mathbf{b}_s \\ \mathbf{b}_g \end{pmatrix} = \begin{pmatrix} \mathbf{W}_s \mathbf{P}_s^T (\mathbf{x} - \bar{\mathbf{x}}) \\ \mathbf{P}_g^T (\mathbf{g} - \bar{\mathbf{g}}) \end{pmatrix} \quad (4.5)$$

---

<sup>†</sup>For simplicity of notation we always use  $t$  to signify the number of elements in a particular parameter vector, although the actual number of elements in the shape and grey-level parameter vectors is usually different.

where  $\mathbf{W}_s$  is a diagonal matrix of weights,  $w_{s1}, w_{s2}, \dots, w_{st}$ , one for each shape parameter, allowing for the difference in units between the shape and grey models (see below). We apply PCA to these vectors, giving a further model

$$\mathbf{b} \approx \mathbf{Q}\mathbf{c} \quad (4.6)$$

where  $\mathbf{Q}$  are the eigenvectors of the covariance of  $\mathbf{b}$  over the training set and  $\mathbf{c}$  is a vector of *appearance* parameters,  $c_1, c_2, \dots, c_t$ , controlling both the shape and grey-level appearance of the model. Since the shape and grey-model parameters have zero mean,  $\mathbf{c}$  does too.

Since the columns of  $\mathbf{Q}$  are orthogonal, we can obtain  $\mathbf{c}$  for a given  $\mathbf{b}$  simply:

$$\mathbf{c} = \mathbf{Q}^T \mathbf{b} \quad (4.7)$$

Note that the linear nature of the model allows us to express the shape and grey-levels directly as functions of  $\mathbf{c}$

$$\mathbf{x} \approx \bar{\mathbf{x}} + \mathbf{P}_s \mathbf{W}_s \mathbf{Q}_s \mathbf{c} \quad , \quad \mathbf{g} \approx \bar{\mathbf{g}} + \mathbf{P}_g \mathbf{Q}_g \mathbf{c} \quad (4.8)$$

where

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q}_s \\ \mathbf{Q}_g \end{pmatrix} \quad (4.9)$$

An example image can be synthesised for a given  $\mathbf{c}$  by generating the shape-free grey-level image from the vector  $\mathbf{g}$  and warping it using the control points described by  $\mathbf{x}$ .

### 4.2.1 Choice of shape parameter weights

The elements of  $\mathbf{b}_s$  have units of distance, those of  $\mathbf{b}_g$  have units of intensity, so they cannot be combined directly. Because  $\mathbf{P}_g$  has orthogonal columns, varying  $\mathbf{b}_g$  by one unit moves  $\mathbf{g}$  by one unit. To make  $\mathbf{b}_s$  and  $\mathbf{b}_g$  commensurate, we must estimate the effect of varying  $\mathbf{b}_s$  on the sample  $\mathbf{g}$ . To do this we systematically displace each element of  $\mathbf{b}_s$  from its optimum value on each training example, and sample the image given the displaced shape. The RMS change in  $\mathbf{g}$  per unit change in shape parameter  $b_{si}$  gives the weight  $w_{si}$  to be applied to that parameter in equation (4.5).

## 4.3 Example of a face model

In this section we present the face model used in this thesis, built using the technique outlined above. The model was built using a training set of 768 images (details of the training images used are given in Appendix B). For each of these images the shape landmark points were located by hand. Figure 4.1 shows a selection of typical images taken from the training set. Most of the training images were greyscale, with a small number also available in colour. The model used for the most of the work in this thesis was built by first converting all the images to grey-scale, although we later show how colour images can be used to build colour models.

A key feature of the set of training images is the range of variability. If we are to build a model capable of generalising to new faces, the training set must contain a wide range of variation in identity, lighting, expression and pose (at least up to the range of pose we intend the model to work with).

The resulting model contained 85 modes of variation, each controlling a particular combination of shape and texture variation. This was sufficient to capture 98% of the variation in the training set.

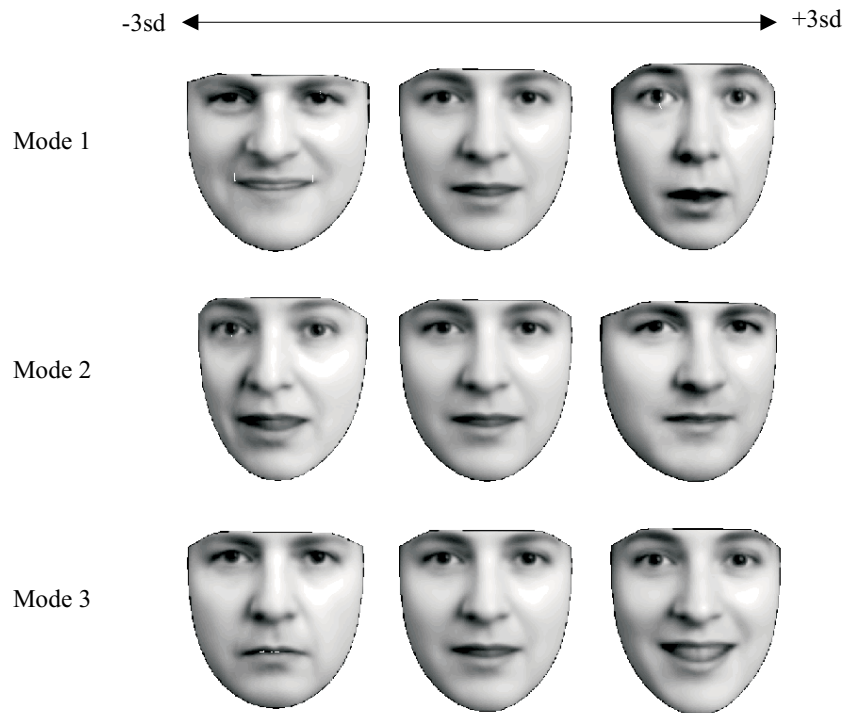


**Figure 4.1:** Selection of typical face images from the training set.

### 4.3.1 Visualisation

Recall from equation 4.8 that we can reconstruct an example face by choosing suitable values of the model vector  $\mathbf{c}$ . The range of allowable values for each element of  $\mathbf{c}$ ,  $c_i$  can be estimated by noting that the eigenvalues obtained in the PCA give the variance of the training data in the direction of the corresponding eigenvector. We can visualise the *modes of variation* of the appearance model in the same way as the Point Distribution Model (see Figure 3.2), by varying each of the model parameters separately. The effect of varying the first three parameters of the Appearance Model is shown in Figure 4.2.

At this point we can note some characteristics of the model. Each mode of variation describes a combination of effects. The first mode shows pose change, and there

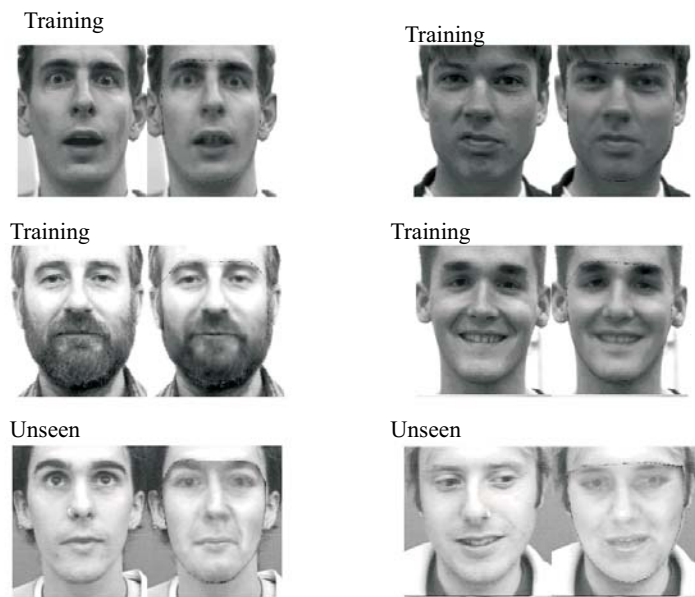


**Figure 4.2:** The effect of varying the first three parameters of the appearance model between  $\pm 3$  s.d's.

is clearly change in the identity of the face, as well as the expression and lighting conditions. This is to be expected, since no distinctions are made between images during training - equal weight is given to each face regardless of its characteristics. This compounding of several 'real' sources of variation into single modes of variation is not ideal for some interpretation and synthesis tasks. Ideally, the model would have distinct sets of parameters with which we could control and analyse properties such as pose and expression, independently of other types of variation. This problem is addressed later in the thesis.

### 4.3.2 Fitting the model by hand

Given a new image, with a set of hand annotated landmark points, we can generate the model’s closest possible representation of the data, the so called, ‘best-fit’. This is a two-step process: first we calculate the parameters of the shape and shape-free grey-level models using equation 4.5. Applying the appropriate shape parameter weights and concatenating the parameter vectors, we then use equation 4.7 to find the parameter vector,  $\mathbf{c}$ , for the Appearance Model. We then reconstruct the image using equation 4.8. Figure 4.3 shows the result of fitting the model to images from the training set and to unseen test images (which have also been hand-landmarked).



**Figure 4.3:** Reconstruction of images from training set and unseen images.

### 4.3.3 Limitations of the reconstruction method

Ideally, the perfect model would be able to reconstruct any new face presented to the system. This assumes that the training procedure has captured sufficient variation to generalise to unseen faces. Observation of the reconstructions in Figure 4.3 shows that the reconstruction error is generally less satisfactory for the unseen images. A major limitation in the method used to compute the reconstruction is that it is *shape-dominated* - any error in the face shape causes the wrong set of pixels to be projected into the region model, which can have a serious effect on the model parameters and subsequent reconstruction. This is not only a problem if the landmarks are badly placed, but is a limitation of the model itself. Even a slight inability of the shape part of the model to represent the given landmarks of the unseen image can cause a large sampling error in the region model.

We believe this to be a limitation of the face interpretation scheme presented by Lanitis *et al* [61]. In Lanitis' method, faces are located using an Active Shape Model. The located region is then sampled and interpreted using a grey-level model. As Figure 4.3 shows, even with careful hand placement of the landmark points, shape dominated model fitting leads to reconstruction errors.

In Chapter 6 we introduce a unified image interpretation scheme which fits both shape and texture simultaneously. At this point we simply present an example of the result obtainable. The method allows the same model to reconstruct unseen data more satisfactorily. The given landmark points do not dominate matching - the shape is allowed to vary slightly in order to produce a better texture reconstruction as illustrated in Figure 4.4.

### 4.3.4 Specificity

The key requirement of a successful face model is that it should not be able to generate implausible examples of faces. We can qualitatively assess the specificity of the model



**Figure 4.4:** Reconstruction using unified fitting method. Left - original image, Centre - fitting with shape-dominated scheme, Right - unified fitting scheme.

by generating faces using random values of the model vector. Whilst not proving that *all* images the model could ever generate are legal, this is an efficient test when faced with the 80 plus dimensions of the model. We define the model's scope as the images that can be produced by choosing any values of the parameter vector  $\mathbf{c}$  such that:

$$-rs_i < c_i < rs_i \quad (4.10)$$

where  $r$  is the number of standard deviations of variation allowed. Typically we set  $r = 2$ , restricting all generated faces to be within 2 standard deviations of the average of the training set. For normally distributed parameters this choice should include over 95% of plausible model instances. Figure 4.5 shows a range of images randomly generated by the model. Most of the faces appear plausible - the white specks seen in some images are created when the shape varies so much as to cause triangles in the warping algorithm to overlap.

## 4.4 Summary

In this chapter we have described the construction of Appearance Models. An Appearance Model combines a Point Distribution Model with a Shape-Free Grey-Level Model to produce a unified representation of facial appearance.



The model is controlled by a compact set of parameters. By varying these parameters we can visualise the space represented by the model. The model is general and specific, capable of representing unseen faces, but not of generating implausible examples. The drawback of the approach is the compounding of real-world sources of variation into single model parameters. We address this issue in the following chapters.

At this point we have introduced Appearance Models as a representation, and shown how they can be used to generate synthetic face images. In Chapter 6 we will describe a method that uses Appearance Models for face location and interpretation.



**Figure 4.5:** A selection of random faces generated by the model.

# Chapter 5

## Partitioned Models

This chapter describes methods of improving the specificity of the face Appearance Model. We introduce methods of partitioning a model into separate subspaces, producing representations that encapsulate specific sources of variation such as identity and expression. We demonstrate how face images can be projected onto these subspaces and manipulated.

### 5.1 Motivation

A vital part of the model-based approach to face understanding is the ability of the model to generalise to as wide a range of faces as possible. We have shown how the Appearance Model representation achieves this, representing a wide range of individuals' faces, with various expressions in a range of pose and lighting conditions. Unfortunately, the price of this generality is the confounding of the separate sources of variation. A single parameter of the Appearance Model can affect the face in many ways - for example, changing both identity and expression. A more source-specific representation would provide advantages for interpretation, synthesis and tracking applications and may also simplify model building.

### 5.1.1 Interpretation

Face interpretation tasks involve, by definition, the extraction of information that has meaning in human terms. The information content of a face image can be naturally divided into 3 categories:

- inter-face variation (identity)
- intrinsic intra-face variation (expression/speech)
- extrinsic intra-face variation (pose/lighting)

In model-based face interpretation, information is obtained by analysing the model parameters extracted by matching to a face image. The analysis of this multivariate data is much more tractable if a representation can be found in which the measured variables behave orthogonally. Whilst the model parameters are orthogonal *over the whole training set*, they turn out to be correlated over particular types of variation, such as identity, expression or pose. The aim of model partitioning presented here is to produce orthogonal sets of parameters that encapsulate particular aspects of the appearance of faces in images.

### 5.1.2 Synthesis

Appearance Models can generate realistic reconstructions of faces. By varying the model parameters we can manipulate these reconstructions, producing new, novel images which remain convincing faces, but are nevertheless very different from the original image. Without any correspondence between real-world variation and the individual model parameters, it is difficult, however, to achieve any useful manipulation of face images, such as simulated speech or expression change. Partitioning the model aims to allow the manipulation of particular facial affects, for example, changing the

expression of a face from sad to happy, without affecting other properties such as identity.

### 5.1.3 Tracking

One of the key aims of this project is reliable face tracking. A useful system ought to deal with a wide range of individuals, and should not need priming with prior information about the identity of person being tracked. All analysis, including identification should be automatic. The model used must be capable of representing different identities. In a tracking scenario, however, this generality becomes a handicap. Once the tracker has a confident ‘lock’ on the face, there ought to be an extra powerful constraint - the *identity* of the face must remain fixed. A model that allows the identity to change throughout tracking clearly lacks specificity, and thus robustness. Eliminating identity variation completely (by modelling a single individual, say) would, however, prevent the model from fitting to unknown faces. Ideally we would like independent control of the identity part of the model, so that the variability can be controlled as required.

### 5.1.4 Model building

One problem with simple Appearance Models is selecting a sufficiently large training set to account for all sources of variation. As well as providing the raw training images, additional information is available - for example, many images are multiple shots of the same individual and many images are labelled with expression. By focusing on describing each type of real-world variation separately it may be possible to reduce the number of training images required. This idea is the basis of an iterative model building approach developed by Costen *et al* [28].

## 5.2 Modelling subspaces

The Appearance Model defines allowable *modes of variation* by Principal Component Analysis of a large training set. This produces a parameterised model capable of generating new, unseen (but always plausible) examples of faces. In this chapter we attempt to find modes of variation which can only produce plausible examples of *certain types of variation*. For example, we would like to encapsulate face variation due to expression change only.

The full Appearance Model can be thought of as encapsulating face variation in a high dimensional *vector space* defined by the model parameters,  $\mathbf{c}$ , given in equation 4.7. We seek projections of the data onto lower dimensional *subspaces* controlling particular types of variation.

Although we might attempt to build Subspace Models by analysis of the original shape and grey-level data, we choose to perform the analysis in the space defined by the Appearance Model. The methods described in this chapter require the calculation of within-class covariance matrices. If these are built using the original data they will usually be singular, since the dimensionality of the raw data is usually greater than the number of training examples. Calculating the within-class covariance matrix in the frame of the Appearance Model avoids this problem. A further benefit arises, since the mapping from raw points and pixel values into Appearance Model parameters constitutes a reduction in dimensionality from several thousand dimensions to a few tens. In many applications we will need to perform analysis on multiple subspaces such as pose, expression, identity and lighting at the same time. By building these subspaces in the Appearance Model space, the large dimensionality reduction (and correspondingly large matrix multiplication) is only performed once and the mapping from Appearance Model parameters to subspace parameters is comparatively cheap.

A potential danger of this method is that of encapsulating too little variation in the original Appearance Model. For example, it may be that subtle movement of

the eyebrows is statistically insignificant over the whole training set, but might be important for interpreting expressions. This effectively means that the Appearance Model has rejected real variation as noise. The results presented in this chapter show that this problem can be avoided in practise.

We have investigated several methods of estimating suitable subspaces using still images as training data. Each of these involves extra knowledge about specific images, such as the identity or expression of the individual. In each type of analysis we have adopted a linear approach to the problem of isolating real-world sources of variation. Initially, we assume that there exists a representation in which sources of variation are linearly independent. Thus, we assume one can, for example, manipulate the expression of a face without changing its identity or pose. This turns out to be a useful approximation. In Chapter 7 we introduce an improved approximation based on analysing video sequences.

### 5.3 Linear Discriminant Analysis

Given that an Appearance Model provides a compact description of the training data, we seek a further analysis of the data that yields a description of specific sources of variation. A natural approach is to seek a description of the data that maximises separation of subclasses related to that particular source. For example, to build a model of expression variation, one might try to define a set of modes of variation which give the maximum separation between the classes, *happy*, *sad*, *etc.*, whilst minimising other types of variation. A common technique for this addressing this type of problem is *Linear Discriminant Analysis (LDA)*. For a detailed background to discrimination and classification techniques, the reader is referred to books by Hand [47], and Johnson and Wichern [54]. LDA has been applied to face analysis in an eigenface formulation by Belhumeur *et al* [7].

The basic requirement for LDA is a set of training data to which class labels have

been attached. Usually there exist many training examples per class. Discriminant analysis can be applied naturally to the problem of calculating an identity subspace; it is easy to apply class labels to the training images. Calculating an expression subspace using this technique is more difficult, requiring a rather arbitrary choice of labelling scheme.

### 5.3.1 Formulation

We build a *Discriminant Subspace* by taking an existing Appearance Model, and attempting to find linear transformations of the modes of variation which yield new modes describing specific attributes such as identity or expression. For each of the examples in the original Appearance Model training set, we attach a class label, such as the identity or expression of the face. We then calculate the vector of Appearance Model parameters,  $\mathbf{c}_k$ , for each example,  $k$ , using equation 4.7 (given the known locations of the landmarks). This produces a training set of  $N$  vectors each uniquely assigned to one of  $l$  classes,  $\{C_1, C_2, \dots, C_l\}$ . We define a *between-class* scatter matrix as

$$\mathbf{B} = \sum_{i=1}^l N_i (\bar{\mathbf{c}}_i - \bar{\mathbf{c}})(\bar{\mathbf{c}}_i - \bar{\mathbf{c}})^T \quad (5.1)$$

where  $\bar{\mathbf{c}}$  is the mean of all the examples,  $\bar{\mathbf{c}}_i$  is the mean of class  $i$ , and  $N_i$  is the number of examples in class  $i$ . A *within-class* scatter matrix is defined as

$$\mathbf{W} = \sum_{i=1}^l \sum_{\mathbf{c}_k \in C_i} (\mathbf{c}_k - \bar{\mathbf{c}}_i)(\mathbf{c}_k - \bar{\mathbf{c}}_i)^T \quad (5.2)$$

We seek an orthogonal mapping between the model parameters,  $\mathbf{c}$ , and a lower-dimensional space, defined by a new vector of model parameters,  $\mathbf{d}$ , according to:

$$\mathbf{c} \approx \mathbf{D}\mathbf{d} \quad (5.3)$$



where  $\mathbf{D}$  is a matrix of orthogonal eigenvectors. Since  $\mathbf{D}$  is orthogonal, given a set of parameters,  $\mathbf{d}$ , the projection back into the original space will be given by:

$$\mathbf{d} = \mathbf{D}^T \mathbf{c} \quad (5.4)$$

The optimal choice of  $\mathbf{D}$  is the orthonormal matrix which maximises the ratio of the determinants of the within-class and between-class covariance matrices.

$$\begin{aligned} \mathbf{D}_{opt} &= \arg \max_{\mathbf{D}} \frac{|\mathbf{D}^T \mathbf{B} \mathbf{D}|}{|\mathbf{D}^T \mathbf{W} \mathbf{D}|} \\ &= [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_m] \end{aligned} \quad (5.5)$$

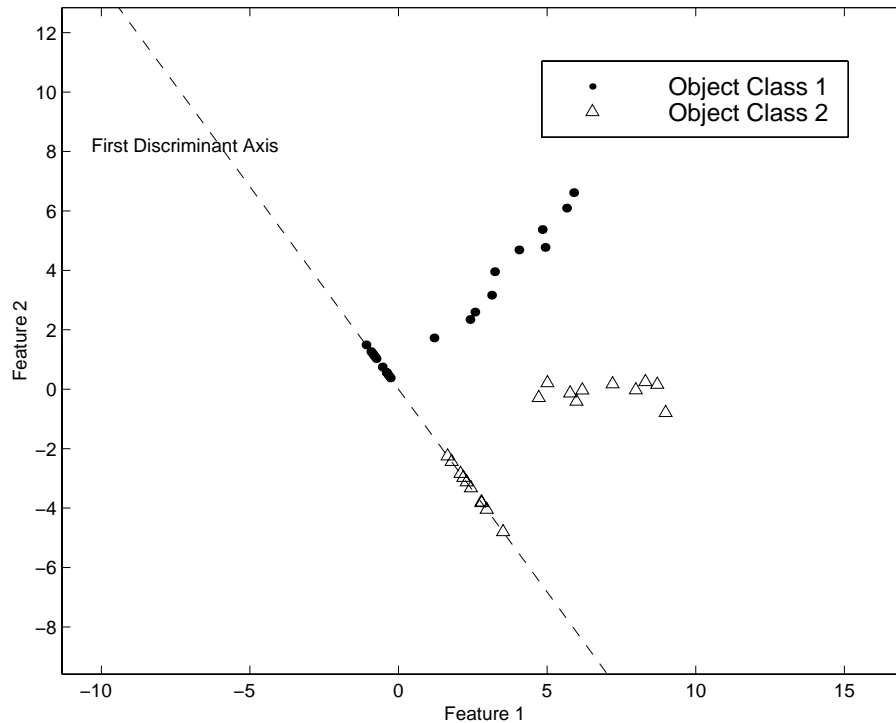
where the columns of  $\mathbf{D}_{opt}$ ,  $\{\mathbf{e}_i | i = 1, 2, \dots, m\}$  are the  $m$  generalised eigenvectors of  $\mathbf{B}$  and  $\mathbf{W}$  corresponding to the  $m$  largest generalised eigenvalues,  $\{\lambda_i | i = 1, 2, \dots, m\}$ :

$$\mathbf{B} \mathbf{e}_i = \lambda_i \mathbf{W} \mathbf{e}_i, \quad i = 1, 2, \dots, m \quad (5.6)$$

The eigenvectors corresponding to non-zero eigenvalues represent the basis vectors of the space in which between-class variation is maximised at the expense of within-class variation. There are a maximum of  $(l - 1)$  non-zero eigenvalues. The corresponding eigenvalues reflect the amount of separation achieved by each basis vector. The eigenvector corresponding to the largest eigenvalues represents the greatest separation, the second represents the next most separation, and so on up to  $m$ . The analysis is performed on the Appearance Model parameters rather than the raw data (points and grey-level samples), since the rank of  $\mathbf{W}$  is at most  $(N - l)$ . Since the dimensionality of the raw data is likely to be much greater than  $N$ ,  $\mathbf{W}$  would always be singular, meaning that it would be possible to choose axes such that the within-class spread was zero. By performing LDA in the much lower dimensionality of the Appearance Model space we avoid this problem.

Figure 5.1 illustrates 2D discriminant analysis for a synthetically generated two-

class problem. Here the circles represent objects from one class, the triangles objects from a second class. The dashed line shows the first *Discriminant Axis*. This axis gives the principal direction of group separation.



**Figure 5.1:** Linear Discriminant Analysis in two dimensions. Examples from each class are shown scattered in 2D - each is also shown projected onto the single discriminant axis. This projection yields the optimum group separation.

## 5.4 Residual subspaces

As well as the directions describing a particular type of variation, we are also interested in the remaining orthogonal directions not spanned by the Discriminant Subspace. For example, in the case of identity variation we could use LDA to build an identity subspace. The residual subspace will represent that variation which is not related to the identity of faces.

In order to get a reliable estimate of the variation in the residual space we calculate a *Residual Subspace* by projecting out of the full Appearance Model any variation that is explained by an existing subspace model. Given the Appearance Model parameters of a training example,  $\mathbf{c}$ , we can combine equation 5.4 and 5.3 to give a estimate  $\mathbf{c}'$  of the model parameters that result *if the variation is explained by only the subspace model*:

$$\mathbf{c}' = \mathbf{D}\mathbf{D}^T\mathbf{c} \quad (5.7)$$

We can then calculate a vector of residual variation  $\delta\mathbf{c}$ , that was not explained by the Subspace Model:

$$\delta\mathbf{c} = \mathbf{c} - \mathbf{c}' \quad (5.8)$$

We calculate  $\delta\mathbf{c}$  for each training image, and perform PCA on these to find the eigenvectors which describes the residual variation. Variation in the residual subspace is parameterised by a vector of eigenvector weights,  $\mathbf{r}$ . The mapping between Appearance Model parameters,  $\mathbf{c}$  and the Residual Model parameters,  $\mathbf{r}$ , is given by:

$$\mathbf{c} \approx \mathbf{R}\mathbf{r} \quad (5.9)$$

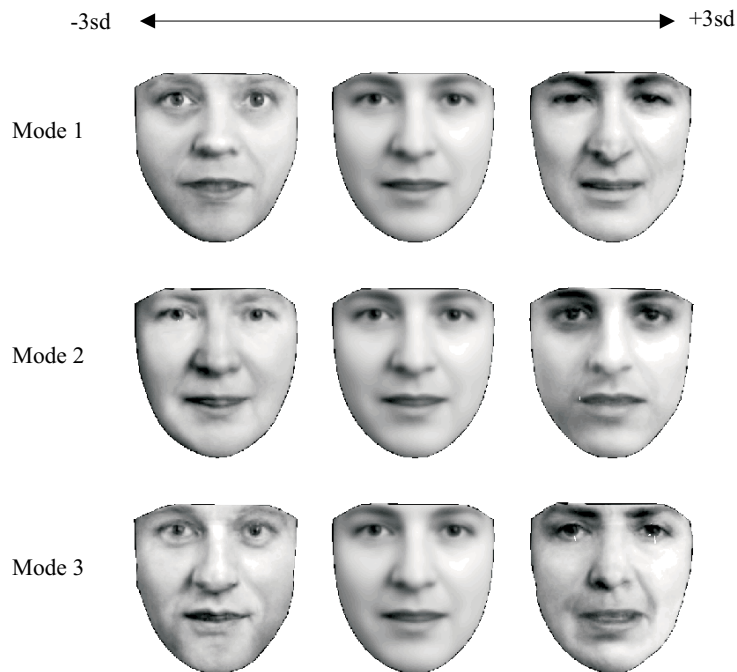
$$\mathbf{r} = \mathbf{R}^T\mathbf{c} \quad (5.10)$$

where  $\mathbf{R}$  is a matrix of orthogonal eigenvectors. Since the mean of  $\delta\mathbf{c}$  will be zero over the training set, there is no constant term in equations 5.9 or 5.10.

Since  $\delta\mathbf{c}$  contains no variation in the space defined by  $\mathbf{D}$ , the spaces defined by  $\mathbf{D}$  and  $\mathbf{R}$  are mutually orthogonal; any change to the Discriminant Parameters,  $\mathbf{d}$ , has no effect on the Residual Parameters,  $\mathbf{r}$ . Moreover, the dimensionality of  $\mathbf{D}$  and  $\mathbf{R}$  sum to the dimensionality of the original space.

## 5.5 Identity model using LDA

A large number of the training examples used to build the original Appearance Model were multiple images of the same person. In total, we labelled between 5 and 10 images of each of 50 individuals. We used these images and labels to perform Linear Discriminant Analysis on the Appearance Model. The resulting Discriminant Model had 49 modes of variation. We can visualise the effect of varying the value of  $\mathbf{d}_{identity}$  by using equation 5.3 to compute  $\mathbf{c}$  and then reconstructing the image using equation 4.8. The effect of varying the first 3 parameters of  $\mathbf{d}_{identity}$  is shown in Figure 5.2.

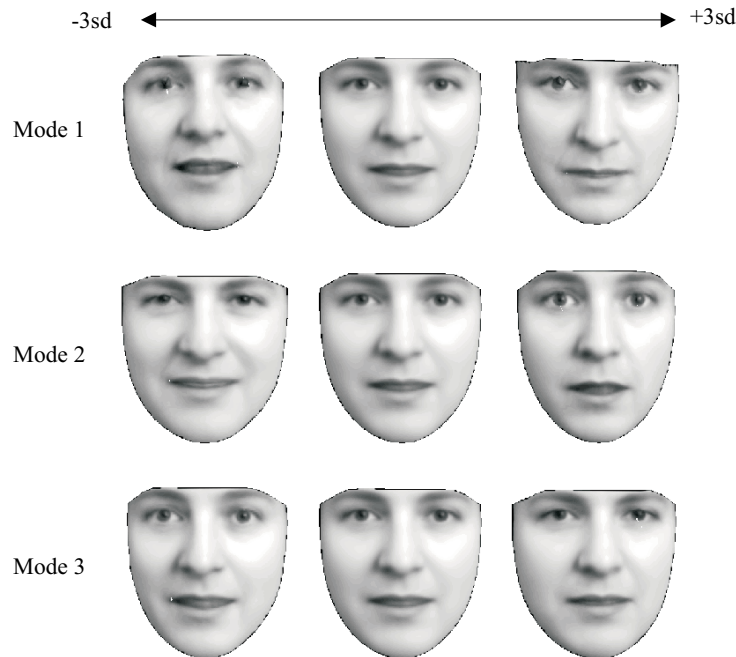


**Figure 5.2:** Effect of varying the first 3 parameters of the ‘identity’ subspace model built using LDA.

The modes show considerable variation in the nose, eyes, eyebrows and mouth, as well as some variation in the overall shape of the face. There appears to remain a small amount of expression and lighting variation, but little variation in pose.

### 5.5.1 Non-identity model

Given the identity model illustrated in Figure 5.2, we can compute a residual model which describes non-identity variation as described in Section 5.4. This analysis results in a subspace defined by 36 parameters. The effect of varying the first 3 parameters of  $\mathbf{r}_{non-id}$  is shown in Figure 5.3.



**Figure 5.3:** Effect of varying the first 3 parameters of the ‘non-identity’ subspace model built by analysis of data after ‘projecting-out’ identity variation.

### 5.5.2 Projecting images onto subspaces

Using equations 5.4 and 5.10, we can visualise the effect of projecting an image onto either of the subspaces calculated above. We can think of this as visualising the ‘identity’ and ‘non-identity’ components of the image separately. The identity

and non-identity spaces are mutually orthogonal and together encapsulate all the information in the Appearance Model. Figure 5.4 shows some original images together with their respective projections onto the identity and non-identity subspaces.



**Figure 5.4:** Original images projected onto identity and non-identity subspaces respectively.

Figure 5.4 illustrates how Discriminant Analysis breaks down the original space into spaces which approximate identity and non-identity variation. It is particularly obvious that the pose of the faces is encapsulated in the non-identity space. The limitation of the partitioning is seen in the two lower images. For an ideal partitioning, we would expect the ‘identity’ images to be identical, since the images are of the same person. However, the large change in expression appears to have some affect on the system’s estimate of identity.

A further interesting effect is observed in the top two images (woman and old man). The narrow eyes of the old man are regarded as part of the identity, whilst

the wide eyes of the woman are regarded as part of the non-identity space. In fact, the old man’s eyes *do* appear narrow in all the training images, whereas the woman shows a variety of eyelid positions.

These remaining interactions between the identity and non-identity spaces can be addressed further during tracking; we present techniques for dealing with these effects in Chapters 7 and 8.

## 5.6 Expression model

As well as interpreting identity, we are interested in the automatic interpretation of expression. To this end, we have attempted to derive a subspace corresponding to variation in the facial expression. Moreover, in synthesis applications, one of the most useful ways to manipulate a face is to change its expression. We seek a subspace that allows the manipulation of expression independently of other types of variation.

A large subset of the images used to train the Appearance Model is provided with expression labels. The labels were provided by a panel of 25 observers who were asked to classify the faces into one of seven expressions, (*happy, sad, neutral, afraid, disgusted, surprised, angry*)\*. Naturally, such classification will result in a model of no greater than 6 dimensions. Figure 5.5 shows some typical examples from the set of images used for expression analysis.

Linear Discriminant Analysis produced a subspace model with 6 parameters. The effect of varying the first 3 parameters of the subspace vector for expression,  $\mathbf{d}_{expression}$  is shown in Figure 5.6. We can also visualise the corresponding ‘non-expression’ model. Figure 5.7 shows the effect of varying the first 3 parameters of the non-expression vector,  $\mathbf{r}_{non-expression}$ .

The non-expression modes show obvious change in pose and identity, whilst the

---

\*For this data we are grateful to Dr. Jane Whittaker, North Manchester Children’s Hospital

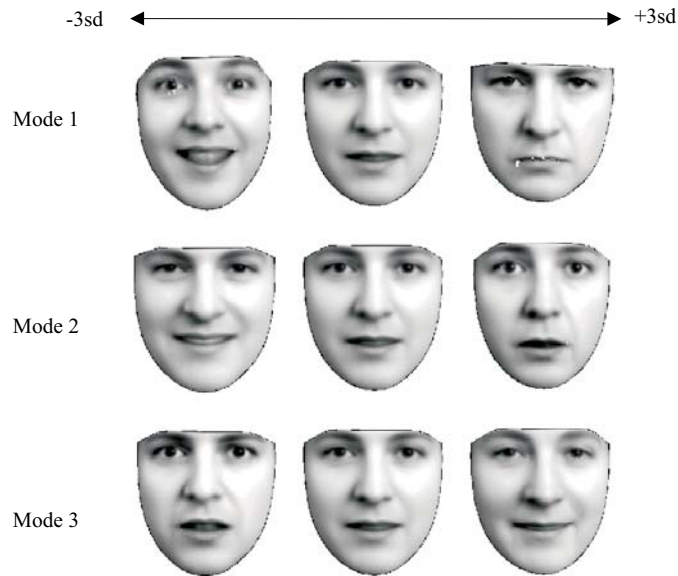


**Figure 5.5:** Training examples marked with expression labels.

expression appears fairly constant.

Discriminant Analysis is harder to apply to expression modelling than identity modelling, since there is a greater uncertainty in the subjective labels (we assume the identity labels are all perfect). By building an expression model in exactly the same way as the identity model (in other words, just changing the labelling scheme) misclassifications cause problems. Consider the calculation of a within-class covariance matrix for expression, where we try to encapsulate the ‘non-expression’ variation. Because of misclassification, there is bound to be some observed within-class variation which *is* due to expression. The effect of these problems can be seen in Figure 5.6 where there is noticeable identity variation encapsulated by the ‘expression’ parameters.

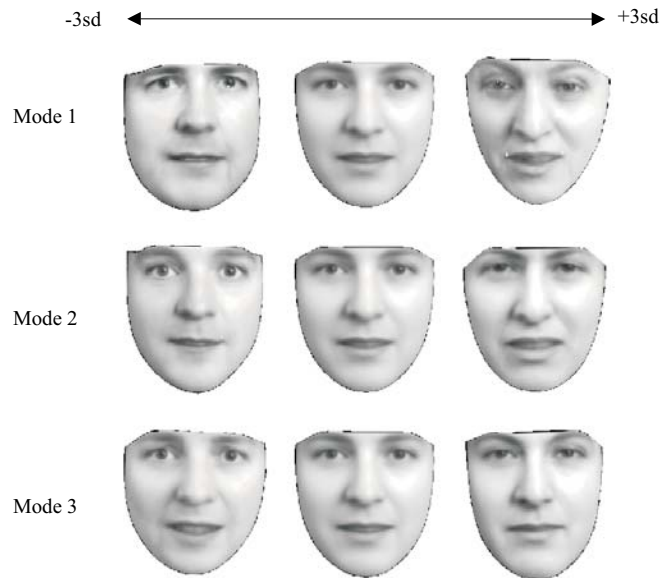




**Figure 5.6:** Effect of varying the first 3 parameters of the ‘expression’ subspace model built using LDA.

### 5.6.1 Projection onto expression subspaces

We can visualise the effect of projecting images onto the expression and non-expression subspaces calculated above. Figure 5.8 shows some original images with their reconstructions in the two spaces. This separation of expression and non-expression, whilst providing a reasonable first approximation is less effective than the separation of identity and non-identity. In particular, there appears to be some remaining identity variation in the expression space. This is probably due to the effect of subjective classification of the images. Particularly interesting is the apparent correlation of expression with pose as seen in the lower image. In fact, most of the people in the training set who show the expression ‘disgusted’, do tend to look downwards.



**Figure 5.7:** Effect of varying the first 3 parameters of the ‘non-expression’ subspace model built using LDA.

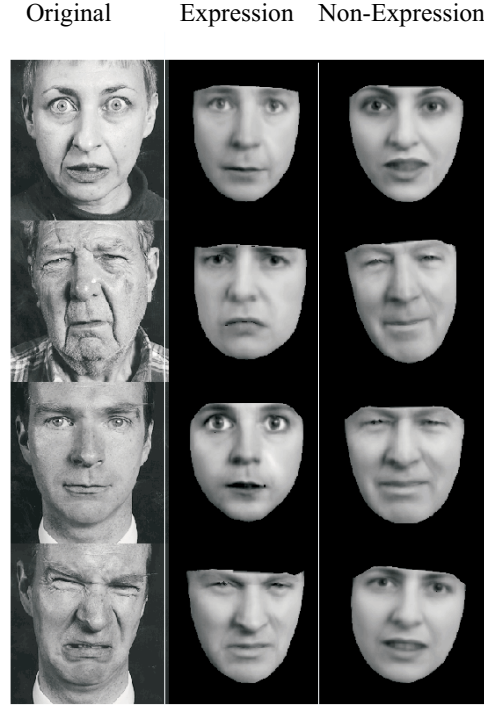
## 5.7 Face manipulation

An important property of Appearance Models is their generative nature. An Appearance Model is capable of generating synthetic images with close to photo-realistic quality. By varying the model parameters it is possible to change the appearance of a reconstructed face; the partitioned model parameters allow the manipulation of specific aspects of appearance. This is of potential value in animation<sup>†</sup>, and other forms of manipulation, for example, ‘photo-fit’ style forensic applications.

Given an input face, the first stage is to calculate the Appearance Model parameters,  $\mathbf{c}$ . These may be derived by analysis of images with hand-placed landmarks, or alternatively, automatically derived using Active Appearance Model search which is described in Chapter 6. The example is then projected onto the appropriate sub-

---

<sup>†</sup>This technology has recently been exploited by Createc Ltd., a special effects media company connected to the National Film and Television School.



**Figure 5.8:** Original images projected onto expression and non-expression subspaces respectively.

spaces to calculate the subspace parameters,  $\mathbf{d}$  and  $\mathbf{r}$ :

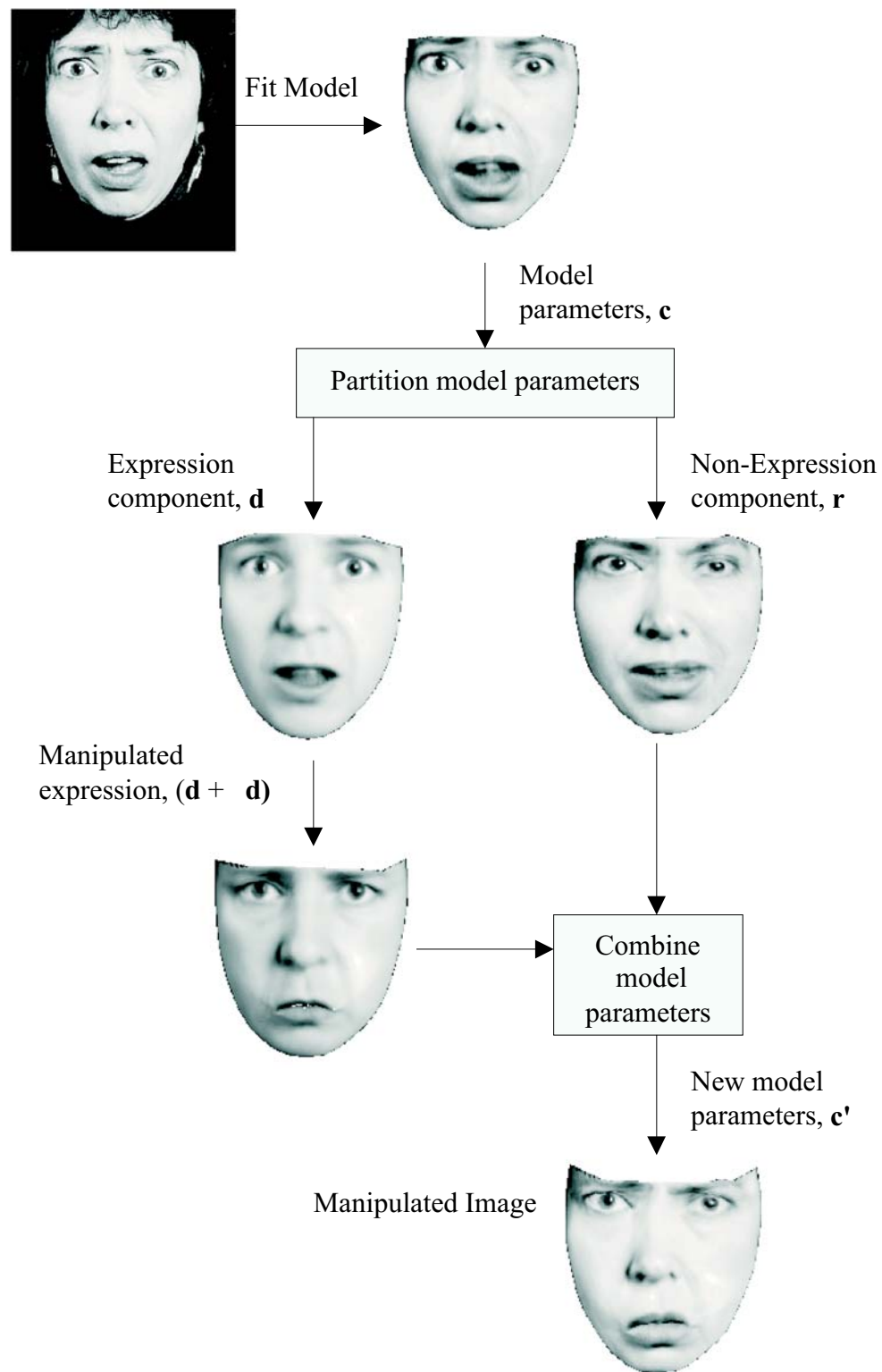
$$\mathbf{d} = \mathbf{D}^T \mathbf{c} \quad (5.11)$$

$$\mathbf{r} = \mathbf{R}^T \mathbf{c} \quad (5.12)$$

We can manipulate either the *Discriminant Parameters* or the *Residual Parameters* by the addition of a vector of required perturbations,  $\delta \mathbf{d}$  or  $\delta \mathbf{r}$  respectively. We can then regenerate a set of Appearance Model parameters,  $\mathbf{c}'$  according to:

$$\mathbf{c}' = \mathbf{D}(\mathbf{d} + \delta \mathbf{d}) + \mathbf{R}(\mathbf{r} + \delta \mathbf{r}) \quad (5.13)$$

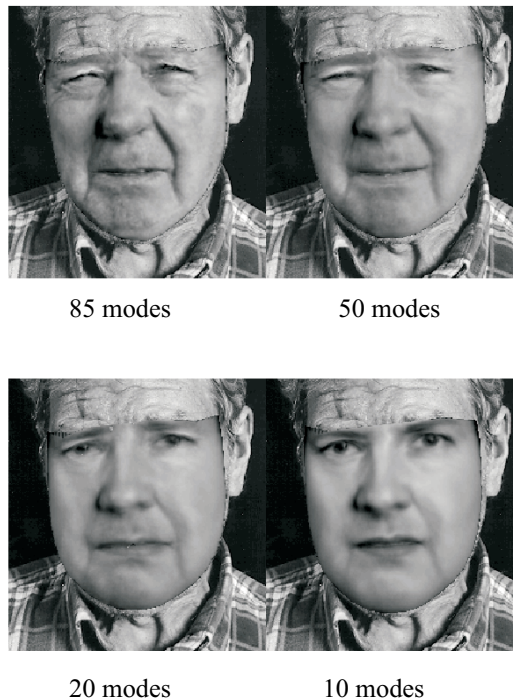
Usually one of either  $\delta\mathbf{d}$  or  $\delta\mathbf{r}$  will be set to zero, in order to restrict the manipulation to one type of variation. Given the new parameter vector,  $\mathbf{c}'$ , the new image can be reconstructed as described in Section 4.2. This procedure is shown schematically in Figure 5.9.



**Figure 5.9:** Schematic diagram of face manipulation method.

### 5.7.1 Retaining the integrity of fine texture

Principal Component Analysis captures the major sources of variation in the training set. Since small-scale details (such as freckles) occur in fairly random positions in different faces, their statistical significance can be indistinguishable from noise over the training set, and thus they are not captured by PCA. This tends to lead to a loss of fine texture in image reconstructions. Figure 5.10 illustrates this effect; we show reconstructions of training images, using progressively fewer model parameters. As we use less of the parameters that correspond to small eigenvalues, the faces retain their global characteristics but become less textured.



**Figure 5.10:** Images lose texture as they are reconstructed using fewer model parameters.

Whilst we can use the model to manipulate the appearance of faces, the synthetic

reconstructions lack fine texture. Visually, the fine texture is important, particularly for features such as facial hair. We have addressed this problem in reconstruction by treating the texture as a separate component that can be added or removed from a face in any configuration.

Recall from Chapter 4 that the linear nature of the model allows us to express the shape and grey-levels directly as functions of  $\mathbf{c}$

$$\mathbf{x} \approx \bar{\mathbf{x}} + \mathbf{P}_s \mathbf{W}_s \mathbf{Q}_s \mathbf{c} \quad , \quad \mathbf{g} \approx \bar{\mathbf{g}} + \mathbf{P}_g \mathbf{Q}_g \mathbf{c} \quad (5.14)$$

where

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q}_s \\ \mathbf{Q}_g \end{pmatrix} \quad (5.15)$$

After fitting the model to the image, the reconstructed grey-level vector  $\mathbf{g}$  will differ from the *actual* grey-levels in the image,  $\mathbf{g}'$  by a quantity  $\delta\mathbf{g}$ . This vector is stored as the ‘fine-texture’ of the face. After manipulating the image and thus generating a new grey-level vector, we then add back the fine-texture vector,  $\delta\mathbf{g}$ .

Figure 5.11 shows the effect of changing the expressions of faces, *without* the texture retaining step. In Figure 5.12 we show the same manipulation using the texture retaining step. The manipulation with added texture preserves the detail in the image.



**Figure 5.11:** Manipulating expression *without* retaining image texture.



**Figure 5.12:** Manipulating expression whilst retaining fine texture.

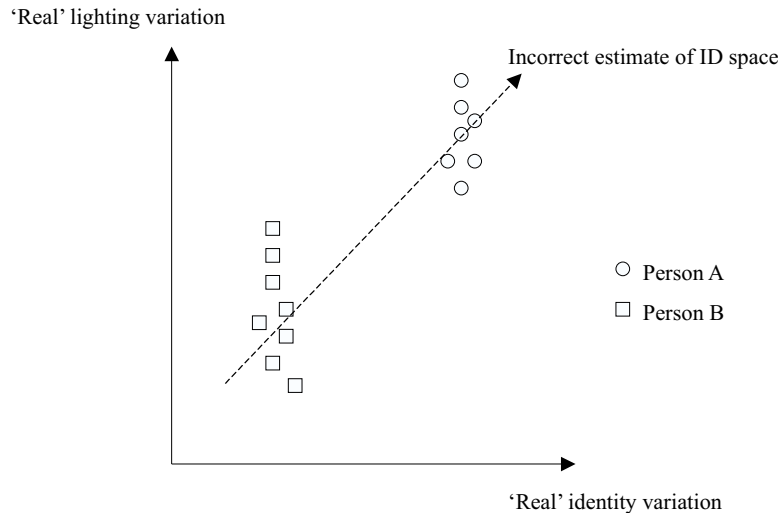


## 5.8 Alternatives to LDA

Linear Discriminant Analysis (LDA) requires that the training data falls into separate classes, and that the training data be accurately labelled. Whilst this is straightforward for identity, it is not obvious how the same approach could be used to build, for example, a lighting subspace.

Even in situations where labelling is natural, limited training data can introduce bias. Figure 5.13 illustrates this schematically.

We consider a hypothetical two-dimensional case in which we have just two people and one type of lighting variation. Unfortunately, although we have several training images of each individual, under various lighting conditions, the range of lighting observed *for each face* is very different, and does not even overlap. If we apply LDA in this case, the resulting discriminant function, although *optimal for the training data*, is clearly not correct. This is basically a problem of missing data. There is a smaller than correct contribution to the part of the within-class covariance matrix responsible for lighting and thus lighting would be interpreted as part of between-class variation.



**Figure 5.13:** Bias caused by poor training data.

Unfortunately it is very difficult to eliminate such bias from the training data, since we would have to ensure that each person was represented in an equal spread of conditions. This would be difficult even in principal and almost impossible in practice.

An alternative formulation is to build a model of purely within-class variation. In the case of identity, we know for certain that observed-within class spread is genuine; we can confidently label the identity of the training images, hence there is no other explanation of within-class spread. We might further hope that over all the training groups, we will observe many types of within-class variation, all making a contribution to the covariance matrix. The only assumption we make is that any within-class spread observed for a particular individual could be mapped onto another individual, i.e. that the space is *co-linear*. By this method we can produce a subspace which describes *non-identity* variation.

An important property of this method is that it gives a means of building models of lighting and pose without having to label the training data with such attributes. We take training data in which the pose and expressions of the images are fixed, with only the lighting varied. We could use the identity labels (which are known to be reliable) to estimate within-class spread. This within-class spread would represent lighting variation. The same analysis could be applied to calculate a subspace describing pose variation.

This method may be explored in the future to build explicit models of pose and lighting; currently we are restricted to a combination of pose and lighting encapsulated by the non-identity model. Other researchers [8] have shown that it is possible to model variation due to lighting change using a small number ( $<10$ ) of dimensions.

## 5.9 Summary

In this chapter we have described models which isolate specific sources of real-world variation. We have shown that Linear Discriminant Analysis leads to approximate solutions for identity and expression models. Approaches based on the analysis of within-class variation could be used to construct models of pose and lighting variation.

Given a *discriminant model*, it is possible to describe the residual subspace in a *residual model*. This is particularly useful in the case of identity, where we can create a model of all variation except identity. We show in later chapters how this proves useful for tracking.

Both the discriminant and residual models can be fitted to landmarked faces. By manipulating the model parameters, we can vary particular characteristics of the faces. For example, expression can be modified without changing identity. This provides a more useful image synthesis and manipulation tool than the Appearance Model alone.

The approach is limited by the assumption that the possible variation of individual faces is identical, and that factors such as pose and identity are linearly independent. A further limiting factor is the large amount of training data required for each type of variation to be modelled. More efficient and reliable methods for estimating the correct partitioning of the models are the subject of ongoing research [28].

In many applications, simply requiring that an image be a legal face is not enough, we must apply further rules. Subspace models are essentially a refinement of the Appearance Model's specificity, tailored for particular tasks.

# Chapter 6

## Active Appearance Models

In this chapter we introduce a new technique in model-based vision known as the *Active Appearance Model* or *AAM*. The AAM approach provides a means of using Appearance Models directly for image interpretation. The method was first introduced by Edwards *et al* [37] and described in more detail by Cootes *et al* [20].

We describe the motivation behind using Appearance Models for image search and discuss related approaches. The Active Appearance Model algorithm is then presented with demonstrations of its application. The recognition experiments described by Lanitis [61] are repeated using an AAM, and we show that the performance surpasses that obtained using the Active Shape Model approach, for images without occlusion. However, the AAM approach performs badly when there is occlusion.

### 6.1 Motivation

In Chapter 3 we described the method of Cootes *et al* [27] who used models of shape and local grey-level appearance in ASM search, to locate variable objects in new images. Lanitis *et al* [65] used this approach to interpret face images. First, face shapes were located using an ASM. The located face was then warped into a

normalised ‘shape-free’ frame. Parameters of the shape model and of a shape-free intensity model were used for interpretation.

In Chapter 4 we described how Edwards *et al* [39] extended this work to produce a combined model of shape and grey-level appearance. This model is more complete and specific than the separate models of shape and grey-level appearance, but there is no obvious way to use it directly to interpret images. Until recently, the only approach available was to use an ASM to locate face shapes in new images, and then to calculate the best-fit of the Appearance Model to the image region found. If the fit was found to be outside the legal range of the Appearance Model, the solution was rejected.

Ideally, having produced a full model of shape and grey-level appearance we should use this model directly for the interpretation task, achieving *interpretation by synthesis*. An outline of the approach is as follows:

1. *Given an image, find any region(s) of the image that the model can plausibly represent.*
2. *If such a plausible representation exists, use the model parameters to interpret the meaning of the region(s)*

The first point above emphasises the importance of the Appearance Model described in previous chapters. Because the model is *general*, we can be confident that if a face is present, the model can represent it. Crucially though, because the model is *specific*, we can be confident that if the model cannot represent a region, then that region is not a face.

## 6.2 Background

The task of model fitting can be regarded as a high-dimensional optimisation problem, in which we seek a set of model parameters that minimise the difference between the reconstructed image and the image data itself. Given that an effective model needs around 80 parameters, the task appears daunting.

### 6.2.1 Global optimisation

Jones and Poggio [55] have constructed models which are similar in principle to Appearance Models; they too have addressed the problem of matching high-dimensional models to images. Their experiments were based on a face model of effectively 63 dimensions. Using a stochastic gradient descent method they attempted to calculate the best values of the 63 parameters required to match unseen image data. The task was made easier by giving the model very good starting conditions in terms of pose, angle and scale. The results given were not particularly encouraging: The algorithm required 9 minutes to converge for a single image using fast SGI hardware - the resolution of the image was not reported, but appeared to be fairly low. A comprehensive assessment of the reliability was not given, although the authors suggest that local minima occasionally caused problems. We have also attempted to fit Appearance Models to images using similar optimisation techniques. We observed reasonable reliability given very good starting conditions but found, like Jones and Poggio, that the time required to find solutions was unacceptable.

### 6.2.2 Directed optimisation

The Active Shape Model algorithm provides a good example of a large optimisation problem made tractable by a directed search method. In the ASM case, a high-dimensional shape model is fitted to image data. Rather than repeatedly trying new

configurations driven by a scalar fit value, as in standard optimisation algorithms, the ASM algorithm uses measurements made at the current configuration to predict a better configuration. For a particular placement of the model, each landmark point *actively* searches a local region of the image for a better location. Active Appearance Models also attempt to solve Appearance Model fitting in a directed way.

### 6.2.3 Related work

The development of our new approach has benefited from insights provided by two earlier papers. Covell [30] uses a non-iterative technique for locating landmark points in images. In this approach, local region models around key landmark points are used to direct landmarks to the correct place. The AAM described here can be viewed in some respects as an extension of this idea for a full model of appearance.

Black and Yacoob [9] use local, hand crafted models of image flow to track facial features, but do not attempt to model the whole face. The AAM can be thought of as a generalisation of this method, in which the image difference patterns corresponding to changes in each model parameter are learnt and used to modify a model estimate.

In a parallel development, Sclaroff and Isidoro [83] have demonstrated ‘Active Blobs’ for tracking. They use image differences to drive tracking, learning the relationship between image error and parameter offset in an off-line processing stage. Active Blobs are derived from a single example, allowing deformations consistent with low energy mesh deformations (derived using a Finite Element method). A simple polynomial model is used to allow changes in intensity across the object. In contrast (see below), AAMs learn what are valid shape and intensity variations from their training set.

## 6.3 Active Appearance Model search

We now address the problem of matching an Appearance Model to image data. Given an image to be interpreted, an Appearance Model, and a reasonable starting approximation, we present an efficient scheme for adjusting the model location and model parameters, so that a new synthetic example is generated, which matches the image more closely. We begin by outlining the basic idea, before giving details of the algorithm.

### 6.3.1 Overview of AAM search

We wish to treat interpretation as an optimisation problem in which we minimise the difference between a new image and one synthesised by the appearance model. A difference vector  $\delta\mathbf{g}$  can be defined:

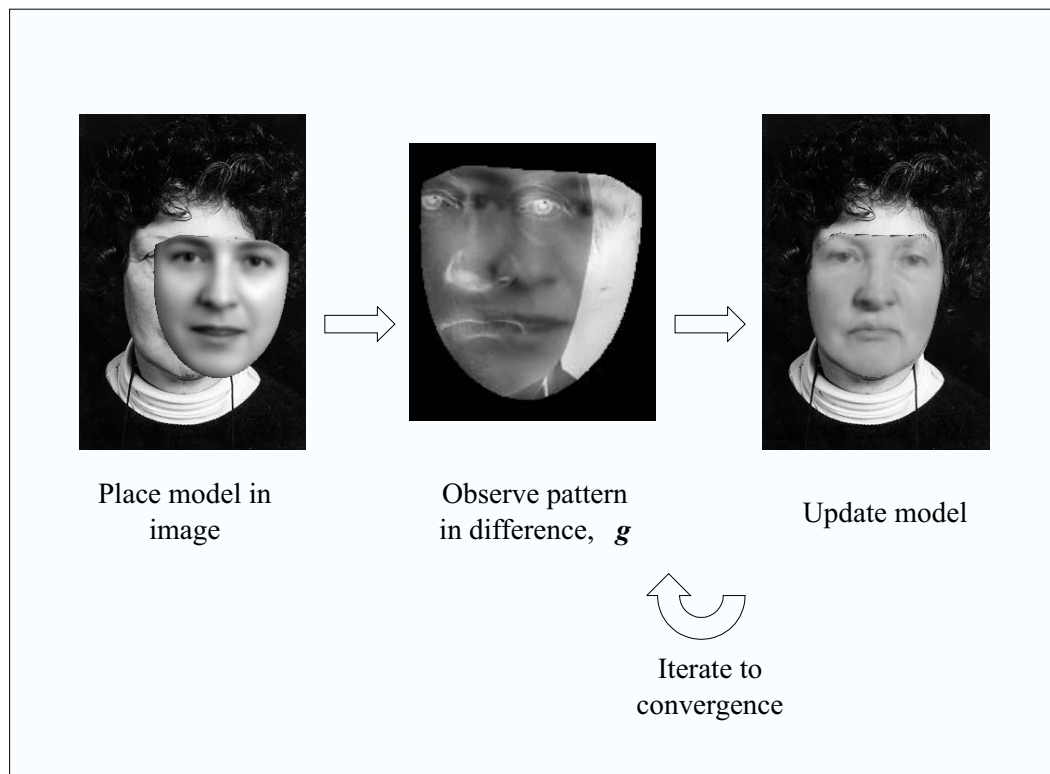
$$\delta\mathbf{g} = \mathbf{g}_i - \mathbf{g}_m \quad (6.1)$$

where  $\mathbf{g}_i$  is the vector of grey-level values sampled from the image, and  $\mathbf{g}_m$ , is the vector of grey-level values generated using the current model parameters.

To locate the best match between model and image, we wish to minimise the magnitude of the difference vector,  $|\delta\mathbf{g}|^2$ , by varying the model parameters,  $\mathbf{c}$  as defined in equation 4.7. For simplicity of notation we will assume that scale, translation and rotation parameters are included as elements of the vector  $\mathbf{c}$ . Since an Appearance Model typically has many parameters, this appears at first to be a difficult high-dimensional optimisation problem. We note, however, that since each attempt to match the model to a new image is actually a similar optimisation problem it is possible to learn something about how to solve this class of problems in advance. By providing *a-priori* knowledge of how to adjust the model parameters during image search, we arrive at an efficient run-time algorithm. In particular, the spatial pat-



tern in  $\delta\mathbf{g}$  encodes information about how the model parameters should be changed in order to achieve a better fit. There are two parts to the problem: learning the relationship between  $\delta\mathbf{g}$  and the error in the model parameters,  $\delta\mathbf{c}$ , and using this knowledge in an iterative algorithm for minimising  $|\delta\mathbf{g}|^2$ . This approach is illustrated in Figure 6.1.



**Figure 6.1:** Overview of AAM search scheme.

### 6.3.2 Learning to correct the model parameters

The simplest model we could choose for the relationship between  $\delta\mathbf{g}$  and the error in the model parameters,  $\delta\mathbf{c}$ , (and thus the correction which needs to be made) is linear:

$$\delta \mathbf{c} = \mathbf{A} \delta \mathbf{g} \quad (6.2)$$

This linear model turns out to be a good enough approximation. To find  $\mathbf{A}$ , we perform multivariate linear regression [54] on a sample of known model displacements,  $\delta \mathbf{c}$ , and corresponding difference images,  $\delta \mathbf{g}$ . We generate these sets of corresponding model and image errors by randomly perturbing the ‘true’ model parameters for images in which they are known. These can either be the original training images - or as in all the experiments described in this thesis - synthetic images generated by the Appearance Model itself. In the case of synthetic images the parameters are exactly known, and the images are not corrupted by noise. The only issue is what background to use - we have obtained good results using a white noise background with an intensity range matching that of the modelled image patch, though it would be worth investigating other possibilities.

As well as perturbations in the model parameters, we also model small displacements in 2D position, scale, and orientation. These four extra parameters are included in the regression, but for simplicity of notation, they can be regarded simply as extra elements of the vector  $\delta \mathbf{c}$ . To retain linearity we represent the pose using  $(s_x, s_y, t_x, t_y)$  where  $s_x = s \cos(\theta)$ ,  $s_y = s \sin(\theta)$ . In order to obtain a well-behaved relationship it is important to choose carefully the frame of reference in which the image difference is calculated. The most suitable frame of reference is the shape-normalised patch described in Chapter 4.

We calculate a difference thus. Let  $\mathbf{c}_0$  be the known appearance model parameters for the current image. We displace the parameters by a known amount,  $\delta \mathbf{c}$ , to obtain new parameters  $\mathbf{c} = \delta \mathbf{c} + \mathbf{c}_0$ . For these parameters we generate the shape,  $\mathbf{x}$ , and normalised grey-levels,  $\mathbf{g}_m$ , using (4.8). We sample from the image, warped using the points,  $\mathbf{x}$ , to obtain a new sample vector  $\mathbf{g}_s$ . The sample error is then  $\delta \mathbf{g} = \mathbf{g}_s - \mathbf{g}_m$ .

This process is repeated for many values of  $\delta\mathbf{c}$  and many images. Multivariate regression is performed to obtain  $\mathbf{A}$ , the matrix of coefficients expressing the approximate linear relationship between  $\delta\mathbf{g}$  and  $\delta\mathbf{c}$ .

The best range of values of  $\delta\mathbf{c}$  to use during training is determined experimentally. Ideally we seek to model a relationship that holds over as large a range of errors,  $\delta\mathbf{c}$ , as possible. However, the real relationship is found to be linear only over a limited range of values. Our experiments on the face model suggest that the optimum perturbation is around 0.5 standard deviations (over the training set) for each model parameter, about 10% in scale, 15 degrees in angle, and 10 pixels in x and y translation.

A key difference between Active Appearance Models and the Active Blob approach of Sclaroff and Isidoro [83] is the way in which the relationship between displacement and image difference is estimated. In the Active Blob approach the relationship is estimated by assuming that orthogonal displacements will produce orthogonal image differences. In our approach we do not make this assumption, replacing a pseudo-inverse calculation with a full linear regression method using generated training data.

### 6.3.3 Regression results for the face model

We applied the method described above to the face Appearance Model described in Section 4.3. After performing linear regression, we calculated the  $R^2$  statistic [54] for each parameter perturbation,  $\delta c_i$  to measure how well the displacement  $\delta\mathbf{c}$  was ‘predicted’ by the error vector  $\delta\mathbf{g}$ . The average  $R^2$  value for the 80 parameters was 0.82, with a maximum of 0.98 (the 1st parameter) and a minimum of 0.48. This suggests a reasonably linear relationship.

### 6.3.4 Iterative model refinement

Given a method for predicting the correction that needs to be made in the model parameters we can straightforwardly construct an iterative method for solving our optimisation problem.

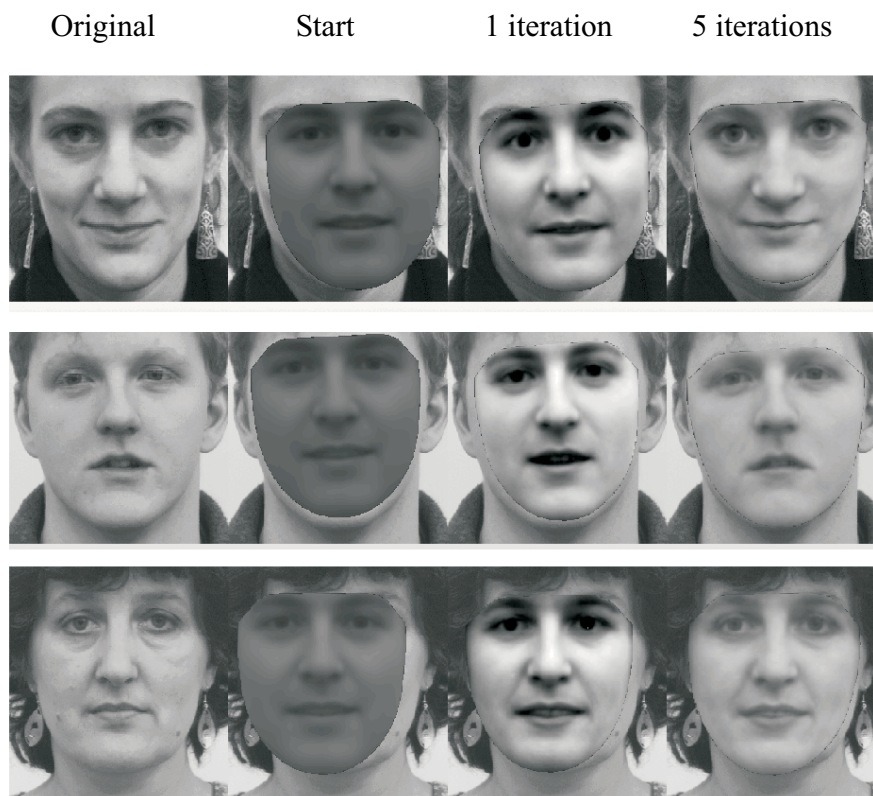
Given the current estimate of model parameters,  $\mathbf{c}$ , and the image sample at the current estimate,  $\mathbf{g}_s$ , one step of the iterative procedure is as follows:

- evaluate the error vector  $\delta\mathbf{g} = \mathbf{g}_s - \mathbf{g}_m$
- evaluate the current error  $E = |\delta\mathbf{g}|^2$
- Compute the predicted displacement,  $\delta\mathbf{c} = \mathbf{A}\delta\mathbf{g}$
- set  $k = 1$
- let  $\mathbf{c}' = \mathbf{c} - k\delta\mathbf{c}$
- sample the image at this new prediction, and calculate a new error vector,  $\delta\mathbf{g}'$
- if  $|\delta\mathbf{g}'|^2 < E$  then accept the new estimate,  $\mathbf{c}'$ ,
- otherwise try at  $k = 0.5$ ,  $k = 0.25$  etc.

This procedure is repeated until no improvement is made to the error,  $|\delta\mathbf{g}|^2$ , and convergence is declared.

## 6.4 Examples

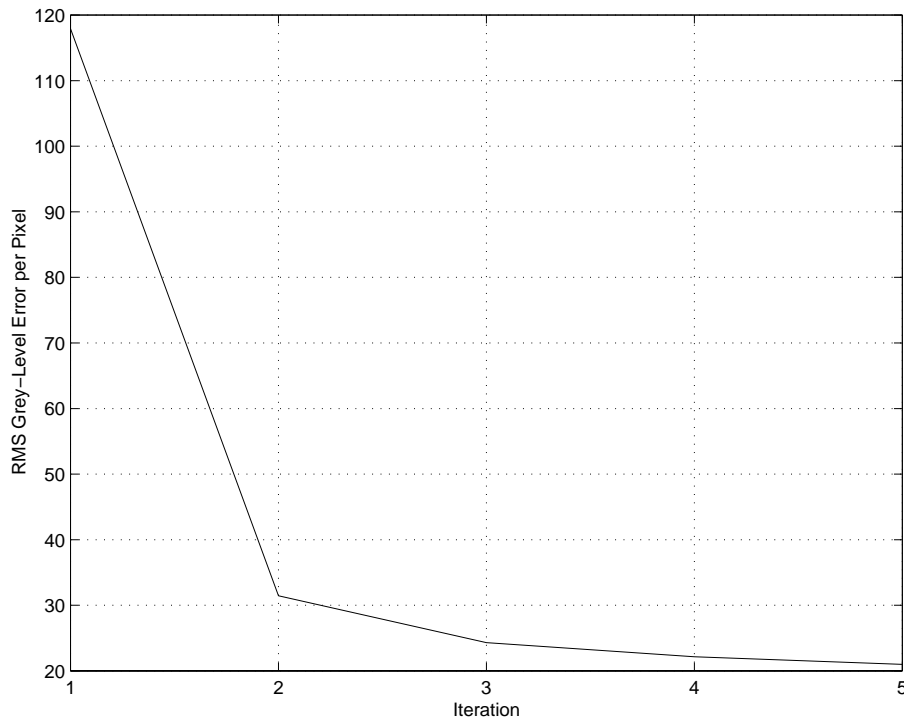
The AAM algorithm works well if given a reasonably good starting approximation. Figure 6.2 illustrates several examples of AAM search. Figure 6.3 shows a graph of the RMS value of grey-level error per pixel versus iteration during a typical search. Our current implementation takes approximately 150ms per iteration on a Pentium II 450 MHz processor.



**Figure 6.2:** Examples of AAM search. Original image on left. Iterations 1,2,5 shown on right.

## 6.5 AAM search versus hand-fitting

As is shown in Figure 4.3 in Chapter 4 the ‘best-fit’ of an Appearance Model to unseen data can be unsatisfactory. This can be caused by an over-dependence on the shape as given by the landmark points. If the landmarks are badly placed, or if the shape model does not fit the landmarks very well, the resulting grey-level sample vector can fall outside the range of variability learnt by the model. As a result the ‘best-fit’ of the grey-level model can appear poor. The Active Appearance Model on the other hand, is concerned only with minimising the grey-scale difference between pixels of the model and pixels in the image. In order to do this, we expect that the shape should be similar to the positions of the hand-placed landmarks, but they are not forced to be as near as possible, as in direct reconstruction. By allowing the



**Figure 6.3:** Typical search performance. RMS value of grey-level error per pixel is shown as a function of iteration number. Image grey-levels are in the range 0-255.

shape to ‘relax’ the perceived fit can be improved. This effect was shown in Figure 4.4. In some cases, the Active Appearance Model can be used to find a ‘better’ set of landmark points than a human operator. We show an example of this in Section 9.2.

## 6.6 Comparison with ASM-based recognition

As a preliminary assessment of Active Appearance Models, we performed the same recognition tests as described by Lanitis [61] and summarised in Section 3.4.2. We performed AAM search on the training set and test images, recording the resulting model parameters for each image. A human operator initialised the search in each image by locating the centre of the left eye. Using the recovered model parameters we calculated recognition rates in the same way as Lanitis [61]. The results, along

with those of Lanitis are given in Table 6.1.

	Normal test set		Difficult test set	
	Correct	Within 3	Correct	Within 3
Shape model	50.3%	66.6%	15.6%	31.1%
Shape-free grey model	78.7%	87.3%	31.1%	53.3%
Local grey-level models	77.3%	89.7%	28.9%	57.8%
Shape + shape-free models	85.3%	93.3%	34.4%	56.7%
Shape + local models	80.0%	90.3%	34.4%	66.7%
All methods	92.0%	97.0%	48.9%	77.4%
<b>Active Appearance Model</b>	<b>97.5%</b>	<b>97.5%</b>	<b>12.9%</b>	<b>25.2%</b>

**Table 6.1:** Classification results using Active Appearance Model versus Active Shape Model.

The results show that the AAM method performs better than any of the ASM-based methods for the ‘easy’ test images, but performs very badly on the ‘difficult’ set. This is because, unlike the ASM, the AAM currently has no means of dealing with occlusion and the search failed to converge in almost all cases where significant occlusion was present.

We present a more extensive evaluation of recognition performance in Chapter 8.

## 6.7 Summary

In this chapter we have described a novel approach to fitting high-dimensional models to image data. The Active Appearance Model is an iterative, directed search method that uses measurements made at each current estimated solution to drive the model towards better solutions. The training algorithm uses multivariate regression to learn the relationship between offsets in the model parameters (and pose) and the patterns in the grey-level difference vector between model and image.

AAM search is fast, converging in a few iterations, typically taking less than 1 second. This compares well with standard optimisation techniques which can take several minutes (even without any pose offset). The iterative nature of the search method makes AAMs ideal for tracking; at each frame in a sequence the AAM is likely to be close to the new solution, and should converge extremely quickly, perhaps requiring only one iteration.

A key limitation of the Active Appearance Model is the need for good positional initialisation; in our experiments we found that the centre of the model needed to be placed within about 20 pixels of the correct location for reliable convergence. A further drawback to the AAM method is the inability to deal with occlusion.



# Chapter 7

## Tracking Faces

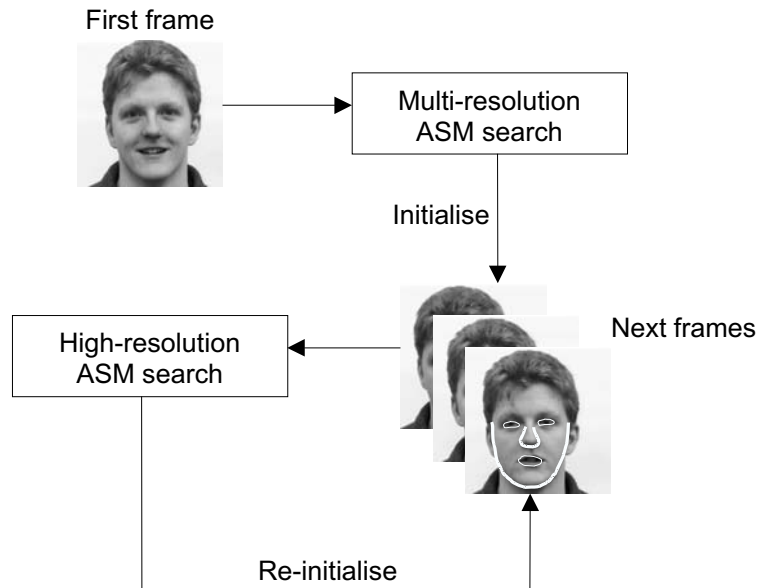
This chapter presents a novel scheme for tracking faces, based on Active Appearance Models. We begin by reviewing model-based tracking of objects using Active Shape Models, describing the ASM tracking scheme of Lanitis [61], and Kalman filter-based schemes such as those described by Baumberg *et al* [6] [5] and Blake and Isard [11]. Our tracking scheme is based on decoupling the sources of appearance variation into identity and non-identity parts. In particular, we utilise the fact that identity should remain fixed whilst tracking a given individual. Using this knowledge, we develop a method capable of refining the identity/non-identity decoupling automatically during tracking. We demonstrate that this refinement allows a stable estimate of identity to be obtained without significant degradation in tracking accuracy.

### 7.1 Simple tracking using ASMs

Since ASMs can be used to locate objects rapidly in individual images, tracking objects through video sequences is a natural application. The iterative nature of ASMs makes them ideal for this task, given that there are usually only small image changes between frames. Once tracking is underway, the ASM can be initialised in

each frame at the position found in the previous frame. Usually very few iterations are then required to reach convergence in the new frame.

This scheme was used by Cootes *et al* [26] for tracking sequences of the left ventricle in echo-cardiograms. Lanitis [61] used the same scheme for tracking faces in video sequences. Multi-resolution search is used to initialise the model in the first frame, after which the search is performed at only the highest resolution level. This is illustrated in Figure 7.1.



**Figure 7.1:** Illustration of Lanitis' simple tracking scheme.

## 7.2 Kalman filtering

Kalman filtering is an established technique for optimal tracking of discrete processes, and is often used in computer vision where measurements on video data are made at discrete time intervals. Baumberg *et al* [6] have applied Kalman filtering to ASM tracking and Blake *et al* [10] to similar contour tracking methods. We do not intend to give a full description of Kalman Filter theory, but an overview of the most important

points will be necessary. The reader is referred to the reference texts by Gelb [44] and Brown and Huang [15] for more detail.

### 7.2.1 Basic theory

Tracking involves estimating the state of a system at a series of time steps based on measurements made on the system. In most tracking applications there exists some prior knowledge of the system dynamics - though there is always at least one non-deterministic term (otherwise the dynamic model would completely define the track in advance). The key components of a Kalman filter are a vector model of the pseudo-random dynamic *process* and a recursive algorithm for processing noisy *measurements* of the state of the system. The Kalman Filter provides a *least-squares optimal estimate* of the state of a dynamic system given a discrete process model and noisy measurements.

The Kalman Filter models the system in terms of a *state vector*,  $\mathbf{x}$ . Regardless of how time discretisation arises in the physical world, the following formulation is adopted:

$$\mathbf{x}_{k+1} = \phi_k \mathbf{x}_k + \mathbf{w}_k \quad (7.1)$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \quad (7.2)$$

where

$\mathbf{x}_k = (n \times 1)$  state vector at time  $t_k$

$\phi_k = (n \times n)$  process update matrix relating  $\mathbf{x}_{k+1}$  to  $\mathbf{x}_k$ .

$\mathbf{w}_k = (n \times 1)$  vector of white process noise.

$\mathbf{z}_k = (m \times 1)$  measurement vector at time  $t_k$ .

$\mathbf{H}_k = (m \times n)$  matrix giving the relationship between measured vector and the state

vector at time  $t_k$ .

$\mathbf{v}_k = (m \times 1)$  vector of measurements errors.

It is assumed that the covariance matrices for the vectors  $\mathbf{w}_k$  and  $\mathbf{v}_k$  are known (or can be estimated). It is also assumed that the vectors,  $\mathbf{w}_k$  and  $\mathbf{v}_k$  are temporally uncorrelated. The covariance matrices are given by:

$$E[\mathbf{w}_k \mathbf{w}_i^T] = \begin{cases} \mathbf{Q}_k, & i = k \\ 0, & i \neq k \end{cases} \quad (7.3)$$

$$E[\mathbf{v}_k \mathbf{v}_i^T] = \begin{cases} \mathbf{R}_k, & i = k \\ 0, & i \neq k \end{cases} \quad (7.4)$$

$$E[\mathbf{w}_k \mathbf{v}_i^T] = 0 \quad (7.5)$$

### 7.2.2 Example model

Imagine a 1-D process in which a scalar value,  $x$ , follows an *integrated random walk*. In this model,  $x$  moves between frames with a velocity,  $\dot{x}$ . The velocity itself changes between frames by the addition of white Gaussian noise.

The state vector,  $\mathbf{x}$ , and the state transition matrix,  $\phi$ , for this process are given by:

$$\mathbf{x} = \begin{pmatrix} x \\ \dot{x} \end{pmatrix} \quad (7.6)$$

$$\phi_i = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad (7.7)$$

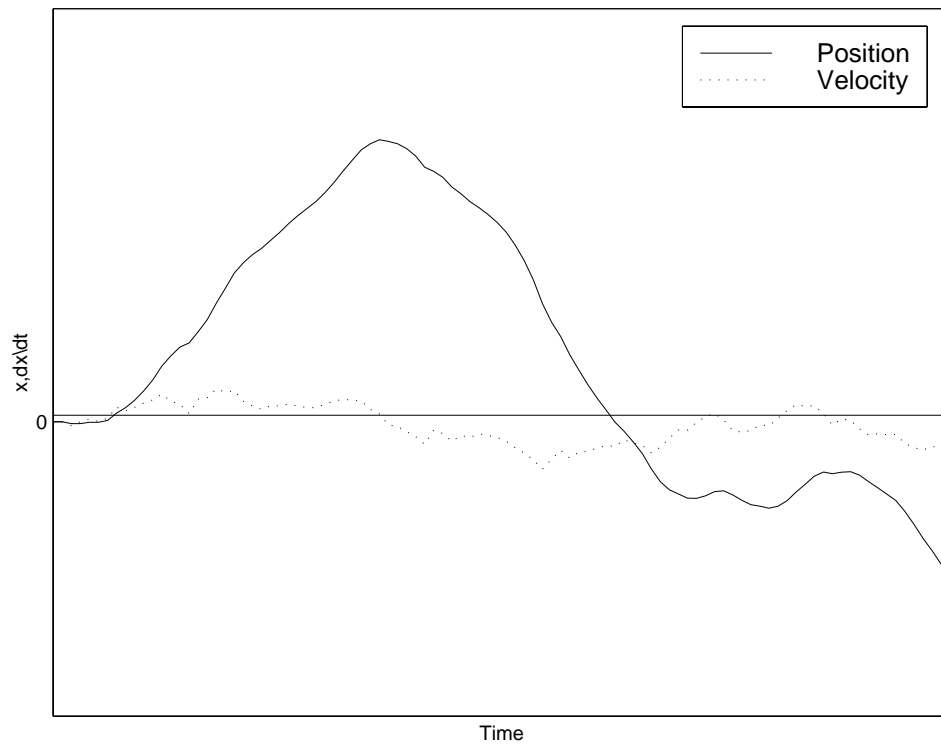
thus, Equation 7.1 becomes:

$$\begin{pmatrix} x_{k+1} \\ \dot{x}_{k+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_k \\ \dot{x}_k \end{pmatrix} + \begin{pmatrix} 0 \\ u_k \end{pmatrix} \quad (7.8)$$

where  $u_k$  is a Gaussian white noise sequence. Let us assume that the only measurement,  $z_k$ , we can make on the system is of the position,  $x$ , corrupted by Gaussian noise,  $v_k$ . Equation 7.2 thus becomes:

$$z_k = x_k + v_k \quad (7.9)$$

An example of an integrated random walk sequence is shown in Figure 7.2. Notice how the position changes direction as the velocity crosses zero.



**Figure 7.2:** Example of a 1-dimensional integrated random-walk.

### 7.2.3 Kalman update procedure

A full derivation of the Kalman Filter update equations is given by Brown and Huang [15]; here we give the important results. The Kalman filter aims to provide an estimate  $\hat{\mathbf{x}}_k$  of the state vector at time  $t_k$ . At time  $t_k$  we already have an *a priori* estimate of the state vector,  $\hat{\mathbf{x}}_k^-$ . We also have an estimate of the error covariance,  $\hat{\mathbf{P}}_k^-$  associated with  $\hat{\mathbf{x}}_k^-$ , where:

$$\hat{\mathbf{P}}_k^- = E[(\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k^-)(\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k^-)^T] \quad (7.10)$$

Given a new measurement,  $\mathbf{z}_k$ , the new estimate,  $\hat{\mathbf{x}}_k$  is calculated as a linear combination of the noisy measurement and the *a priori* estimate according to:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-) \quad (7.11)$$

where  $\mathbf{K}_k$ , gives a weighting between the measurement and *a priori* estimate, and is known as the *Kalman Gain*. The Kalman Gain is the optimal weighting factor in the least-squares sense, and can be calculated by:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (7.12)$$

The current best estimate of the estimation error covariance,  $\mathbf{P}_k$  is given by:

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \quad (7.13)$$

The Kalman filter procedure is completed by projecting the estimates of  $\mathbf{x}_k$  and  $\mathbf{P}_k$  forward to give *a priori* estimates at time  $t_{k+1}$ , according to:

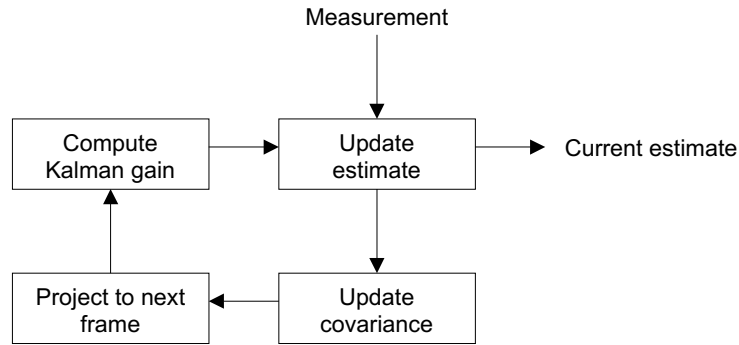
$$\mathbf{x}_{k+1}^- = \phi_k \hat{\mathbf{x}}_k \quad (7.14)$$

$$\mathbf{P}_{k+1}^- = \phi_k \mathbf{P}_k \phi_k^T + \mathbf{Q}_k \quad (7.15)$$

The full Kalman filter algorithm can be summarised thus:

1. Initialise with *a priori* estimates of  $\mathbf{x}_k^-$  and  $\mathbf{P}_k^-$ .
2. Compute Kalman gain,  $\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$
3. Update with new measurement:  $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-)$
4. Update error covariance,  $\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-$
5. Project ahead to  $t_{k+1}$ :  $\mathbf{x}_{k+1}^- = \phi_k \hat{\mathbf{x}}_k$ , and  $\mathbf{P}_{k+1}^- = \phi_k \mathbf{P}_k \phi_k^T + \mathbf{Q}_k$
6. Return to step 2.

Figure 7.3 illustrates the Kalman filter algorithm schematically.



**Figure 7.3:** Schematic diagram of Kalman filter algorithm.

## 7.3 Filtered ASM tracking

Baumberg [5] describes the use of ASMs with Kalman filtering to track sequences of moving people. An ASM is built from training images of moving people, from which the moving ‘blob’ representing the walking individual is first extracted using simple image processing. Recall the analysis of Section 3.1 in which the vector  $\mathbf{x}$  of  $N$  points

defining the shape is given by:

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b} \quad (7.16)$$

where  $\mathbf{b}$  is vector of model parameters and  $\mathbf{P}$  a matrix of orthogonal eigenvectors.

A shape,  $\mathbf{x}$ , is projected into the image by scaling, rotation and translation using:

$$\begin{pmatrix} X_i \\ Y_i \end{pmatrix} = \mathbf{M} \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad (7.17)$$

where  $X_i$  and  $Y_i$  are the projection of the control points,  $x_i$  and  $y_i$  into the the image. The transformation consists of a translation by  $t_x$  and  $t_y$  and scaling/rotation given by  $\mathbf{M}$ :

$$\mathbf{M} = \begin{pmatrix} a_x & -a_y \\ a_y & a_x \end{pmatrix} = \begin{pmatrix} s \cos \theta & -s \sin \theta \\ s \sin \theta & s \cos \theta \end{pmatrix} \quad (7.18)$$

In this formulation, the shape vector, when projected into the image frame, is given by:

$$\mathbf{X} = \mathbf{Q}(t_x, t_y)(\bar{\mathbf{x}} + \mathbf{P}\mathbf{b}) + \mathbf{t} \quad (7.19)$$

where

$$\mathbf{t} = (t_x, t_y, \dots, t_x, t_y)^T \quad (1 \times 2N) \quad (7.20)$$

$$\mathbf{Q} = \begin{pmatrix} \mathbf{M} & 0 \\ & \ddots \\ 0 & \mathbf{M} \end{pmatrix} \quad (2N \times 2N) \quad (7.21)$$

The ASM tracking system must track the shape parameters,  $\mathbf{b}$ , the alignment parameters,  $(a_x, a_y)$ , and the translation parameters,  $(t_x, t_y)$ . These are incorporated into a



Kalman filter framework using suitable dynamic models.

### 7.3.1 Dynamic models

The person-tracking system proposed by Baumberg [5] is primarily concerned with tracking individuals walking across a scene. Baumberg's system regarded the origin of the model as undergoing uniform 2D motion with additive random noise in both velocity and acceleration. The alignment parameters were assumed to be constant with added noise.

Baumberg's tracking system assumes that the shape parameters,  $b_i$ , vary independently in time. This assumption was made because over the training set:

$$E(b_i b_j) = 0 \quad i \neq j \quad (7.22)$$

This allows the shape-parameters to be tracked with a bank of independent 1-D Kalman filters with the state update equation taking the simple form:

$$b_i^{(k+1)} = b_i^{(k)} + w_i^{(k)} \quad (7.23)$$

where  $w_i^{(k)}$  is taken from a Gaussian noise sequence of zero mean and variance  $\mu$ .

### 7.3.2 Discussion

Baumberg's person tracking system works on the assumption that the shape parameters vary independently. Baumberg [5] and Baumberg and Hogg [4] describe automatic methods for building models for which this assumption is valid. Their motivation is similar to ours - to build specific models capable only of generating legal variation. They address the problem of generating models with legal spatiotemporal dynamics by training the system on image sequences. In this thesis, we describe face

tracking using an Active Appearance Model. We propose a method that does not rely on image sequences during training, but instead uses the prior knowledge provided by the Partitioned Model.

In our face Appearance Model, each model parameter represents a combination of inter-personal and intra-personal variation. Indeed, we showed in Chapter 5 that ID and non-ID variation could be represented as *linear combinations* of the original model parameters. With this knowledge, we see that, in our case, the assumption of dynamic independence of the model parameters is invalid. We know that, in order to produce a pure non-ID variation (i.e. the variation needed to track a single person), we must manipulate several model parameters simultaneously.

The independent one-dimensional filters used by Baumberg are highly attractive due their computational efficiency and the prior knowledge of variance available from training. In the absence of decoupled dynamics, setting up an appropriate state-space model would be extremely difficult. Fortunately, the partitioned appearance model described in Chapter 5 provides a framework in which we can justifiably use a set of independent one-dimensional filters. Varying the parameters of the non-ID model should not change identity. So, whilst in any particular sequence we might observe incidental correlation between the non-ID parameters, we know that correlated variation is not *essential* in order to track an individual, as would be the case with the raw Appearance Model parameters.

## 7.4 Tracking using a Partitioned AAM

In this section we introduce the use of Active Appearance Models and Partitioned Models for face tracking, demonstrating a method of using the knowledge that identity must be fixed during a sequence. This approach was first proposed by Edwards *et al* [39] in an ASM framework [39] [40] [37]. Here the work is extended to the Active Appearance Model framework.

### 7.4.1 Motivation

This thesis has repeatedly argued the need for models with high specificity. We note that during tracking, the basic Active Appearance Model lacks specificity due to its ability to change the apparent identity. The problem lies in the fact that we wish to use the same model to fit to *any* person, which means that it must be able to represent inter-personal variation. However, once tracking a particular individual, inter-personal variation should be forbidden. Fortunately, the Kalman Filter framework provides a means of representing this knowledge in terms of simple dynamic models, by applying separate models to the identity and non-identity parts of the Partitioned Active Appearance Model.

### 7.4.2 Overview

Active Appearance Models, like ASMs, are attractive for tracking due to their iterative nature. Given a reasonable starting position, just one iteration of an AAM is often sufficient to make reasonable parameter adjustments to match the image. By incorporating Kalman filtering, we attempt to make optimal use of the data available from an image sequence.

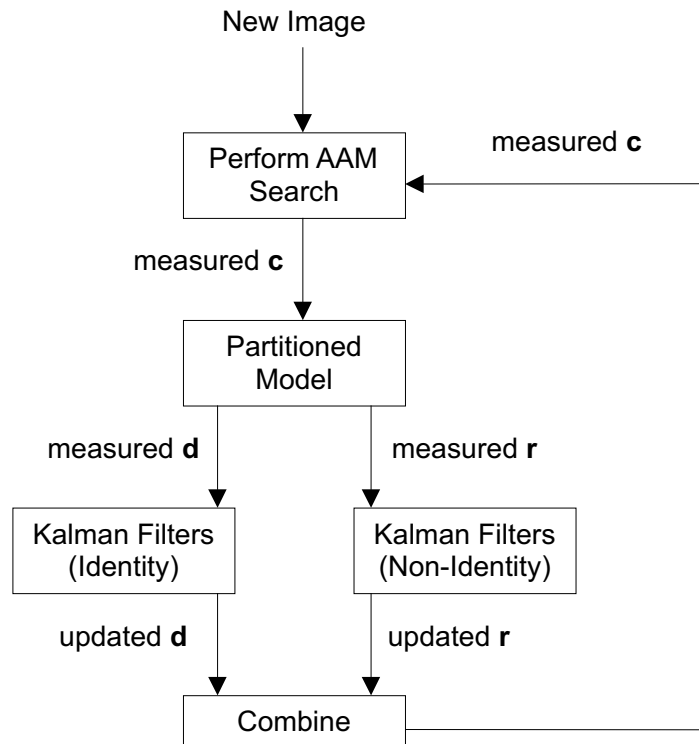
In order to use independent parameter filters, we must first find a basis in which the assumption of independence is valid. To a first approximation, partitioning the Appearance Model into identity and non-identity subspaces provides this basis. We can use the AAM to provide a measurement of the full model parameter vector,  $\mathbf{c}$ . This vector is then used to estimate the identity parameters,  $\mathbf{d}$ , and the non-identity parameters,  $\mathbf{r}$ . In addition we also track the scale, rotation and translation similarly to Baumberg [5].

The set of parameters we wish to track is thus:

1. Alignment parameters,  $(a_x, a_y)$

2. Translation parameters,  $(t_x, t_y)$
3. Identity parameters,  $d_i$
4. Non-identity parameters,  $r_i$

The basic idea of the filtered tracking scheme is shown diagrammatically in Figure 7.4. Two separate banks of one-dimensional Kalman filters are used - the dynamic structure of the identity filters is different to that of the non-identity filters.



**Figure 7.4:** Schematic diagram of decoupled and filtered tracking algorithm.

### 7.4.3 Tracking translation, scale and orientation

The dynamic model used by Baumberg [5] for tracking the translation parameters, was appropriate in a system where the usual type of motion was a fairly smooth path

across the image. The test sequences we have used for evaluating face tracking are of people talking to a camera whilst moving their heads in a quasi-random fashion. This choice of test sequence was made to reflect the type of variation expected in many application situations; e.g. a person interacting with a computer screen, standing at an ATM, or acting in front of a camera. This type of random motion means we cannot apply a dynamic model with significant deterministic components. The integrated random walk model effectively says that, in the default situation, the velocity will be unchanged between frames, unless altered by a stochastic acceleration. A simple random-walk model says that by default, the position remains fixed, unless altered by a stochastic velocity. The random-walk model is a *first order model* or *position model*, the integrated random walk a *second order model* or *position-velocity(PV) model*. Generally, the higher the order of a model the more susceptible it becomes to instability and divergence, due the effect of the higher-derivative components. In the experiments presented in this thesis, a second-order model was applied to the translation parameters, giving the following update equations:

$$\begin{pmatrix} t_x^{k+1} \\ \dot{t}_x^{k+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} t_x^k \\ \dot{t}_x^k \end{pmatrix} + \begin{pmatrix} 0 \\ u_x \end{pmatrix} \quad (7.24)$$

$$\begin{pmatrix} t_y^{k+1} \\ \dot{t}_y^{k+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} t_y^k \\ \dot{t}_y^k \end{pmatrix} + \begin{pmatrix} 0 \\ u_y \end{pmatrix} \quad (7.25)$$

These equations can be written as a single state update equation:

$$\begin{pmatrix} t_x^{k+1} \\ \dot{t}_x^{k+1} \\ t_y^{k+1} \\ \dot{t}_y^{k+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} t_x^k \\ \dot{t}_x^k \\ t_y^k \\ \dot{t}_y^k \end{pmatrix} + \begin{pmatrix} 0 \\ u_x \\ 0 \\ u_y \end{pmatrix} \quad (7.26)$$

The process noise covariance matrix for a position-velocity model is given by (see Brown and Huang [15]):

$$\mathbf{Q} = \begin{pmatrix} \frac{S_x}{3} & \frac{S_x}{2} & 0 & 0 \\ \frac{S_x}{2} & S_x & 0 & 0 \\ 0 & 0 & \frac{S_y}{3} & \frac{S_y}{2} \\ 0 & 0 & \frac{S_y}{2} & S_y \end{pmatrix} \quad (7.27)$$

where  $S_x$  and  $S_y$  are the respective spectral amplitudes of the white noise functions,  $u_x$  and  $u_y$ . In the equations given, all distances are assumed to be measured in pixels, and time measured in number of frames. The estimated values for  $S_x$  and  $S_y$  must be chosen to scale along with image size and frame rate.

The alignment parameters,  $a_x$  and  $a_y$  are assumed to move between frames with a random velocity, in other words, to follow a random-walk. The two parameters are grouped to give the combined update equation:

$$\begin{pmatrix} a_x^{k+1} \\ a_y^{k+1} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} a_x^k \\ a_y^k \end{pmatrix} + \begin{pmatrix} u_{ax} \\ u_{ay} \end{pmatrix} \quad (7.28)$$

The process noise covariance matrix is simply:

$$\mathbf{Q} = \begin{pmatrix} S_{ax} & 0 \\ 0 & S_{ay} \end{pmatrix} \quad (7.29)$$

where  $S_{ax}$  and  $S_{ay}$  are the respective spectral amplitudes of the white noise functions,  $u_{ax}$  and  $u_{ay}$ .

#### 7.4.4 Tracking the model parameters

In the tracking scheme illustrated in Figure 7.4 we treat the identity parameters as essentially fixed; the AAM is regarded as making noisy measurements of a system

with constant value. The model parameters are treated independently and tracked with individual 1-D filters. The update equation for a particular identity parameter,  $d_i$  is simply:

$$d_i = \text{constant} \quad (7.30)$$

The initial filter parameters,  $Q_i$  and  $P_{0_i}^-$  are given by:

$$Q_i = 0 \quad P_{0_i}^- = V_{d_i} \quad (7.31)$$

where  $V_{d_i}$  is the estimated initial measurement noise in the identity parameter  $d_i$ . Typically, we chose the initial value of  $V_{d_i}$  to be 3 times the standard deviation of the corresponding model parameter, a value which was found to give good performance.

We treat the residual, non-identity parameters as following a random-walk. The update equation is thus:

$$r_i^{k+1} = r_i^k + u_{r_i} \quad (7.32)$$

with the filter parameters,  $Q_i$  and  $P_{0_i}^-$  given by:

$$Q_i = S_{r_i} \quad P_{0_i}^- = V_{d_i} \quad (7.33)$$

where  $S_{r_i}$  is the estimate spectral noise amplitude of the random component. This quantity can be estimated by observation of typical sequences. When estimating process noise amplitudes, a generous estimate will usually result in a more stable tracking system, at the expense of absolute tracking accuracy.

## 7.5 Limitations of decoupled AAM tracking

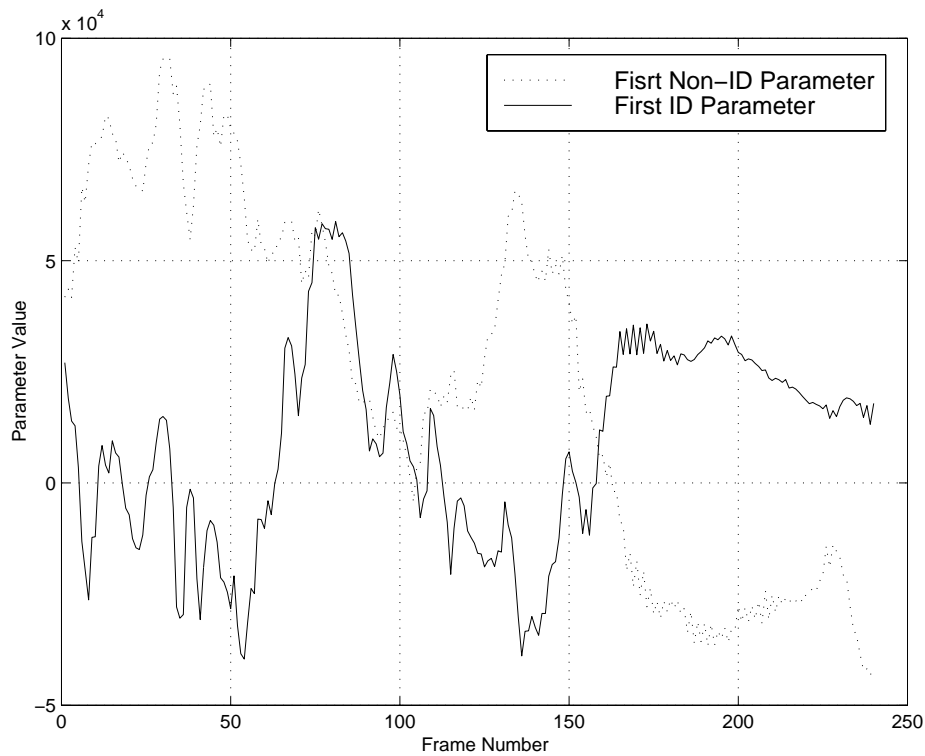
The success of Kalman filtering depends on the appropriateness of the model used. If the dynamic model is not appropriate for the actual data, there is a danger of making things worse rather than better. The tracking scheme described above depends on the assumption that decoupling into identity and non-identity models is possible. Before we can apply it, we must first establish that the decoupling is producing identity and non-identity signals for which the tracking filters are appropriate. The constant filter used to track identity is only valid in a *signal processing* sense if the identity measurement is truly constant. We know that in a *physical* sense, the true measure of identity must be constant - this means that any observed systematic variation in the identity parameters during a sequence must be due to the inadequacy of the partitioned model.

We begin by examining some example tracks of identity and non-identity parameters for a typical sequence. These are the unfiltered measured parameters taken from a typical sequence of a person speaking whilst varying pose and expression (a more detailed description of the database from which this sequence is taken is given in Section 8.2). Figure 7.5 shows the paths of  $d_1$  and  $r_1$ , the first identity and non-identity parameters. There are two obvious features: firstly, the identity parameter does not appear to be a simple constant corrupted by noise, there is clearly some systematic drift, and secondly, this drift appears to be (anti-)correlated in some way with drift in the non-identity parameter. These observations suggest that the assumption of constant identity used in the design of the Kalman filtering scheme is not valid.

## 7.6 Dynamically updating the partitioned model

Whilst the tracking scheme shown in Figure 7.4 is attractive, we have seen that the partitioned model does not give sufficient separation of identity and non-identity





**Figure 7.5:** First 2 identity and non-identity parameters for a typical sequence.

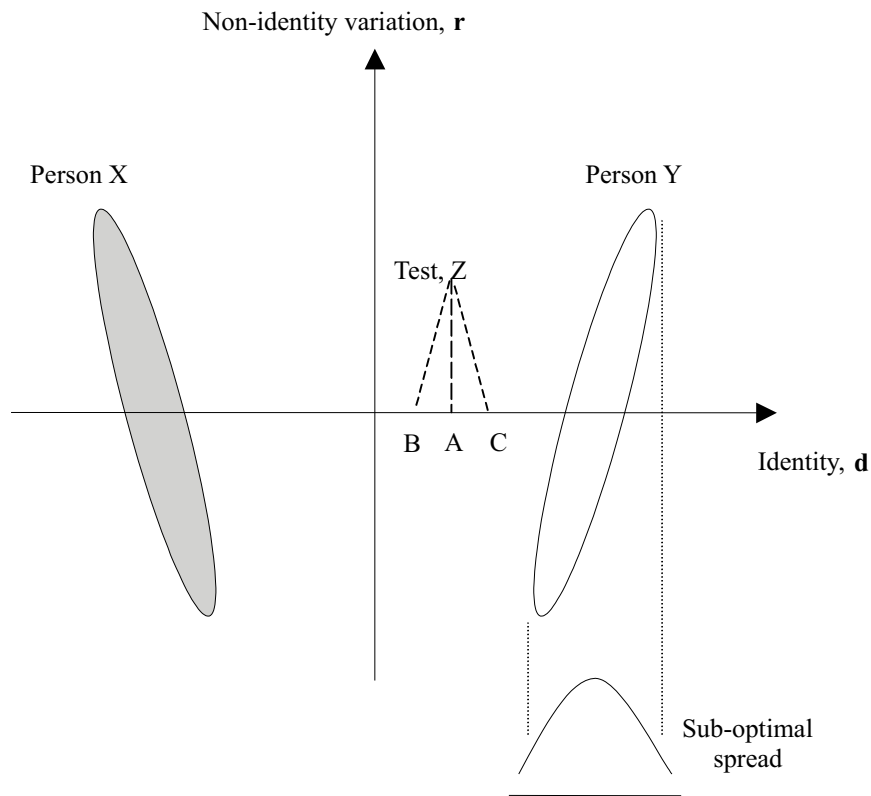
components for it to be workable. Observation of the parameters suggests, however, that there may be a simple relationship between the two that might be used to remove the correlated variation leaving an identity signal that behaved as a noisy constant value. Since we know the *real* identity must be fixed, we use this knowledge to remove unwanted correlation between the parameters.

### 7.6.1 Motivation

The unwanted correlation illustrated in Figure 7.5 is caused by the inability of the partitioned model to deal with individuals who exhibit *different* within-class variability. We reason that by observing the behaviour of the model during a sequence, a refined estimate of the within-class and between-class spaces can be made. This in turn will lead to a more stable estimate of the identity parameters, and finally, the true identity.

The partitioned models are built using a global analysis of the pooled within-class variation, for all individuals. This does not take into account possible differences in the way particular individual faces vary. For example, the way a face image changes with variation in pose will, to some extent, be dependent on the length of the individual's nose. We present a scheme that effectively performs a correction of the global partitioning on-line during tracking.

To illustrate how the problem arises, we consider a simplified example in which appearance is described in a 2-dimensional space as shown in Figure 7.6. We imagine a large number of representative training examples for two individuals, person X and person Y, projected into this space. The optimum direction of identity variation,  $\mathbf{d}$ , and the direction of within-class variation  $\mathbf{r}$ , are shown. A perfect discriminant



**Figure 7.6:** Limitation of Linear Discriminant Analysis: Best identification possible for single example, Z, is the projection, A. But if Z is an individual who behaves like X or Y, the optimum projections should be C or B respectively.

analysis of identity would allow two faces of different pose, lighting and expression to be normalised to a reference view, and thus the identity compared. It is clear from the diagram that an orthogonal projection onto the identity subspace is not ideal for either person X or person Y, but gives the best compromise. Given a fully representative set of training images for X and Y, we could work out in advance the ideal projection. We do not, however, wish (or need) to restrict ourselves to acquiring training data in advance. If we wish to identify an example of person Z, for whom we have only one example image, the best estimate possible is the orthogonal projection, A, in Figure 7.6. We cannot know from a single example whether Z behaves like X (in which case C would be the correct identity) or like Y (when B would be correct) or indeed, neither. The discriminant analysis produces only a first order approximation to class-specific variation.

### 7.6.2 Formulation

We seek to calculate class-specific corrections from image sequences. The framework used is the Appearance Model, in which faces are represented by a parameter vector  $\mathbf{c}$ . Partitioning yields a first order global approximation of the linear subspace describing identity, given by an identity vector,  $\mathbf{d}$ , and the residual linear variation, given by a vector  $\mathbf{r}$ . A vector of appearance parameters,  $\mathbf{c}$  can thus be described by

$$\mathbf{c} = \mathbf{D}\mathbf{d} + \mathbf{R}\mathbf{r} \quad (7.34)$$

where  $\mathbf{D}$  and  $\mathbf{R}$  are matrices of orthogonal eigenvectors describing identity and residual subspaces respectively.  $\mathbf{D}$  and  $\mathbf{R}$  are orthogonal with respect to each other and the dimensions of  $\mathbf{d}$  and  $\mathbf{r}$  sum to the dimension of  $\mathbf{c}$ . Recall that the projection from a vector,  $\mathbf{c}$  onto  $\mathbf{d}$  and  $\mathbf{r}$  is given by

$$\mathbf{d} = \mathbf{D}^T \mathbf{c} \quad (7.35)$$

and

$$\mathbf{r} = \mathbf{R}^T \mathbf{c} \quad (7.36)$$

Equation 7.35 gives the orthogonal projection,  $\mathbf{d}$ , onto the identity subspace - the best available basis for classification given a single example. We assume that this projection is not ideal, since it is not class-specific. Given further examples, in particular, from a sequence, we seek to apply a class-specific correction to this projection. It is assumed that the correction of identity required has a linear relationship with the residual parameters, but that this relationship is different for each individual. Formally, if  $\mathbf{d}_c$  is the true projection onto the identity subspace for a given individual,  $\mathbf{d}$  is the orthogonal projection and  $\mathbf{r}$  is the projection onto the residual subspace, then,

$$\mathbf{d} = \mathbf{d}_c + \mathbf{A}\mathbf{r} \quad (7.37)$$

where  $\mathbf{A}$  is a matrix giving the correction of the identity, for the observed residual parameters.

During a sequence, many examples *of the same face* are seen. At any point in the sequence we use all the measurements from the previous frames to solve Equation 7.37 using standard multivariate linear regression, thus obtaining matrix  $\mathbf{A}$ . At frame  $i$ , the estimated corrected identity,  $\mathbf{d}_c^i$  is given by:

$$\mathbf{d}_c^i = \mathbf{d}^i - \mathbf{A}\mathbf{r}^i \quad (7.38)$$

where  $\mathbf{r}^i$  is the value of the residual vector at frame  $i$ .

### 7.6.3 An adaptive tracking scheme

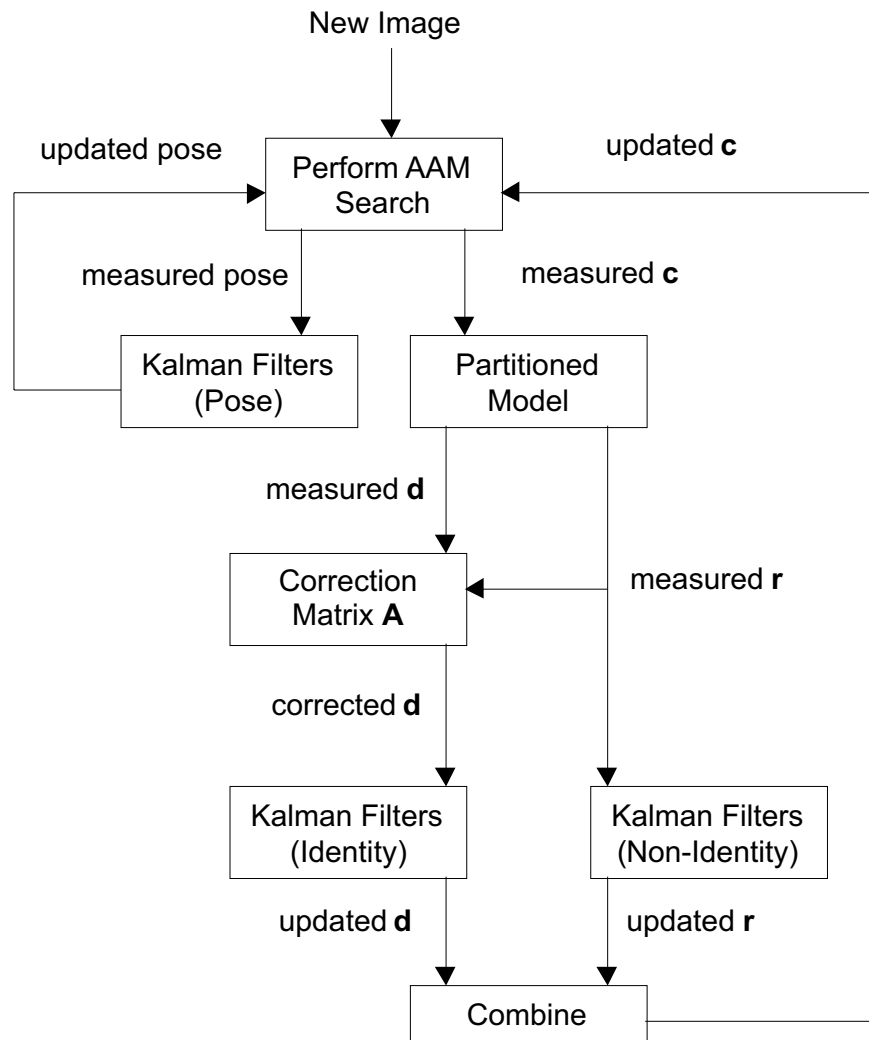
Updating the estimate of the class-specific variation allows us to improve upon the tracking scheme of Figure 7.4. In each frame of a video sequence, an Active Appear-

ance Model can be used to locate the face. The iterative search procedure returns a set of parameters describing the best found match of the model to the data. The combined model parameters are projected into the identity and residual subspaces by equations 7.35 and 7.36. At each frame,  $i$ , the identity vector,  $\mathbf{d}^i$ , and residual vector,  $\mathbf{r}^i$  are recorded. The correction matrix  $\mathbf{A}$  is estimated using all the previously stored values of the identity and residual vectors. The corrected ID parameters,  $\mathbf{d}_c^i$  are calculated using equation 7.38. Until enough frames have been recorded to allow the matrix  $\mathbf{A}$  to be calculated,  $\mathbf{A}$  is set to contain all zeros, so that the corrected estimate of identity,  $\mathbf{d}_c$  is the same as the orthogonally projected estimate,  $\mathbf{d}$ .

Three sets of Kalman filters are used to track 2D-pose, corrected ID variation,  $\mathbf{d}_c$ , and non-ID variation,  $\mathbf{r}$ , using the models described in Sections 7.4.3 and 7.4.4. The full, adaptive tracking scheme with ID-space refinement is shown diagrammatically in Figure 7.7.

## 7.7 Initial evaluation

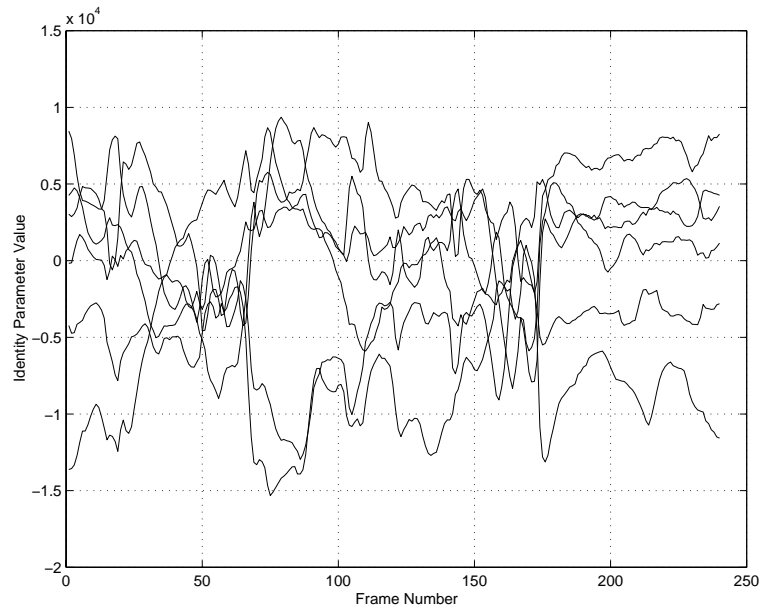
As a preliminary investigation of the tracking schemes we applied the methods to 24 video sequences (each of different individual). Each sequence was 240 frames long (approximately 10 seconds) and was tracked for the full duration using three different schemes - *simple*, *decoupled* and *adaptive*. The filtered schemes were only ‘switched-on’ after 100 frames, thus up to that point all the methods were identical. After frame 100, the decoupled scheme was used to track identity and non-identity parameters separately. Likewise, the adaptive scheme was used to track the separate components but using the complete ID-correction scheme illustrated in Figure 7.7.



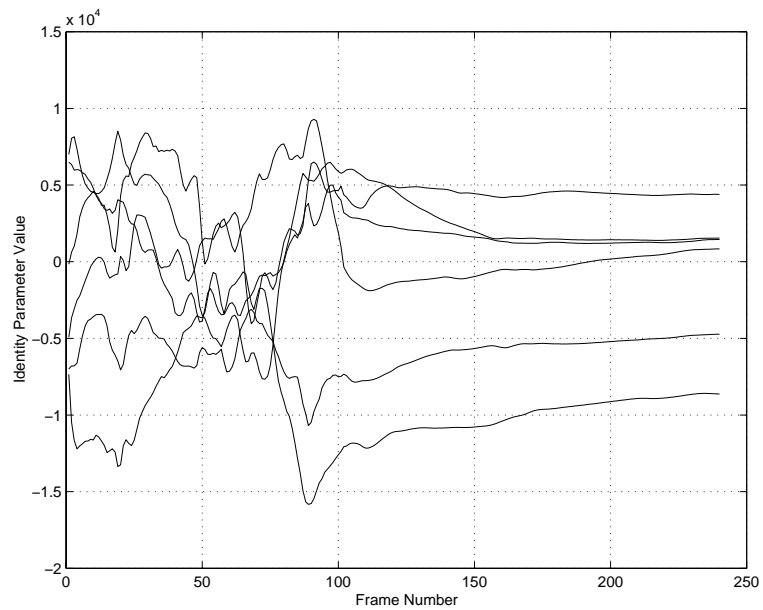
**Figure 7.7:** Schematic diagram of full, refined tracking algorithm.

### 7.7.1 Stability of identity measurement

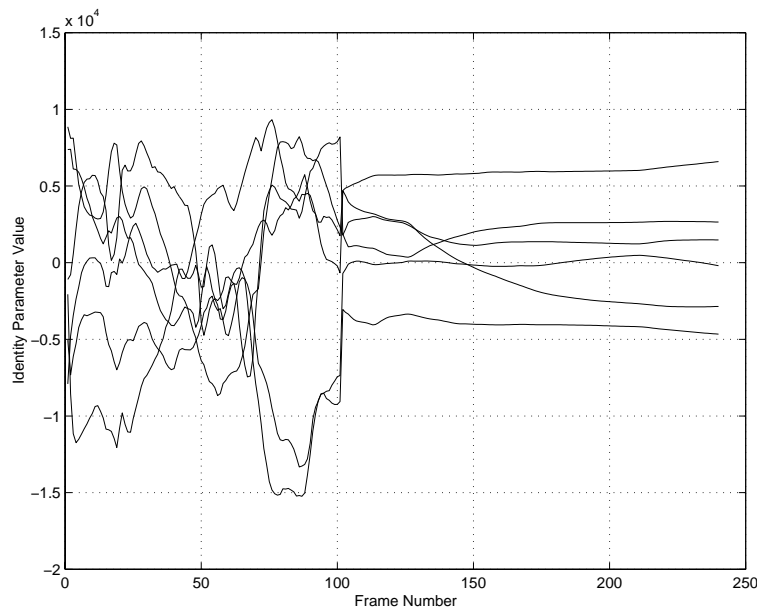
We first illustrate the effects of the various schemes on the estimated identity parameters, showing corresponding plots for a typical test sequence. Figure 7.8 shows the variation of the first 6 identity parameters extracted from a sequence using the simple tracking method. Figure 7.9 shows the variation using decoupled tracking, and Figure 7.10 the variation using the full adaptive scheme. The latter two schemes are only ‘switched-on’ after frame 100. In principle, up to frame 100 the parameters should be identical, however, the starting position in each trial was not guaranteed to be identical, so some difference is expected.



**Figure 7.8:** Typical values of first 6 identity parameters versus frame number, using simple tracking scheme.



**Figure 7.9:** Typical values of first 6 identity parameters versus frame number, using decoupled tracking scheme.



**Figure 7.10:** Typical values of first 6 identity parameters versus frame number, using full, adaptive tracking scheme.

Figures 7.9 and 7.10 show that decoupled and adaptive filtering of the identity parameters have the effect of reducing the observed variation in identity. We derived a quantitative measure of the variation by computing the covariance matrix  $\mathbf{C}_i$  of the identity parameters from the last 50 frames for each sequence,  $i$ . Since the identity parameters are mutually orthogonal the trace of  $\mathbf{C}_i$  gives an estimate of the stability of the identity estimate. By computing  $\mathbf{C}_i$  for each of the 24 sequences we compared the stability of the identity estimate for the different schemes. The average values of  $\text{Trace}(\mathbf{C}_i)$  for the three tracking schemes are shown in the first column of Table 7.1. This shows that the decoupled scheme gives more stable estimates than the simple scheme but that significantly better results are obtained using the adaptive scheme. Both filtered schemes will, of course, always tend towards smaller variance in ID the more examples they see, due to the averaging property of the simple Kalman filters used, however, in these experiments, the adaptive scheme was found to reduce variance more quickly.



	Average Trace ( $10^8$ )	Relative Fit Error (%)
Simple	4.80 0.16	0
Decoupled	4.08 0.75	11.72 0.17
Adaptive	2.67 0.35	2.57 0.16

**Table 7.1:** Measure of ‘stability’ of identity estimates for 3 tracking methods compared with the average percentage difference in reconstruction error.

### 7.7.2 Reconstruction error

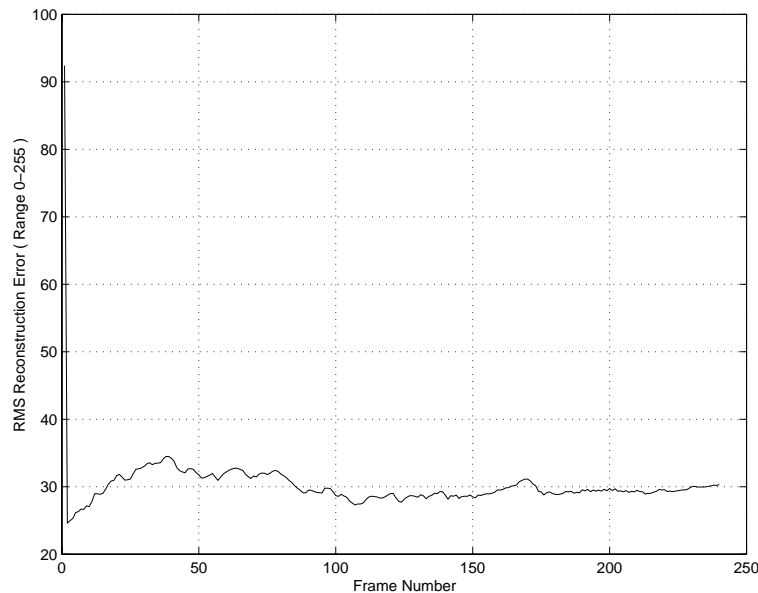
A key aim of filtered tracking is to provide a stable estimate of the identity parameters (and thus the estimated identity) of the face in the sequence. Both the decoupled scheme and the adaptive scheme will tend to produce stable estimates of the identity parameters after many frames, however, since decoupling alone is not sufficient to remove systematic variation in identity parameters, the Kalman filter model is inappropriate for this scheme. An expected consequence is degradation of tracking accuracy. By using the adaptive scheme we expect less degradation in tracking accuracy.

Using the same set of 24 sequences, we measured the average reconstruction error,  $R$ , for each frame of each sequence, using the three tracking schemes.  $R$  is defined as the RMS difference in grey-level values over each of the  $N$  sample points in the shape-free model framework, given by:

$$R = \sqrt{\frac{|(\mathbf{g} - \mathbf{g}_m)|^2}{N}} \quad (7.39)$$

where  $\mathbf{g}$  is the vector of grey-level samples from the image and  $\mathbf{g}_m$  is the vector of samples from the current instance of the model.

As in the experiments above, the filtered schemes were only activated after the first 100 frames. Figure 7.11 plots the average value of  $R$  obtained by tracking the sequences using the simple, unfiltered scheme.

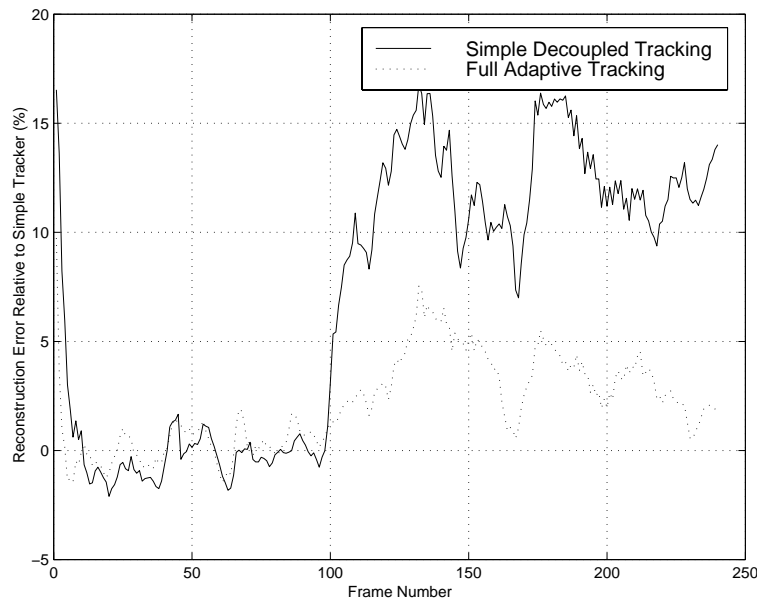


**Figure 7.11:** Reconstruction error during tracking using simple, unfiltered scheme.

To compare the filtered tracking schemes with the simple scheme we computed the difference between the respective reconstruction errors for each sequence. Figure 7.12 shows the average percentage difference in reconstruction error (compared with the simple tracking scheme) for the simply decoupled and adaptive tracking schemes.

The second column of Table 7.1 shows the average reconstruction error for the last 50 frames of the 24 test sequences using the three tracking methods, alongside the corresponding estimate of stability in ID parameters.

The results presented in Figure 7.12 show that decoupled tracking produces a significantly *worse* reconstruction error than simple tracking, whilst the degradation using adaptive tracking is only slight. Since we hypothesise that the adaptive method should increase robustness by preventing the system changing the identity inappropriately, one might expect the adaptive tracking scheme to perform *better* than raw



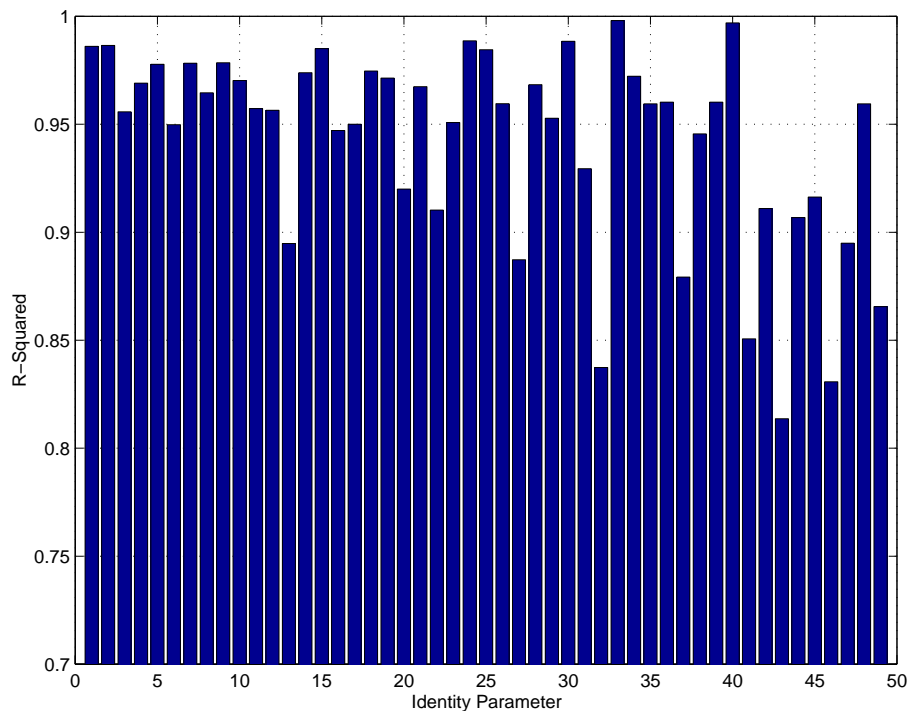
**Figure 7.12:** Average percentage difference in reconstruction error (compared with the simple tracking scheme) for the simply decoupled and adaptive tracking schemes.

tracking, a result which was not observed. When assessing these results it is important, however, to note that the simple tracking scheme uses a search method which drives the model in order to directly minimise reconstruction error. The adaptive scheme adds extra requirements of consistency in the identity, requirements which are applied *after* the search in each frame is complete. If the simple tracking is itself very good, there is no guarantee that this adjustment will improve reconstruction error. We expect that the adaptive tracking scheme will prove more robust than simple tracking in more difficult, noisy images, when the extra constraints on identity provide an increase in the specificity of the system.

At this point we have demonstrated that it is possible to apply tracking schemes which yield stable estimates of identity. We have shown that an adaptive scheme shows less degradation in reconstruction error and an increase in the stability of the measured identity parameters. In the following chapter we present results from more extensive tracking experiments which show that the stable estimate of identity can be used to achieve higher recognition performance than can be achieved using individual frames from sequences.

### 7.7.3 Linear relationship between parameters

The adaptive tracking scheme is based on multivariate linear regression. The matrix,  $\mathbf{A}$  is used to ‘predict’ a correction to the identity parameters based on the non-identity parameters. We estimated the quality of the prediction by measuring the R-squared statistic [54] for each parameter, obtained from the linear regression. An R-squared value of unity corresponds to a perfect linear relationship. We computed the R-squared values for each identity parameter at the end of each of the 24 sequences. Figure 7.13 shows the average value for each parameter. These results suggest that the required identity correction is well approximated by a linear model.



**Figure 7.13:** Average value of R-squared statistic for each identity parameter, indicating a strong linear relationship between the identity and non-identity parameters.

## 7.8 Summary

In this chapter we have introduced a novel tracking scheme using Active Appearance Models. The approach was motivated by previous successful demonstrations of model-based tracking using Active Shape Models combined with Kalman filtering.

Kalman filtering with an ASM is realised through the assumption of independence of the model parameters. For our full appearance-based face model, we know this assumption is invalid: changing a parameter of the full model can change both identity and non-identity components, whereas during tracking identity should be fixed. A partitioned model provides a framework in which identity variation is decoupled from other variation. Unfortunately, the decoupling is not sufficiently good as to result in constant identity parameters over a sequence.

We have described a method that applies a further correction to the original decoupling, taking into account variation which is specific to the individual being tracked. The method exploits the fact that the *true identity* must be fixed during a sequence.

We have shown that the minimum reconstruction error is achieved with simple tracking, but that there is considerable instability in the estimate of identity. The decoupled scheme produces a slightly more stable estimate of identity, but shows a significant increase in reconstruction error. The adaptive tracking scheme provides a large improvement in the stability of ID measurement whilst showing a much smaller degradation in reconstruction accuracy.

In the following chapter we present further experiments, which show how the scheme can be used for improved dynamic identity recognition.

# Chapter 8

## Interpreting Sequences

This chapter presents the results of experiments in video sequence interpretation, concentrating particularly on person identification. Using a database of video sequences, we show that dynamic interpretation offers improved recognition compared to the analysis of static images. In order to achieve this improvement it is necessary to use the measurement framework described in Chapter 7, which accounts for the different types of variation present in sequences of different individuals.

### 8.1 Interpretation by tracking

The tracking scheme shown in Figure 7.4 represents a holistic approach: robust tracking is achieved through interpretation. It happens that the key feature to isolate for adaptive tracking, *identity*, is the feature in which we are often most interested for interpretation. Partitioning the Appearance Model using static images gives us a first-order approximation to the subspace that defines identity. The on-line refinement in the adaptive tracking scheme provides a further class-specific linear correction to the first-order estimate, using a video sequence of an individual. Thus, we might expect to achieve better identity recognition when analysing video.

Whilst some systems might be confounded by variability in sequences of a given individual, our system actually requires variability in order to learn the extra on-line correction. This makes intuitive sense: multiple views of an individual ought to increase the amount of useful information - for example, the length of the nose is better estimated from a profile than a frontal image.

In this chapter, we demonstrate the use of the adaptive tracking scheme presented in Chapter 7 for enhanced identification from sequences. We compare the recognition performance of the dynamic scheme with that obtained by static analysis of single frames from sequences.

## 8.2 Experimental framework

In this section we describe the experimental framework used to evaluate the identification performance of adaptive tracking system and to compare the results with simpler schemes. The experiments are based on matching unseen *probe* images and sequences to pre-registered *gallery* images and sequences.

### 8.2.1 The interpretation task

We present an assessment of *verification* performance - the key requirement in many access control applications. Given an probe face and a *claimed* identity, the system must determine if the claim is correct by comparing the probe with a gallery of known faces. A fair test should not allow the classifier to assume that all the probes exist within the gallery.

In our experiments we cross-compared every probe face with every gallery face, by measuring the Euclidean distance between the identity vectors,  $\mathbf{d}_p$  (probe) and  $\mathbf{d}_g$  (gallery) obtained from the images or sequences:

$$D = \sqrt{(\mathbf{d}_g - \mathbf{d}_p)(\mathbf{d}_g - \mathbf{d}_p)^T} \quad (8.1)$$

We evaluated performance in three scenarios:

- *Static-Static* - Probe and Gallery identity vectors are obtained from static images
- *Dynamic-Static* - Probe identity vectors are obtained by adaptive tracking
- *Dynamic-Dynamic* - Probe and Gallery identity vectors are obtained by adaptive tracking

The system was asked to return a ‘hit’ every time the distance between a gallery face and a probe face was below a certain threshold,  $T$ . The verification *decision rule* is:

$$\text{if } D < T \quad \text{person is the same} \quad (8.2)$$

$$D > T \quad \text{person is not the same} \quad (8.3)$$

We calculated both the True Positive Fraction (TPF) and False Acceptance Rate (FAR) for various levels of  $T$ , thus producing an ROC-curve as described in Section 2.14.2.

### 8.2.2 Test data

In order to test the system it was necessary to collect a database of test sequences. We have observed previously, particularly in results from the FERET test described in Chapter 2, that most systems are poor at identification when there is a time delay between the capture of ‘gallery’ and ‘probe’ images. The discrepancy between



time-separated and same-day recognition is almost certainly due to lack of variability between the same-day image pairs. Often the person is filmed in the same position, with the same lighting conditions, etc. To address this, we captured two sets of sequences with an interval of 4 months between sessions.

The ‘gallery’ set consists of one sequence each of 24 different individuals. The set covers an age range of 21 to 50 years, and contains both men and women, although there is, at present, only a single example of a non-Caucasian face. Each sequence is 20 seconds long, during which the individual was asked to recite a paragraph of text. Pose variation was obtained by asking the person to follow a moving target whilst reciting. The maximum pose deviation from the fronto-parallel view was typically 30 degrees laterally and 20 degrees vertically.

The ‘probe’ set consists of 7 sequences of each individual in the training set. Unfortunately, 2 of the individuals in the gallery were unavailable 4 months later, thus the probe set consists of a total of  $7 \times 22 = 154$  sequences. Each of these sequences is approximately 10 seconds long. For each sequence, the individual was asked to repeat the same piece of text, but was instructed to do so in one of seven ‘styles’ - happy, sad, afraid, angry, surprised, disgusted or neutral. Pose variation was again obtained by asking the individual to follow a target whilst reciting. The aim was to capture a range of both pose and expression in the probe set. Whilst no effort was made to ‘fix’ the lighting conditions, neither did we have the resources to deliberately generate lighting variation. These results must be viewed in the context of fairly fixed lighting conditions, although it is important to note that the lighting conditions *are* different between the gallery and probe set.

The data in both gallery and probe sets was digitally captured at 24 frames per second. The images are 24-bit colour, subsequently reduced to 8-bit greyscale at a resolution of 640x482 pixels. Figures 8.1 and 8.2 show some typical data from the probe and gallery sequences respectively.



**Figure 8.1:** Some examples frames from sequences of individuals in the ‘probe’ set.

### 8.3 Static-Static recognition

We wish to compare the performance of the dynamic interpretation scheme with straightforward static recognition. A typical static recognition scenario might consist of a single gallery image used to register the individual. Indeed, this is the minimum possible data with which recognition can be performed - in the ideal case this is all we should need. For the purpose of this experiment we extracted just one frame (the first) from each of the gallery sequences. This would correspond to the user having his/her photograph taken to register on the system.

Each gallery image was registered by performing Active Appearance Model search. A human operator initialised the search by locating the centre of the left eye. The search was supervised to check that the AAM converged, ensuring that the measurement was sensible. It took approximately 7 seconds for the operator to register each image. The gallery of 24 images was registered in less than 3 minutes.

The test images consisted of random frames taken from the probe sequences. For



**Figure 8.2:** Some examples frames from sequences of individuals in the ‘gallery’ set.

each individual we chose 10 random images. These images were then interpreted using Active Appearance Model search, again with hand initialisation. It was essential to supervise the initialisation, in order to make a fair comparison with the sequence recognition algorithm; searches which did not converge would lead to an unfairly (in this context) low score for the static recognition system.

Each set of measured parameters for the gallery and probe images was first projected into the *identity* subspace using equation 5.4 as described in Chapter 5, resulting in identity vectors,  $\mathbf{d}_p$  and  $\mathbf{d}_g$  for probe and gallery images respectively.

Table 8.1 at the end of this chapter shows the average distance between gallery and probe images *of the same person* (same-person distance) and compares this with the average distance between the gallery images and *all* the probe images (all-person distance). The differences between the two cases are not great, indeed the same-person distance is not always less than the all-person distance. This apparently poor separation of the individuals is reflected in the verification ROC plot for static-static recognition shown as the dashed line in Figure 8.4. The curve shows that a True Positive Fraction of only 63% is obtained for a False Acceptance Rate of 20%. For the same FAR, the participants in the time-separated part of the FERET test [81]

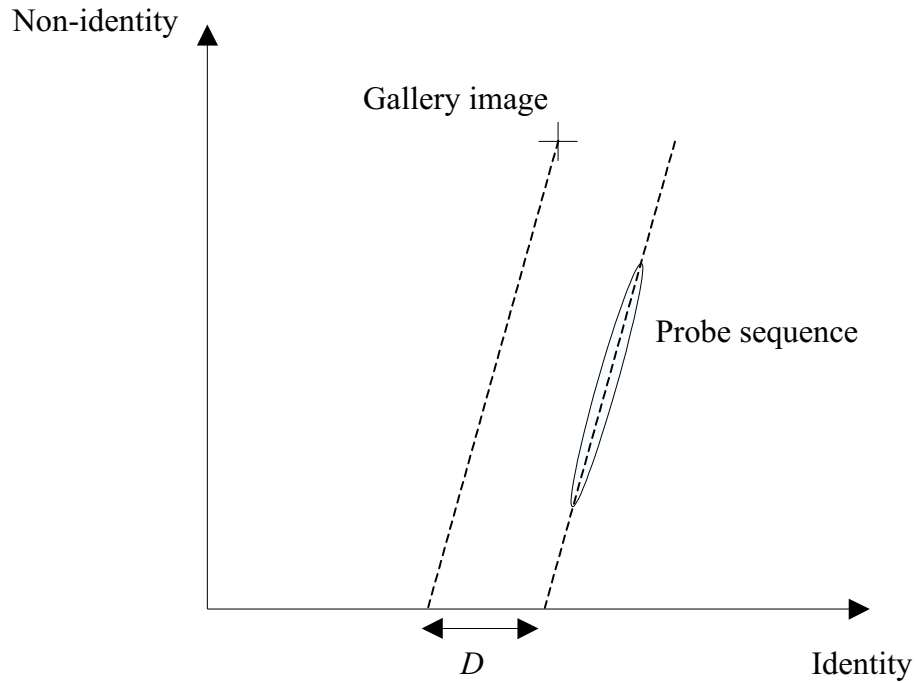
reported TPF's of between 72% and 86%. Without a common data set, it is difficult to directly compare our static-static results with the FERET results, however, in both cases, the level of performance is inadequate for an effective access-control system.

## 8.4 Dynamic-Static recognition

We evaluated the dynamic-static recognition system in a similar way to the static-static recognition scheme, using the same registration of the gallery images. Recall that, rather than a straight projection onto the identity subspace, the identity vector for a probe sequence is estimated recursively using the Kalman Filter and correction scheme illustrated in Figure 7.7. When comparing the distance between gallery sequences and probe images, we used the same *corrected* projection observed in the sequence to project the gallery image onto the identity subspace. When we compared the distance between a specific probe sequence and a gallery image, we calculated the distance that would be measured *if the gallery behaved like the probe sequence*. This is illustrated in Figure 8.3.

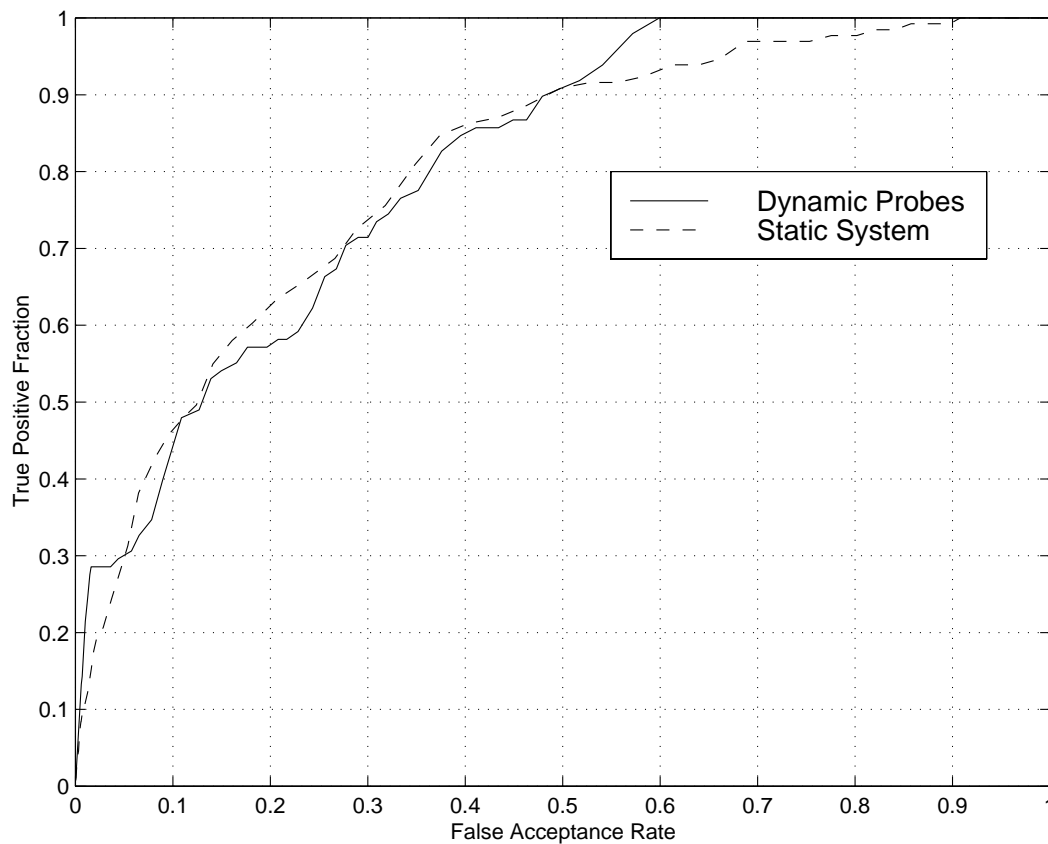
We tested the dynamic recognition system on each of the 168 probe sequences. A human operator gave the system the position of the left eye in the first frame of each sequence. The tracker was then run over the whole sequence. The final filtered, corrected estimate of the identity parameters along with the probe correction matrix,  $\mathbf{A}$  was returned. This correction matrix was also used to adjust the estimates of the gallery identity parameters using equation 7.38. We then compared distances between the probe sequences and gallery images. These distances are referred to as *dynamic-static* distances.

The dynamic-static distances are given in Table 8.2 at the end of this chapter. There is, on average, a slightly greater difference between the same-person and all-person distances, and every same-person distance is less than the corresponding all-person distance. This small improvement is only slightly reflected in the ROC curve.



**Figure 8.3:** Distance between probe sequence and gallery image is calculated by projecting the image in the same way as the sequence.

Figure 8.4 shows the ROC curve for the dynamic-static system as a solid line. The curves for this and the static-static system are very similar, although the dynamic-static system achieves a True Positive Fraction of 100% for a False Acceptance Rate of 60%. For the same TPF, the static-static system displays an FAR of 93%. The performance of the dynamic-static system remains inadequate for an effective access control system.

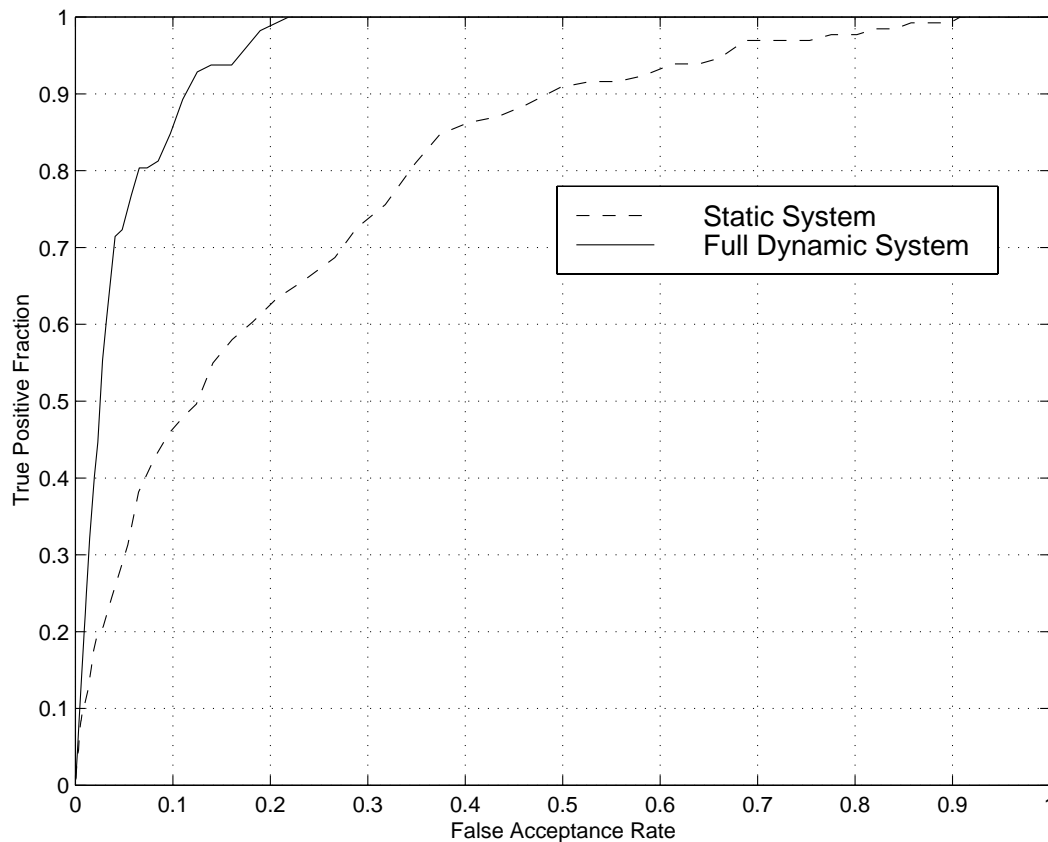


**Figure 8.4:** ROC curves for verification system. Dynamic-static system compared with static-static system.

## 8.5 Dynamic-Dynamic recognition

The dynamic recognition system is limited by using a single gallery image for each individual. In this database we have sequences of the gallery as well as the probes. We performed a further experiment in which the gallery sequences were also tracked (with hand initialisation). Correction matrices,  $\mathbf{A}_g$  and  $\mathbf{A}_p$ , were obtained for each gallery and probe sequence respectively. The adaptive tracking system described in Section 7.6.3 was used to obtain corrected estimates of identity for both the gallery and probe sequences. The distances between corrected gallery and corrected probe identity vectors were measured and are referred to as *dynamic-dynamic* distances.

We tested the dynamic-dynamic recognition system on each of the 168 probe sequences. The same-person and all-person distances are given in Table 8.3. There is a significantly larger difference between the average same-person and all-person distances than was the case in either of the two previous experiments. This increase in performance is clearly reflected by a large improvement in the corresponding ROC curve. Figure 8.5 compares the ROC curve for the dynamic recognition system with that of the static system. A TPF of 98% is achieved for an FAR of 20%. At an FAR of 22% the TPF is 100%.



**Figure 8.5:** ROC curves for verification system. Dynamic-dynamic system compared with static-static system.

## 8.6 Discussion of results

The obvious feature of the results shown in Figures 8.4 and 8.5 is the dramatic improvement obtained by using both a dynamic gallery and dynamic probes.

For both systems using the static gallery, a 20% False Acceptance Rate, corresponds to a True Positive Fraction of around 60% with little difference between using dynamic or static probes. This means that even if we turned away 4 in 10 genuine candidates, 2 in every 10 illegal entry attempts would succeed. It is difficult to imagine the practical application of a system exhibiting this level of performance. The poor ROC curves reflect the relatively small differences between the same-person and all-person differences.

If we use both a dynamic gallery and dynamic probes the situation improves considerably. It is possible to achieve 100% True Positive Fraction for a False Acceptance Rate of just 22%. This sort of performance would be acceptable in many types of application, particularly access control. One of the key requirements of a practical access control system is the ability to achieve a very high True Positive Fraction - it is usually better to allow a few bogus entries than to regularly turn genuine people away. If the TPF is high, the ‘alarm’ raised by a detected intruder can be very dramatic, since it is unlikely to ever be triggered by a genuine person.

Given this large improvement when using a dynamic gallery, it is slightly surprising that when using a static gallery, it appears to make little difference whether the probes are static or dynamic. One possible explanation is that the variation seen in a probe sequence is only sufficient to correct *that* sequence. The configuration of the face in a static gallery image may not correspond to variation seen in the probe sequence. It appears that, in order to obtain a reliable estimate of the gallery identity, the system needs see the gallery image move.

The suggestion that the system needs to see a face move before it can be reliably identified has an interesting parallel in human psychological studies performed at

























The University of Glasgow by Burton *et al* [17]. They evaluated the performance of human subjects when asked to identify individuals in poor quality video sequences. They found that the performance for *familiar* faces - those known to the subject before the experiments, was much greater than when the face was *unfamiliar* - only seen before in still photographs. By using a dynamic gallery, our system ‘learns’ in advance something about the dynamic behaviour of faces to be identified. Such dynamically tracked gallery faces are the system’s equivalent of familiar faces. This relationship is an interesting area for future research.

## 8.7 Summary























In this chapter we have presented an experimental evaluation of the various recognition methods described in this thesis. As other experimenters have shown, static recognition across large changes in image appearance with a time-delay between gallery and probe images is difficult. We address this problem by learning further information about the gallery and probe images by observing their movement through sequences. This can be done using the adaptive tracking scheme described in Chapter 7. It is important to note that this extra information does not need to be stored with the gallery images, it is simply used to ‘correct’ an initial estimate of the low-dimensional identity vector. This is then compared with the corrected version of the gallery identity vector.

The results indicate that it is important to observe variation in both the probe and gallery images. The performance obtained using a static gallery is not sufficient for access control systems - however, by adopting the dynamic-dynamic scheme we have demonstrated much improved performance which would allow practical access control. The need to learn about the gallery sequences in advance has interesting parallels in studies of human recognition performance from video, in which the prior-familiarity of the face appears important.

These results are based on a fairly small sample of individuals, at least compared with common static recognition experiments such as the FERET test. This is due to the vastly increased difficulty of obtaining, storing and processing video data. It is hoped that further increases in the size of database used for these tests will be achieved by pooling resources across research establishments. We intend to make our test data publicly available (see Appendix B).























Person	Distance between gallery/probe		Person	Distance between gallery/probe	
	Same Identity	All Individuals		Same Identity	All Individuals
	3.84	5.14		4.68	5.11
	4.09	4.97		3.47	4.90
	4.54	5.45		4.63	5.04
	5.83	5.35		4.07	5.25
	3.66	6.24		4.19	5.40
	4.05	5.10		5.30	5.87
	4.08	5.46		4.63	5.31
	4.32	5.59		4.58	4.79
	4.04	5.15		3.58	5.01
	4.00	4.54		5.04	5.04
	4.39	5.09		3.81	4.95
<div> Average distance between matches: 4.31  Average distance between all pairs: 5.22 </div>					

**Table 8.1:** Average distance between gallery and probe images using *static-static* recognition scheme.

Person	Distance between gallery/probe		Person	Distance between gallery/probe	
	Same Identity	All Individuals		Same Identity	All Individuals
	4.27	5.81		4.22	8.59
	5.07	6.87		7.00	7.92
	7.27	8.36		7.00	8.48
	3.60	7.94		5.53	5.91
	6.67	8.37		5.97	7.18
	5.41	7.04		5.79	7.03
	5.74	8.08		5.36	5.48
	6.13	7.89		6.45	7.48
	5.28	7.06		4.88	6.07
	5.73	8.08		8.18	8.72
	5.93	7.97		5.42	6.34

<b>Average distance between matches:</b>	<b>5.77</b>
<b>Average distance between all pairs:</b>	<b>7.39</b>

**Table 8.2:** Average distance between gallery and probe images using *dynamic-static* recognition scheme.

Person	Distance between gallery/probe		Person	Distance between gallery/probe	
	Same Identity	All Individuals		Same Identity	All Individuals
	3.99	7.01		3.11	6.91
	4.13	6.82		4.29	7.46
	4.54	6.93		4.31	6.44
	3.91	7.22		2.99	7.42
	5.78	8.17		5.38	9.00
	4.36	6.98		4.49	6.73
	5.21	7.34		3.66	7.01
	3.46	7.56		4.90	6.84
	3.91	6.94		4.57	8.02
	3.09	6.91		3.29	7.37
	4.20	7.99		2.27	7.38

<b>Average distance between matches:</b>	<b>4.08</b>
<b>Average distance between all pairs:</b>	<b>7.29</b>

**Table 8.3:** Average distance between gallery and probe images using *dynamic-dynamic* recognition scheme.

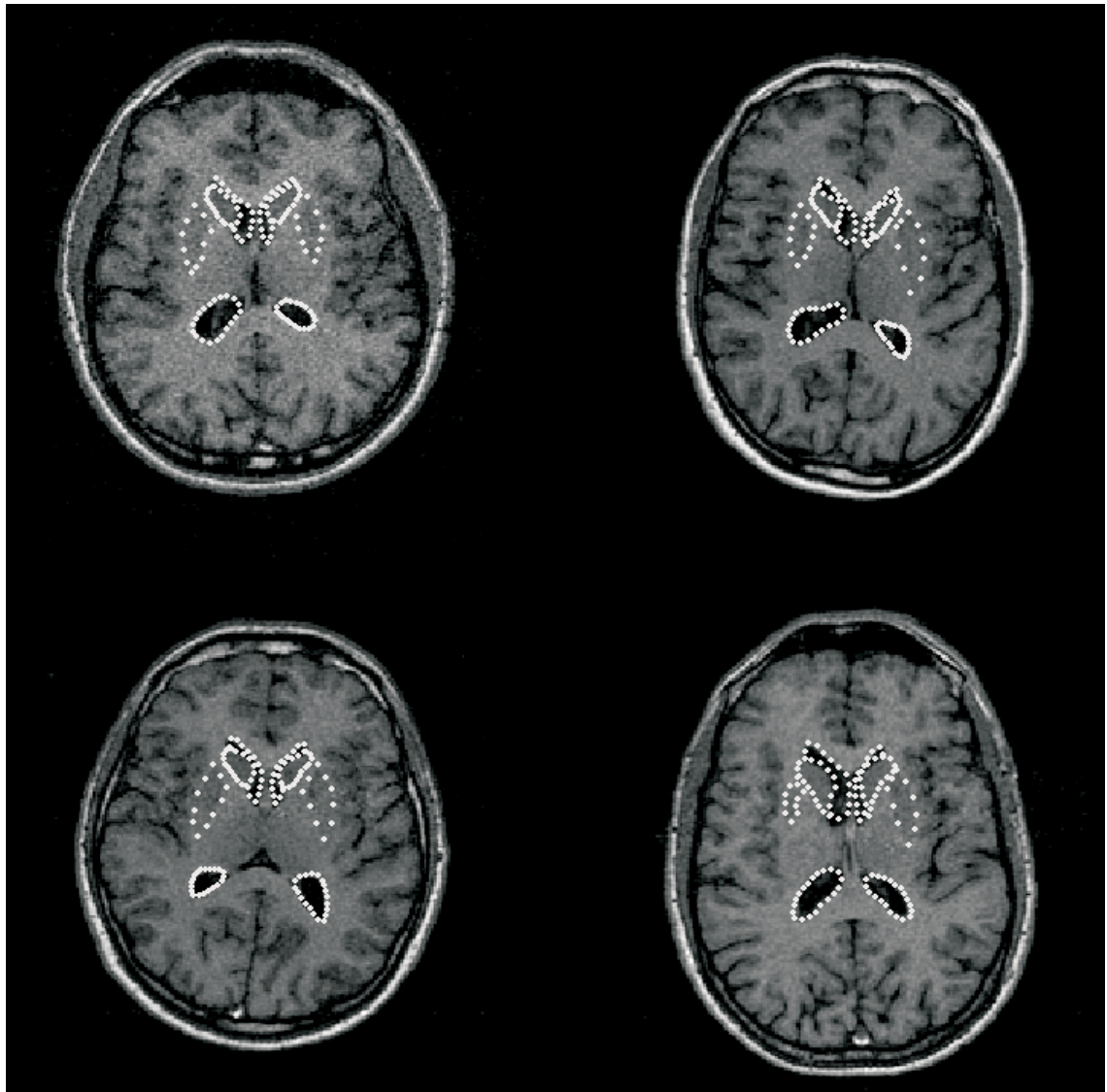
# Chapter 9

## Extensions and Future Work

In this chapter we describe initial work on extensions to the techniques presented and discuss directions for future research.

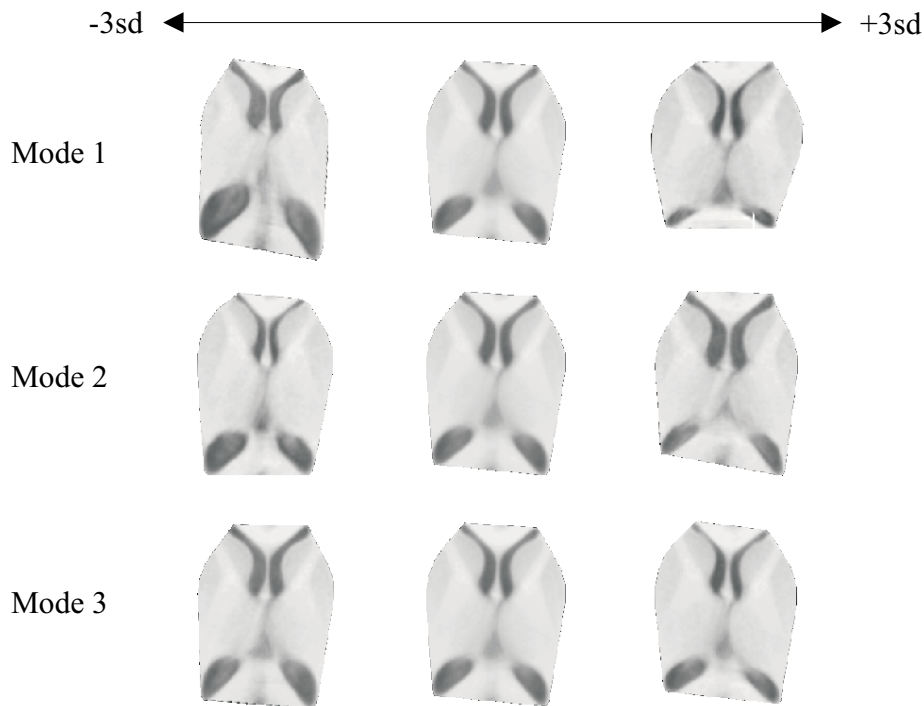
### 9.1 General applicability of AAMs.

Active Appearance Models provide a general approach to image analysis, useful in any situation where 2D view-based models can be constructed. To demonstrate this, we have recently applied the method to images of the brain produced by magnetic resonance imaging (MRI). Figure 9.1 shows some typical images. As can be seen, the original images are quite complex and noisy and thus not suitable for data-driven segmentation.



**Figure 9.1:** Example images of MRI brain slices with landmarks overlaid.

We used a set of 73 training images to build an Active Appearance Model. Figure 9.2 shows the first three modes of variation of the resulting model. Note that the resolution is apparently higher than that of the original images. This occurs because the shape-free grey-level part of the model was built by interpolating over a more closely spaced grid than the original data. The shape-normalisation step ensures that the interpolated measurements are valid.



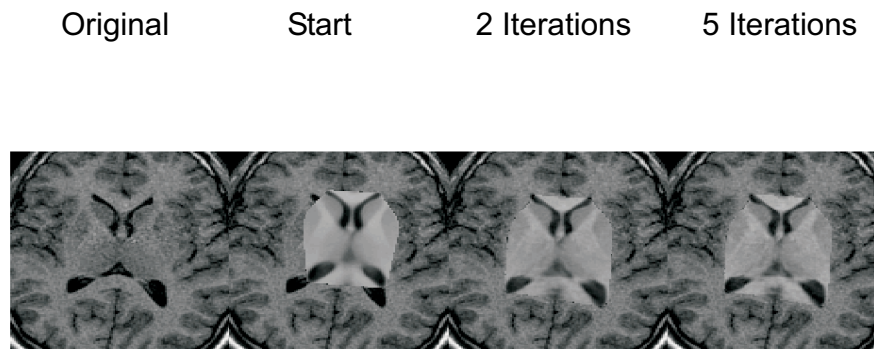
**Figure 9.2:** First 3 modes of variation of brain model.

Figure 9.3 shows some stages of Active Appearance Model search on a previously unseen example image. Despite the noisy nature of the image, the search is successful, even from a fairly poor starting approximation.

## 9.2 Automatic landmarking

One of the main difficulties of Appearance Models is the need to hand-landmark a large set of training images. An automatic or semi-automatic scheme would confer obvious benefits. A clue to how this might be achieved is found by analysing some examples of brain segmentation using the above model. To test the Active Appearance Model we reapply it to the *training data* and measure its performance. In fact, a true performance measure is not easy to find. In Active Shape Model search, a



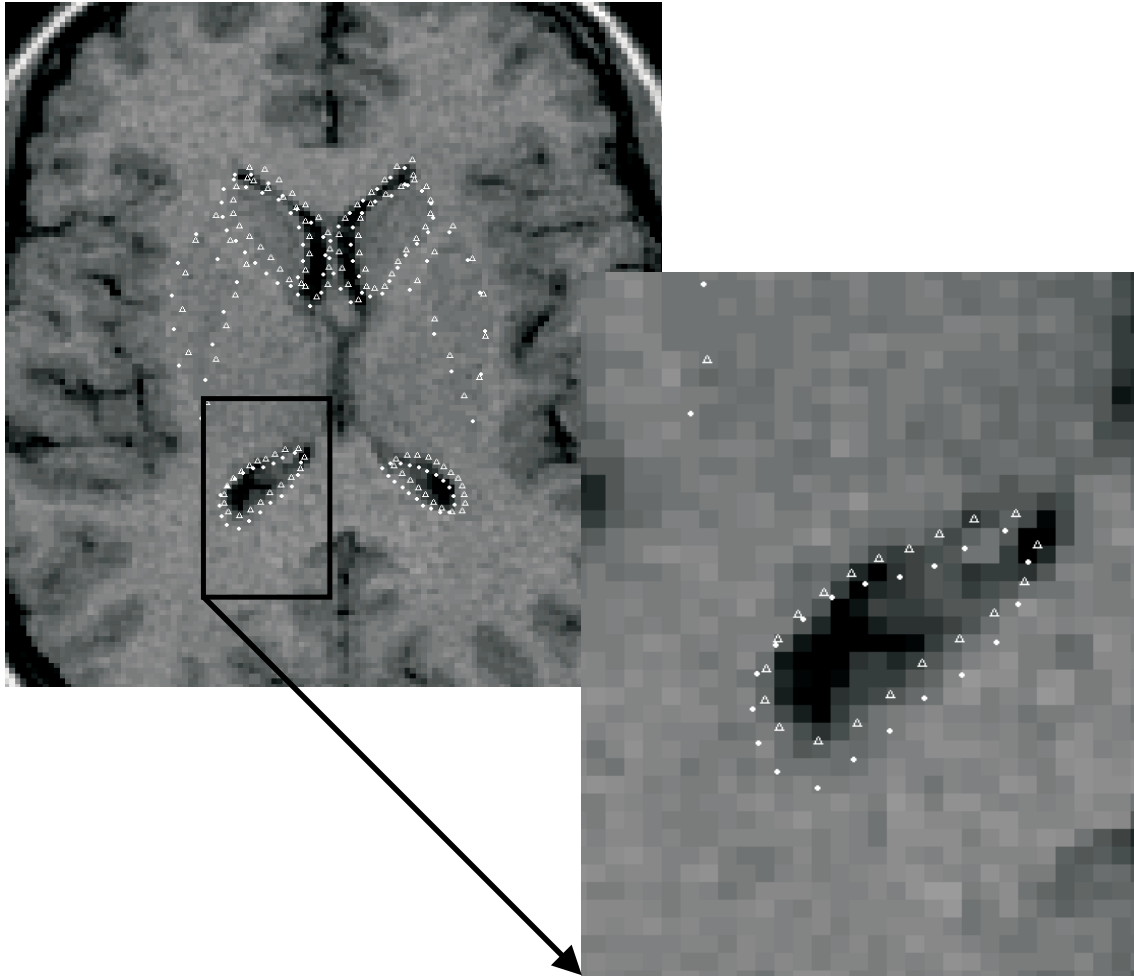


**Figure 9.3:** AAM search applied to a previously unseen brain image.

common measure of performance is the distance between the hand-placed landmark points and the points found by Active Shape Model search. This does not, however, account for the fact that the original landmarks may be badly placed, and therefore the automatically located points might actually be better placed. When analysing the performance of an Active Appearance Model, the problem is even more acute, since the AAM’s performance includes its ability to match to the texture of the image.

A particularly attractive property of AAMs is their ability, under certain circumstances to find *better* landmarks than the original hand-placed landmarks. The following example of AAM search using the brain model shows this clearly. Figure 9.4 shows a small detail of a brain image from the training set, together with the hand-placed landmarks for that feature, which are indicated by circles. Clearly, the landmarks ought to have been placed around the dark shape, and on average, *over the whole training set*, that *is* where they were placed, whereas in this particular image the mark-up has been done badly. If we run the AAM on this image (remembering that this is one of the training set) the result is extremely interesting. The position of the model points after performing AAM search are shown as triangles. In this case, the model has placed the landmark points where the operator ought to have placed them. By learning over the whole training set, roughly what the various regions ought to look like, the model is able to reinterpret the training images and

correct errors made by the operator who marked the images.



**Figure 9.4:** Detail of training image - in this particular case, the landmarks (circles) are badly placed whilst the result of AAM search (triangles) is closer to the desired position.

This then presents the possibility of an iterative mark-up scheme - we can use the re-estimated positions of the landmark points to rebuild the Appearance Model, with fewer mark-up errors than in the original. This sort of iterated scheme may form the basis of an automatic landmarking method. The principle by which it works is the noise reduction obtained by Principal Component Analysis. Variation due to random misplacements of the landmarks is interpreted as noise and thrown out during modelling. This will only work in the case of randomly misplaced landmarks;

systematic misplacements will be interpreted as valid landmark positions.

## 9.3 Extending models to colour

This thesis has concentrated on the analysis and synthesis of grey-scale face images. We would also like to extend the method to analyse colour images. In general, we would always like to use as much image information as possible. If three input channels (red, green, and blue) are available, we should use them all. The more information is used, the greater the likelihood of building a specific model. There is only a limited range of colour variation that is legally allowed in human faces. If we correctly encapsulate the variation then we have an even more specific model than our grey-scale model. In fact it has been shown that colour provides a very powerful specificity constraint when applied to face images. Raja *et al* [79] have built detailed models of skin colour distribution and used them to detect skin coloured regions in images, as part of a face analysis system. We propose a more powerful model; by incorporating colour into an Appearance Model, we not only model the global colour variation, but the legal range of spatial distribution of skin colour for faces. For example, a single global colour model would include both black and white skin colours. However, a Colour Appearance Model goes further by rejecting potential face-like regions in which the distribution of colours is illegal, for example a black forehead and white cheeks (perhaps missing the odd chimney sweep).

Since Appearance Models can be used for synthesis and animation, it is essential that they can be produced in colour - grey-scale images would not suffice for most modern media applications. By correctly encapsulating colour variation in training images, we can ensure that synthetic reconstructions show plausible colours and spatial distributions of colours.

We have built an experimental colour model with a small set of training data, consisting of 24 images of different individuals captured in different conditions using

a colour camera. The formulation is identical to that of the Appearance Model described in Chapter 4, with the exception that the extracted vector of grey-scale values is replaced with a combined vector of red, green and blue values. Thus, instead of the grey-level sample vector,  $\mathbf{g}_{im}$ , we have a colour sample vector,  $\mathbf{g}_{im}(rgb)$ , given by:

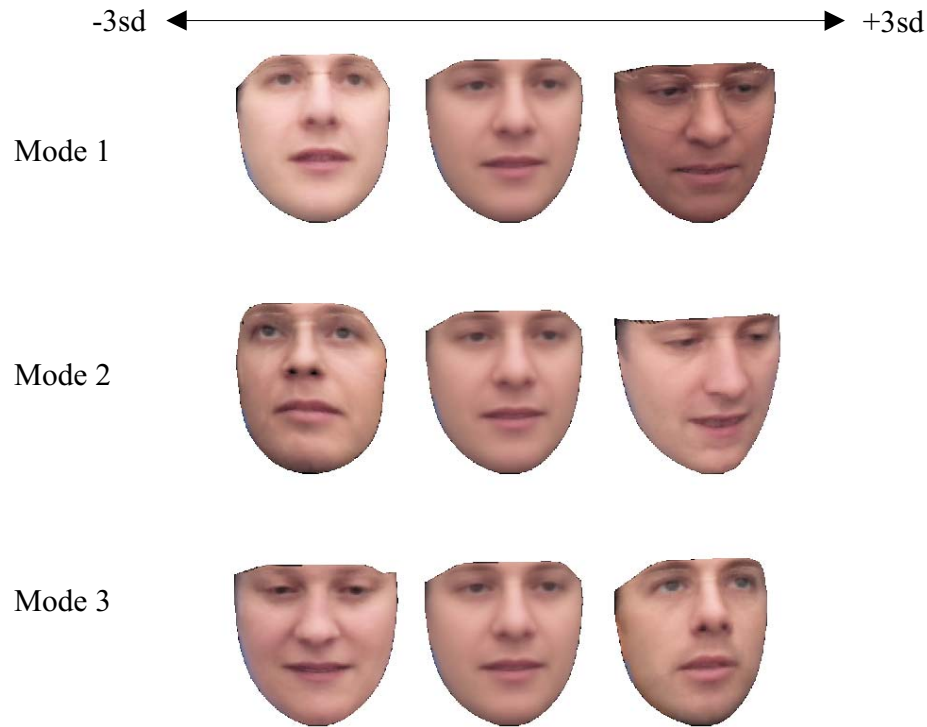
$$\mathbf{g}_{im}(rgb) = (r_1, r_2, \dots, r_n, g_1, g_2, \dots, g_n, b_1, b_2, \dots, b_n) \quad (9.1)$$

where  $r_i$ ,  $g_i$  and  $b_i$  are the pixel values extracted from the red, green and blue channels after warping. All the remaining stages are as in Section 4.2, including the normalisation step. The shape-free region model should encapsulate correlation between the colour channels. Reconstruction using colour models is exactly the same as for grey-scale models, except that the reconstructed vector,  $\mathbf{g}_{im}(rgb)$ , is separated into its three channels to display the image. In this way we can visualise the modes of variation of the Colour Appearance Model in the same way as for the grey-scale model. Figure 9.5 shows the first three modes of variation of the Colour Appearance Model.

At this stage we have not included colour in the Active framework, mainly due to a shortage of appropriate data. We anticipate the performance to be at least as good, and possibly better than grey-scale models, due the further increase in specificity provided by colour. In particular, we expect greater accuracy around regions such as the eye, where the contrast in colour images is greater than in grey-scale images.

## 9.4 Recognising expression

Although many applications of face interpretation are concerned with identity recognition we are also interested in the ability of interpretation systems to recognise expressions. Expression recognition is a more difficult problem to define than identity



**Figure 9.5:** First 3 modes of variation of a colour face model.

recognition. Firstly, a ‘ground truth’ is harder to define - how many types of expression are there? What does it mean to talk about ‘distance’ between expressions? Moreover, expression is a dynamic phenomenon, there are probably limitations to human expression recognition performance from photographs (there was certainly confusion amongst the 25 observers who classified the images for the expression model). Despite these problems we devised a preliminary expression recognition experiment based on 400 images especially captured for psychological expression recognition experiments. We then attempted to use the Active Appearance Model to assign expression labels to each of the 400 images. Some typical examples from the set of ‘expression’ images are shown in Figure 9.6.

In order to evaluate the performance of the Active Appearance Model, we tested the system against 25 human observers. Each observer was shown the set of 400 face images, and asked to classify the expression of each as one of: *happy*, *sad*,



**Figure 9.6:** Typical examples of face images used to evaluate expression recognition performance.

*afraid, angry, surprised, disgusted, neutral.* We then divided the results into two separate blocks of 200 images each, one used for training the expression classifier and the other used for testing. Since there was considerable disagreement amongst the human observers as to the correct expression, it was necessary to devise an objective measure of performance for both the humans and the model. A leave-one-out based scheme was devised thus: Taking the 200 test images, the human observers attached a label to each. This label was then compared with the label attached to that image by the 24 *other* observers. One point was scored for every agreement. In principle this could mean a maximum score of  $24 \times 200 = 4800$  points, however, there were very few cases in which all the human observers agreed, so the actual maximum is much less.

In order to give a performance baseline for this data, a score was calculated several times by generating random choices. The other 200 images were used to train an expression classifier based on the model parameters. This classifier was then tested on the same 200 images as the human observers. The results were as follows:

Random choices score	660	+/- 150
Human observer score	2621	+/- 300
Machine score	1950	

Although the machine did not perform as well as any of the human observers, the results encourage further exploration. The AAM search results were accurate, and we have demonstrated that ID recognition performance is good. This suggests that the simple linear classifier we used limited performance. Further work will need to address a more sophisticated model of human expression characterisation.

## 9.5 Extending the representation

The full face models we have shown are limited to a pose range of around +/- 20 degrees. The limitation is caused by the need to place consistent landmarks on key features over the training set. As pose change increases, correspondence is harder to establish and becomes impossible as features are occluded. This in turn makes the shape normalisation impossible and thus prevents the construction of a specific shape-free region model. Since typical sequences will involve pose changes beyond 20 degrees, this represents a serious limitation.

There are several possible ways to address this problem. The most obvious, but potentially most difficult is to use a 3D model of the face surface. Encapsulating all possible 3D variation would require a large amount of training data that would be more difficult to gather than the existing 2D images. Publicly available 3D data for faces is slowly becoming available and might be used in the future to provide some

degree of flexibility to a rigid 3D template. This 3D model could be used to estimate the 2D projection of faces into images under large pose changes. The key point is that the Active Appearance Model algorithm ought still to work, as long as there exists a method of relating displacements of the 3D model parameters to differences between the model projection and the target image.

A second method of dealing with larger pose variation would be to use multiple 2D models. A first model would represent up to say 20 degrees of variation from frontal, with a further one or maybe two models taking the head all the way to 90 degrees and possibly slightly beyond. For these purposes it is probably sufficient to assume that faces are *on average* symmetrical and build only one model for say, left handed rotation using the reflected version for right handed rotation. This method would require more sets of training images for the different models, and the major difficulty is the integration of the models into a single framework. Ideally the interpretation would deal with images in a smooth manner, rather than constantly switching between models.

The third method we have considered also addresses another problem with the representation, that of missing features, even in frontal images. The most common features whose existence in the image is not certain are the teeth and nostrils, although we also need to deal with the opening and closing of eyes. Rather than attempt to explain the teeth, nose and eyes in 3D we would rather build the concept of ‘visibility’ into the model. This is a similar idea to the ‘z-buffer’ method in computer graphics. It is possible that such a representation could also deal with the variation in visibility due to head rotation.

## 9.6 A half-face model

The problems in dealing with a full pose range occur because establishing correspondence is impossible once features have disappeared. We note however, that it is only



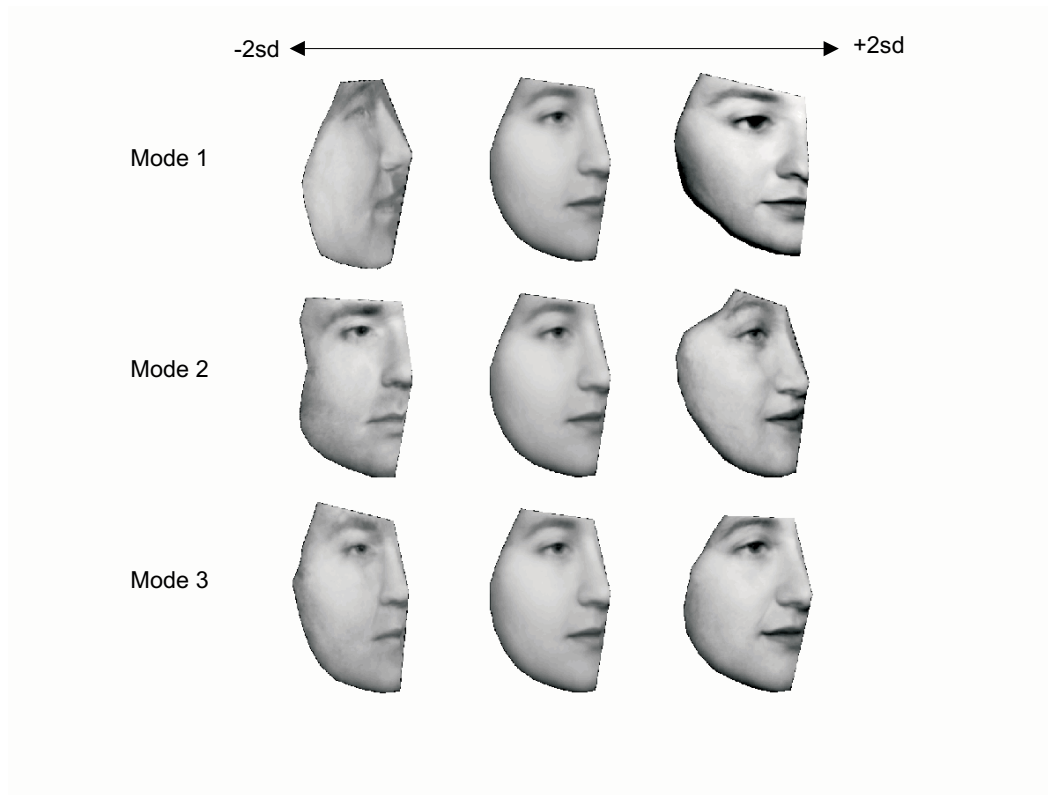
features *on one half of the face* that tend to disappear with pose changes. Correspondence can still be established for points on the visible half of the face. Using this observation, we have constructed a small model by placing landmarks on just one side (the person's right) of face images. The training set consists of 23 images in a full range of pose angles. Figure 9.7 shows a selection of images from the training set along with hand-placed landmarks\*. We have used this small set of images to investigate the feasibility of a model displaying full pose variation; the result is encouraging. Figure 9.8 shows the first three modes of variation of this model. The same model, reflected through 90 degrees would account for the other side of the face.



**Figure 9.7:** Examples of face images used to build a half face model.

---

\*These images and landmarks were kindly provided by Dr. Nicholas Costen, also of the Wolfson Image Analysis Unit.



**Figure 9.8:** First three modes of variation of half face model.

### 9.6.1 Detecting faces

For AAM image search to be successful the model must first be placed reasonably close to the object of interest. Given a particular application, there exist several techniques for generating such initial hypotheses. For example, in a face tracking application, we might initialise the system by detecting regions of the image containing skin coloured pixels, or where motion is detected. Clearly, not all of these cues will be faces, but the system might have the capacity to check many hypotheses. A more satisfactory solution would be to use the Active Appearance Model itself to detect face regions in images. As is shown in Figures 6.2 and 6.3, if the model is close to a solution, the first iteration of the AAM shows a large reduction in reconstruction error from that of the initial placement. By applying one iteration of AAM search initialised over a grid of image locations, it should be possible to rapidly analyse the image for the presence of faces. The grid spacing should be such that any actual face would

be within the capture range of the model for at least one starting position. An extra consideration is the need to search at a range of scales and possibly angles. The use of a lower resolution model would improve the speed at which this detection could take place, although the reduced specificity is likely to increase the number of false-positive hypotheses returned. In a typical system, we envisage such a detection scheme running as a background task, with a full, high-resolution model used to further analyse suggested hypotheses.

### 9.6.2 Dealing with occlusion

Unlike Active Shape Models, Active Appearance Models are not robust to occlusion. This is due to their specific representation; a face with a piece missing is not, according to the model, strictly a face. In any scheme, missing data will reduce the probability that a region is a face, but it would be desirable if image search was still possible in occluded images. In particular, the ideal model would still return the best fit to all the face-like data. Dealing with occlusion is an area where further research is necessary.

### 9.6.3 Efficiency of AAMs

Whilst AAMs are an extremely efficient method of high dimensional optimisation, the fastest current implementation only allows face tracking at around 4 frames per second. This is not fast enough for practical tracking applications, thus we seek methods of speeding up the AAM algorithm still further. One possibility is to use sparse sampling of the image to predict the model correction, rather than every pixel in the model. This sounds like simply using a lower resolution model; the difference is, however, that we can *choose* the set of pixels to sample at each iteration - using those with the most predictive power. The addition of a small element of random sampling would ensure that over a sequence, all the pixels play a part in the error

correction. Cootes *et al* [19] have published preliminary results indicating that sub-sampling can yield a 3-fold increase in speed, although the search proves less reliable in certain circumstances.

#### 9.6.4 Dynamic models

Specificity is the key requirement of models; this is the property that makes AAMs so powerful. During tracking we have demonstrated how specificity can be further improved by partitioning the model into identity and non-identity components. Despite the improvement, this remains a fairly limited dynamic constraint. There certainly exist a larger number of dynamic constraints for face movement in sequences; the laws of Physics limit the movement of muscles in the face. A specific dynamic model would not only be restricted to plausible faces, but plausible face dynamics. Extending the work in this thesis to develop more sophisticated models of dynamics such as those used by Baumberg and Hogg [4] is an area for future research.

### 9.7 Summary

This chapter has given a brief overview of ongoing extensions to the methods presented in this thesis. We have shown an example of the general applicability of AAMs to other types of image. An analysis of the results of AAMs applied to brain images indicates that the model's performance can surpass that of a human operator. This in turn suggests the possibility of semi-automatic methods of placing image landmarks. Appearance Models can be extended to colour in a straightforward manner. We anticipate enhanced performance of colour AAMs due to the further increase in specificity.

We have also shown the application of AAMs to a different type of face interpretation, expression recognition. Whilst the AAM did not perform as well as typical

human observers, the limited experiments provide encouraging results.

Areas for future research include the issue of representation in Appearance Models, seeking in particular a method that allows the representation of parts which may disappear due to large pose changes or occlusion by the lips, etc. Further possible enhancements to the AAM algorithm will include robustness to external occlusion and improvements to the search speed.

Finally, it may be possible to extend our simple dynamic models to build sophisticated, specific models of facial dynamics. The AAM itself could be used to gather the large amount of dynamic training data that would undoubtedly be required to build such models.

# Chapter 10

## Conclusions

This thesis has described the development of unified model-based techniques for the interpretation and synthesis of face images and sequences. In this chapter we summarise the main achievements of the research.

### 10.1 AAMs in machine vision

In Chapter 2 we reviewed several types of models used in face interpretation. Whilst some researchers have concentrated on 3D models, the reduced complexity of 2D models has made their use in face interpretation more common. There exist several commercial face recognition systems based on 2D, view-based models. Most existing techniques have concentrated on either the analysis of shape or of global texture. An early successful attempt to unify the analysis of shape and texture was made by Lanitis *et al* [63], combining Active Shape Models with Shape-Free Grey-Level Models. Lanitis' method only partially addresses the goal of unified interpretation; only the final classifier takes account of the full texture, the initial image analysis seeks to fit a shape model based on small, local region models. Active Appearance Models complete the unification of shape and texture models and provide a single

interpretation method. At the time of writing no other existing technique can provide such fully detailed interpretation with comparable speed. On a typical modern desktop computer, an AAM will converge in typically half a second. The most similar method, Jones and Poggio's *Multidimensional Morphable Models* [55] takes several minutes to fit a model to image data.

### 10.1.1 Appearance Models

The Active Appearance Model is built on the concept of an Appearance Model. In Chapter 4 we explained the construction of these combined models which encapsulate both the shape and texture variation of faces. We demonstrated the ability of a model to generate faces not seen in the training set and importantly, we showed that the model could only generate plausible examples of faces. In a recent extension we have shown how the framework can be easily modified to incorporate colour.

Appearance Models are currently limited to representations where a continuous correspondence field can be specified by hand placed landmarks. This is not possible for a full face beyond a certain range of pose, or for features that may or may not be visible, such as teeth. Furthermore, even in suitable images, the manual location of the landmarks is a laborious task. In Chapter 9 we showed encouraging results which may lead to a workable method of automatic landmark placement. We also suggested various schemes for dealing with large pose ranges and missing features. It is hoped that further development, will allow Appearance Models to deal with such images, and require minimal human effort during model building.

### 10.1.2 Active Appearance Models

The invention of a fast technique for fitting Appearance Models to image data is the key to other developments in this thesis. In our approach, modelling and analysis use a complete representation in a unified framework. Active Shape Models do not

use a complete representation and ASMs plus shape-free texture analysis is not a fully unified approach. The specific nature of an Appearance Model means that if an AAM can be fitted to image data with a small grey-scale (or colour) residual, then we can be confident the region is a face.

The AAM is naturally limited by the representational ability of the underlying Appearance Model. A further limitation is the AAMs current inability to deal with occlusion in images. Further research may yield a new method for dealing with this problem.

AAMs can be applied to many types of image; in Chapter 9 we showed the application of AAMs to MRI brain images. It is anticipated that AAMs will be applied to many image analysis problems.

### 10.1.3 Partitioned Models

An attractive property of Appearance Models is their encapsulation of several types of variation. In the case of faces, this allows us to fit the same model to images of different individuals with various expressions under a range of pose and lighting conditions. In certain situations, this generality can be a handicap. For many types of synthesis and animation, it is desired to manipulate particular ‘real-world’ characteristics of a face independently. For instance, we may wish to change the expression of a face without changing its pose - we would almost certainly not want to change its identity. This is equally important in analysis, particularly the analysis of sequences, where we can be sure that identity is fixed. The full model lacks specificity when used to track multiple frames of the same individual.

We have addressed these issues by partitioning the full model into separate subspaces describing specific types of variation. The most important partitioning is between identity and non-identity, allowing a large increase in specificity for tracking. We have also shown how this partitioning allows successful manipulation of face



images, whilst keeping identity constant.

#### 10.1.4 Interpreting sequences

In Chapters 7 and 8 we introduced a scheme which uses an Active Appearance Model and the identity/non-identity Partitioned Model to track and interpret faces in sequences. The iterative nature of the AAM makes it ideal for tracking; typically it is possible to track a sequence successfully by applying just one iteration per frame. We have thus far achieved a tracking rate of approximately 4 frames per second. The partitioning method uses two complimentary techniques; firstly we attempt to ensure that identity remains fixed during tracking, but also we aim to use any remaining *observed* variation to correct our estimate of the persons identity. This provides a second-order linear correction to the initial estimate of identity. We have only used the method for person recognition in sequences; in the future we intend to develop methods for other types of interpretation such as expression analysis.

Although minimum reconstruction error was achieved with simple tracking, we showed that this scheme produced instability in the estimate of identity. The adaptive tracking scheme based on on-line correction of the identity/non-identity partitioning produced a large improvement in the stability of ID measurement with little degradation in reconstruction accuracy.

Our experimental evaluation of the recognition methods showed that static recognition across large changes in image appearance with a time-delay between gallery and probe images was difficult. By using the adaptive tracking scheme described in Chapter 7 to register both gallery and probe sequences, the recognition performance was shown to improve dramatically - producing results which would make secure access control practicable.

## 10.2 Final statement

In this thesis we have shown how full, generative 2D models can be used practically for image and video analysis. We hope the techniques will see further development and application in this and other areas of computer vision.

# Appendix A

## Warping Face Images

### A.1 Image warping

Suppose we wish to warp an image  $\mathbf{I}$ , so that a set of  $n$  control points  $\{\mathbf{x}_i\}$  are mapped to new positions,  $\{\mathbf{x}'_i\}$ . We require a continuous vector valued mapping function,  $\mathbf{f}$ , such that

$$\mathbf{f}(\mathbf{x}_i) = \mathbf{x}'_i \forall i = 1 \dots n \quad (\text{A.1})$$

Given such a function, we can project each pixel of image  $\mathbf{I}$  into a new image  $\mathbf{i}'$ . In practice, in order to avoid holes and interpolation problems, it is better to find the reverse map,  $\mathbf{f}'$ , taking  $\mathbf{x}'_i$  into  $\mathbf{x}_i$ . For each pixel in the target warped image,  $\mathbf{i}'$  we can determine where it came from in  $\mathbf{i}$  and fill it in. In general  $\mathbf{f}' \neq \mathbf{f}^{-1}$ , but is a good enough approximation.

Below we describe a particular form of  $\mathbf{f}$ , the piece-wise affine interpolator.

Note that we can often break down  $\mathbf{f}$  into a sum,

$$\mathbf{f}(\mathbf{x}) = \sum_{i=1}^n f_i(\mathbf{x}) \mathbf{x}'_i \quad (\text{A.2})$$

Where the  $n$  continuous scalar valued functions  $f_i$  each satisfy

$$f_i(\mathbf{x}_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \quad (\text{A.3})$$

This ensures  $\mathbf{f}(\mathbf{x}_i) = \mathbf{x}'_i$ .

### A.1.1 Piece-wise affine warping

The simplest warping function is to assume each  $f_i$  is linear in a local region and zero everywhere else.

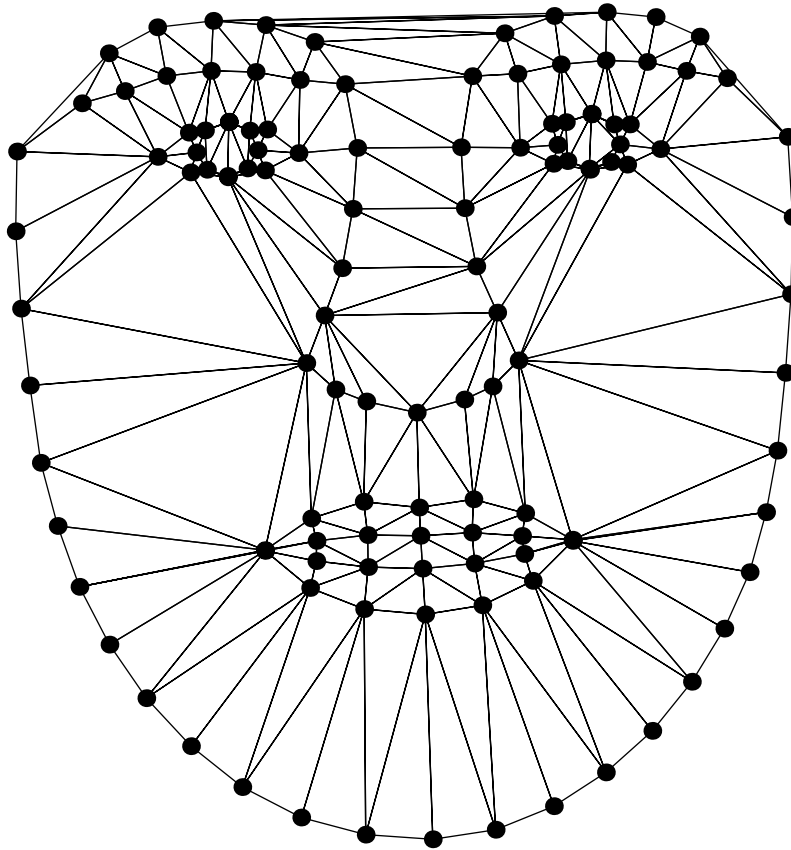
For instance, in the one dimensional case (in which each  $\mathbf{x}$  is a point on a line), suppose the control points are arranged in ascending order ( $x_i < x_{i+1}$ ).

We would like to arrange that  $\mathbf{f}$  will map a point  $\mathbf{x}$  which is halfway between  $x_i$  and  $x_{i+1}$  to a point halfway between  $x'_i$  and  $x'_{i+1}$ . This is achieved by setting

$$f_i(x) = \begin{cases} (x - x_i)/(x_{i+1} - x_i) & \text{if } x \in [x_i, x_{i+1}] \text{ and } i < n \\ (x - x_i)/(x_{i+1} - x_i) & \text{if } x \in [x_{i-1}, x_i] \text{ and } i > 1 \\ 0 & \text{otherwise} \end{cases} \quad (\text{A.4})$$

We can only sensibly warp in the region between the control points,  $[x_1, x_n]$ .

In two dimensions, we can use a triangulation to partition the convex hull of the control points into a set of triangles. An automatic algorithm for the generation of triangles is the *Delaunay Triangulation Method*. Given a set of control points, the method produces a set of triangles such that no data points are contained within any triangle's circumcircle. Figure A.1 shows the result of applying Delaunay Triangulation to the mean shape our Point Distribution Model of faces.



**Figure A.1:** Delaunay triangulation applied to the mean shape of the face PDM.

For the points within each triangle we can apply the affine transformation which uniquely maps the corners of the triangle to their new positions in  $\mathbf{i}'$ .

Suppose  $\mathbf{x}_1$ ,  $\mathbf{x}_2$  and  $\mathbf{x}_3$  are three corners of such a triangle. Any internal point can be written

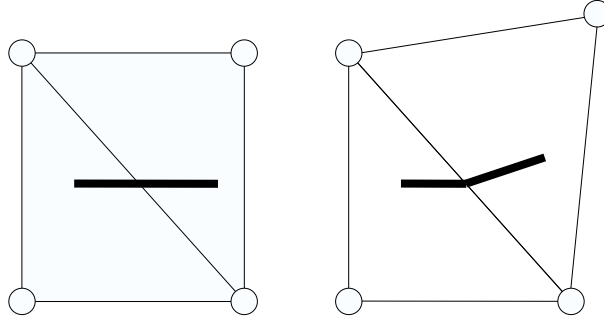
$$\begin{aligned}\mathbf{x} &= \mathbf{x}_1 + \beta(\mathbf{x}_2 - \mathbf{x}_1) + \gamma(\mathbf{x}_3 - \mathbf{x}_1) \\ &= \alpha\mathbf{x}_1 + \beta\mathbf{x}_2 + \gamma\mathbf{x}_3\end{aligned}\tag{A.5}$$

where  $\alpha = 1 - (\beta + \gamma)$  and so  $\alpha + \beta + \gamma = 1$ . For  $\mathbf{x}$  to be inside the triangle,  $0 \leq \alpha, \beta, \gamma \leq 1$ .

Under the affine transformation, this point simply maps to

$$\mathbf{x}' = \mathbf{f}(\mathbf{x}) = \alpha\mathbf{x}'_1 + \beta\mathbf{x}'_2 + \gamma\mathbf{x}'_3\tag{A.6}$$

To generate a warped image we take each pixel,  $\mathbf{x}'$  in  $\mathbf{I}'$ , decide which triangle it belongs to, compute the coefficients  $\alpha, \beta, \gamma$  giving its relative position in the triangle and use them to find the equivalent point in the original image,  $\mathbf{I}$ . We sample from this point and copy the value into pixel  $\mathbf{x}'$  in  $\mathbf{I}'$ . Note that although this gives a continuous deformation, it is not smooth. Straight lines can be kinked across boundaries between triangles (see Figure A.2).



**Figure A.2:** Using piece-wise affine warping can lead to kinks in straight lines.

Piece-wise affine warping is used in all the models presented in this thesis. A

smoother alternative are the thin-plate splines described by Bookstein [13]. These produce deformations that are continuous up to the second derivative, however the computational cost is far greater than for piece-wise affine warping and is not suitable for live analysis. In fact, for face images, thin-plate splines offer no visible improvement in resulting image quality.

# Appendix B

## The Training Images

The 768 training images used to build the models used in this thesis were acquired from several sources.

We used 159 photographs of members of the Wolfson Image Analysis Unit, kindly made available from the earlier work of Andreas Lanitis. This set consists of around 9 examples each of different individuals, with some ethnic variation and an age range of around 18-50.

We used 396 of the 400 images kindly provided by Dr. Jane Whittaker, of the Department of Child Psychiatry, University of Manchester. This set consists of approximately 22 examples each of 19 individuals showing large expression variation. There is some ethnic variation and an age range of around 21-75

We used 150 images provided by The UK Home Office. This set consists of one example of each individual. There is ethnic variation and an age range of around 20-40.

Finally, 63 additional images of current members of the Wolfson Image Analysis Unit were added to include more lighting and pose variation.



Various subsets of the training images and the video sequences used for testing can be obtained by contacting:

The Laboratory Superintendent  
Wolfson Image Analysis Unit  
Stopford Building  
University of Manchester  
Oxford Road  
Manchester  
M13 9PT

# Bibliography

- [1] T. Ahmad, C. Taylor, A.Lanitis, and T. Cootes. Tracking and recognising hand gestures using statistical shape models. *Image and Vision Computing*, 15(5):345–352, June 1997.
- [2] S. Akamatsu, T. Sasaki, H. Fukamachi, N. Masui, and Y. Suenaga. An accurate and robust face identification scheme. In *11<sup>th</sup> International Conference on Pattern Recognition*, pages 217–220, Los Alamitos, California, 1992. IEEE Computer Society Press.
- [3] Y. Aoki and S. Hashimoto. Physical facial model based on 3d-ct data for facial image analysis and synthesis. In *3<sup>rd</sup> International Conference on Automatic Face and Gesture Recognition 1998*, pages 300–305, Nara, Japan, Apr. 1998. IEEE Computer Society Press.
- [4] A. Baumberg and D. Hogg. Generating spatiotemporal models from examples. *Image and Vision Computing*, 14(8):525–532, 1998.
- [5] A. M. Baumberg. *Learning Deformable Models for Tracking Human Motion*. PhD thesis, University of Leeds, 1995.
- [6] A. M. Baumberg and D. C. Hogg. An efficient method for contour tracking using active shape models. Research report series, Division of Artificial Intelligence, School of Computer Studies, University of Leeds, Nov. 1994.
- [7] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *4<sup>th</sup> European Conference on Computer Vision*, 1:45–58, 1996.
- [8] P. N. Belhumeur and D. J. Kriegman. What is the set of images of an object under all possible illumination conditions? *International Journal of Computer Vision*, 28(3):245–260, 1998.
- [9] M. J. Black and Y. Yacoob. Recognizing facial expressions under rigid and non-rigid facial motions. In *1<sup>st</sup> International Workshop on Automatic Face and Gesture Recognition 1995*, pages 12–17, Zurich, 1995.

- 
- [10] A. Blake, R. Curwen, and A. Zisserman. A framework for spatiotemporal control in the tracking of visual contours. *International Journal of Computer Vision*, 11(2):127–145, 1993.
  - [11] A. Blake and M. Isard. *Active Contours*. Springer, 1998.
  - [12] F. L. Bookstein. A statistical method for biological shape comparison. *Journal of Theoretical Biology*, 107:475–520, 1984.
  - [13] F. L. Bookstein. *Morphometric Tools for Landmark Data*. Cambridge University Press, 1991.
  - [14] C. Bregler and S. Omohundro. Non-linear manifold learning for visual speech recognition. In *5<sup>th</sup> International Conference on Computer Vision*, pages 494–499. IEEE Computer Society Press, Los Alamitos, California, 1995.
  - [15] R. Brown and P. Huang. *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley & Sons, Inc., 1992.
  - [16] R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, 1993.
  - [17] A. Burton, S. Wilson, M. Cowan, and V. Bruce. Face recognition in poor quality video: Evidence from security surveillance. *In press: Psychological Science*, 1999.
  - [18] A. Clark and M. Kokuier. A model-based codec with potential for deaf communication. In *12<sup>th</sup> International Conference on Pattern Recognition*, pages 195–197, Los Alamitos, California, 1994. IEEE Computer Society Press.
  - [19] T. Cootes, G. Edwards, and C. Taylor. A comparative evaluation of active appearance model algorithms. In P. Lewis and M. Nixon, editors, *9<sup>th</sup> British Machine Vision Conference*, pages 680–689. BMVA Press, Sept. 1998.
  - [20] T. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *5<sup>th</sup> European Conference on Computer Vision*, pages 484–498. Springer, June 1998.
  - [21] T. Cootes and C. Taylor. A mixture model for representing shape variation. In A. Clarke, editor, *8<sup>th</sup> British Machine Vision Conference*, pages 110–119. BMVA Press, Sept. 1997.
  - [22] T. Cootes, C. Taylor, and A. Lanitis. Active shape models : Evaluation of a multi-resolution method for improving image search. In E. Hancock, editor, *5<sup>th</sup> British Machine Vision Conference*, pages 327–336, York, England, Sept. 1994. BMVA Press.

- 
- [23] T. F. Cootes, D. H. Cooper, C. J. Taylor, and J. Graham. A trainable method of parametric shape description. *Image and Vision Computing*, 10(5):289–294, June 1992.
  - [24] T. F. Cootes, A. Hill, and C. J. Taylor. Medical image interpretation using active shape models: Recent advances. In *14<sup>th</sup> Conference on Information Processing in Medical Imaging, France*, pages 371–372, June 1995.
  - [25] T. F. Cootes, A. Hill, C. J. Taylor, and J. Haslam. The use of active shape models for locating structures in medical images. In H. H. Barrett and A. F. Gmitro, editors, *13<sup>th</sup> Conference on Information Processing in Medical Imaging, Flagstaff, Arizona, USA*, pages 33–47. Springer-Verlag, June 1993.
  - [26] T. F. Cootes, A. Hill, C. J. Taylor, and J. Haslam. The use of active shape models for locating structures in medical images. *Image and Vision Computing*, 12(6):276–285, July 1994.
  - [27] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, Jan. 1995.
  - [28] N. Costen, C. Taylor, and G. Edwards. Derivation of functional axes of variability in a pca framework. In *Submitted to CVPR '99*, 1999.
  - [29] G. W. Cottrel and M. K. Fleming. Face recognition using unsupervised feature extraction. In *International Conference on Neural Networks 1990*, volume 2, pages 65–70, San Diego, 1990.
  - [30] M. Covell. Eigen-points: Control-point location using principal component analysis. In *2<sup>nd</sup> International Conference on Automatic Face and Gesture Recognition 1996*, pages 122–127, Killington, USA, 1996.
  - [31] I. Craw and P. Cameron. Face recognition by computer. In *3<sup>rd</sup> British Machine Vision Conference*, pages 489–507, 1992.
  - [32] I. Craw, D. Tock, and A. Bennett. Finding face features. In *2<sup>nd</sup> European Conference on Computer Vision, Santa Margherita Ligure, Italy*, 1992.
  - [33] 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):1148–1161, 1993.
  - [34] D. DeCarlo and D. Metaxas. Deformable model-based face shape and motion estimation. In *2<sup>nd</sup> International Conference on Automatic Face and Gesture Recognition 1996*, pages 146–150, Los Alamitos, California, Oct. 1996. IEEE Computer Society Press.

- 
- [35] I. Dryden and K. Mardia. General shape distributions in a plane. *Advances in Applied Probability*, 23:259–276, 1991.
  - [36] G. Edwards, A. Lanitis, C. Taylor, and T. Cootes. Statistical model of face images - improving specificity. *Image and Vision Computing*, 16:203–211, 1998.
  - [37] G. Edwards, C. Taylor, and T. Cootes. Interpreting face images using active appearance models. In *3<sup>rd</sup> International Conference on Automatic Face and Gesture Recognition 1998*, pages 300–305, Nara, Japan, Apr. 1998. IEEE Computer Society Press.
  - [38] G. J. Edwards, A. Lanitis, C. J. Taylor, and T. Cootes. Statistical models of face images: Improving specificity. In *7<sup>th</sup> British Machine Vision Conference*, pages 765–774, Edinburgh, UK, 1996.
  - [39] G. J. Edwards, C. J. Taylor, and T. Cootes. Learning to identify and track faces in image sequences. In *8<sup>th</sup> British Machine Vision Conference*, pages 130–139, Colchester, UK, 1997.
  - [40] G. J. Edwards, C. J. Taylor, and T. Cootes. Face recognition using active appearance models. In *5<sup>th</sup> European Conference on Computer Vision*, pages 581–595, 1998.
  - [41] D. Ellis. Trainable methods of mammographic screening. Master’s thesis, University of Manchester, 1997.
  - [42] I. Essa and A. Pentland. Facial expression recognition using a dynamic model and motion energy. In *5<sup>th</sup> International Conference on Computer Vision*, pages 360–367, June 1995.
  - [43] I. A. Essa and A. Pentland. Facial expression recognition using visually extracted facial action parameters. In *1<sup>st</sup> International Workshop on Automatic Face and Gesture Recognition 1995*, pages 35–40, Zurich, 1995.
  - [44] A. Gelb. *Applied Optimal Estimation*. The MIT Press, 1974.
  - [45] S. Gong, S. McKenna, and J. Collins. An investigation into face pose distributions. In *2<sup>nd</sup> International Conference on Automatic Face and Gesture Recognition 1996*, pages 265–270, Los Alamitos, California, Oct. 1996. IEEE Computer Society Press.
  - [46] J. C. Gower. Generalized procrustes analysis. *Psychometrika*, 40:33–51, 1975.
  - [47] D. J. Hand. *Discrimination and Classification*. John Wiley and Sons, 1981.

- 
- [48] J. Haslam. *Model-Based Methods for Medical Image Correction and Interpretation*. PhD thesis, Dept. Medical Biophysics, Manchester University, UK, 1996.
  - [49] T. Heap and D. Hogg. Wormholes in shape space: Tracking through discontinuous change in shape. In *6<sup>th</sup> International Conference on Computer Vision*, pages 344–350, 1998.
  - [50] A. Hill and C. J. Taylor. Model-based image interpretation using genetic algorithms. In P. Mowforth, editor, *2<sup>nd</sup> British Machine Vision Conference*, pages 265–274. Springer-Verlag, Sept. 1991.
  - [51] A. Hill and C. J. Taylor. Model-based image interpretation using genetic algorithms. *Image and Vision Computing*, 10(5):295–300, June 1992.
  - [52] J. Hunter, J. Graham, T. Cootes, and C. Taylor. User programmable visual inspection. In E. Hancock, editor, *5<sup>th</sup> British Machine Vision Conference*, pages 661–670, York, England, Sept. 1994. BMVA Press.
  - [53] A. Jain, L. Hong, and R. Bolle. On-line fingerprint verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):302–313, 1997.
  - [54] R. A. Johnson and D. W. Wichern. *Applied Multivariate Statistical Analysis*. Prentice Hall, 1992.
  - [55] M. J. Jones and T. Poggio. Multidimensional morphable models. In *6<sup>th</sup> International Conference on Computer Vision*, pages 683–688, 1998.
  - [56] M. Kass, A. Witkin, and D. Terzopoulos. Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1987.
  - [57] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *1<sup>st</sup> International Conference on Computer Vision*, London, June 1987.
  - [58] M. Kirby and L. Sirovich. Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):103–108, 1990.
  - [59] A. Kotcheff, A. Redhead, C. Taylor, and D. Hukins. Shape model analysis of thorax radiographs. In *13<sup>th</sup> International Conference on Pattern Recognition*, volume 4, pages 391–395. IEEE Computer Society Press, 1996.
  - [60] M. Lades, J. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburt, R. Wurtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42:300–311, 1993.

- 
- [61] A. Lanitis. *Model-Based Recognition of Variable Objects*. PhD thesis, University of Manchester, 1995.
  - [62] A. Lanitis, A. Hill, T. F. Cootes, and C. J. Taylor. Locating facial features using genetic algorithms. In *International Conference on Digital Signal Processing, Limassol, Cyprus*, pages 520–525, 1995.
  - [63] A. Lanitis, C. Taylor, and T. Cootes. An automatic face identification system using flexible appearance models. In E. Hancock, editor, *5<sup>th</sup> British Machine Vision Conference*, pages 65–74, York, England, Sept. 1994. BMVA Press.
  - [64] A. Lanitis, C. Taylor, and T. Cootes. Automatic face identification system using flexible appearance models. *Image and Vision Computing*, 13(5):393–401, 1995.
  - [65] A. Lanitis, C. Taylor, and T. Cootes. Automatic interpretation and coding of face images using flexible models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):743–756, 1997.
  - [66] A. Lanitis, C. Taylor, T. Cootes, and T. Ahmad. Automatic interpretation of human faces and hand gestures using flexible models. In M. Bichsel, editor, *1<sup>st</sup> International Workshop on Automatic Face and Gesture Recognition 1995*, pages 98–103, Switzerland, June 1995. MultiMedia lab. University of Zurich.
  - [67] A. Lanitis, C. J. Taylor, and T. F. Cootes. A unified approach to coding and interpreting face images. In *5<sup>th</sup> International Conference on Computer Vision*, pages 368–373, June 1995.
  - [68] H. Li, P. Roivainen, and R. Forchheimer. 3d motion estimation in model-based facial image coding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):545–555, 1993.
  - [69] J. Luetttin and S. Dupont. Continuous audio-visual speech recognition. In *5<sup>th</sup> European Conference on Computer Vision*, pages 657–673. Springer-Verlag, 1998.
  - [70] B. F. Manly. *Multivariate statistical methods: A Primer*. Chapman and Hall, London, 1986.
  - [71] B. Moghaddam and A. Pentland. Probabilistic visual learning for object recognition. In *5<sup>th</sup> International Conference on Computer Vision*, pages 786–793, Cambridge, USA, 1995.
  - [72] B. Moghaddam, W. Wahid, and A. Pentland. Beyond eigenfaces: Probabilistic matching for face recognition. In *3<sup>rd</sup> International Conference on Automatic Face and Gesture Recognition 1998*, pages 30–35, Los Alamitos, California, 1998. IEEE Computer Society Press.

- 
- [73] S. C. Nalini K. Ratha, Kalle Karu and A. K. Jain. A real-time matching system for large fingerprint databases. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):799–813, 1996.
- [74] C. Nastar and N. Ayache. Fast segmentation, tracking and analysis of deformable objects. In *4<sup>th</sup> International Conference on Computer Vision*, pages 275–279, Berlin, May 1993. IEEE Computer Society Press.
- [75] C. Nastar, B. Moghaddam, and A. Pentland. Generalized image matching: Statistical learning of physically-based deformations. In *4<sup>th</sup> European Conference on Computer Vision*, volume 1, pages 589–598, Cambridge, UK, 1996.
- [76] M. Okubo and T. Watanabe. Lip motion capture and its application to 3-d molding. In *3<sup>rd</sup> International Conference on Automatic Face and Gesture Recognition 1998*, pages 187–192, Nara, Japan, Apr. 1998. IEEE Computer Society Press.
- [77] A. P. Pentland. Automatic extraction of deformable part models. *International Journal of Computer Vision*, 4(2):107–126, 1990.
- [78] P. Phillips, P. Rauss, and S. Der. Feret(face recognition technology) recognition algorithm development and test results. Technical report no. arl-tr-995, U.S. Army Research Laboratory, Oct. 1996.
- [79] Y. Raja, S. McKenna, and S. Gong. Tracking and segmenting people in varying lighting conditions using colour. In *3<sup>rd</sup> International Conference on Automatic Face and Gesture Recognition 1998*, pages 228–233, Los Alamitos, California, 1998. IEEE Computer Society Press.
- [80] T. Rikert and M. Jones. Gaze estimation using morphable models. In *3<sup>rd</sup> International Conference on Automatic Face and Gesture Recognition 1998*, pages 436–441, Los Alamitos, California, 1998. IEEE Computer Society Press.
- [81] S. Rizvi, P. Phillips, and H. Moon. The feret verification testing protocol for face recognition algorithms. In *3<sup>rd</sup> International Conference on Automatic Face and Gesture Recognition 1998*, pages 48–53, Los Alamitos, California, 1998. IEEE Computer Society Press.
- [82] M. Rydfalk. Candice, a parameterised face. Technical report, Dept. of Electrical Engineering, Linköping University, Sweden, 1987.
- [83] S. Sclaroff and J. Isidoro. Active blobs. In *6<sup>th</sup> International Conference on Computer Vision*, pages 1146–53, 1998.
- [84] C. H. Seal, M. M. Gifford, and D. J. McCartney. Iris recognition for user validation. *British Telecommunications Engineering Journal*, July 1997.



- 
- [85] T. Shakunaga, K. Ogawa, and S. Oki. Integration of eigentemplate and structure matching for automatic facial feature detection. In *3<sup>rd</sup> International Conference on Automatic Face and Gesture Recognition 1998*, pages 94–99, Los Alamitos, California, Oct. 1998. IEEE Computer Society Press.
  - [86] I. Shimizu, Z. Zhang, S. Akamatsu, and K. Deguchi. Head pose determination from one image using a generic model. In *3<sup>rd</sup> International Conference on Automatic Face and Gesture Recognition 1998*, pages 100–105, Los Alamitos, California, Oct. 1998. IEEE Computer Society Press.
  - [87] P. P. Smyth, C. J. Taylor, and J. E. Adams. Automatic measurement of vertebral shape using active shape models. In *7<sup>th</sup> British Machine Vision Conference*, pages 705–714, Edinburgh, Scotland, Sept. 1996. BMVA Press.
  - [88] S. Solloway, C. Hutchinson, J. Waterton, and C. Taylor. Quantification of articular cartilage from mr images using active shape models. In B. Buxton and R. Cipolla, editors, *4<sup>th</sup> European Conference on Computer Vision*, volume 2, pages 400–411, Cambridge, England, April 1996. Springer-Verlag.
  - [89] D. Terzopoulos and D. Metaxis. Dynamic 3d models with local and global deformations: Deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):703–714, 1991.
  - [90] D. Terzopoulos and K. Waters. Analysis and synthesis of facial image sequences using physical and anatomical models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):569–579, 1993.
  - [91] D. Tock and I. Craw. Blink-rate monitoring for a driver awareness system. In *3<sup>rd</sup> British Machine Vision Conference*, pages 518–527, 1992.
  - [92] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.
  - [93] T. Vetter. Learning novel views to a single face image. In *2<sup>nd</sup> International Conference on Automatic Face and Gesture Recognition 1996*, pages 22–27, Los Alamitos, California, Oct. 1996. IEEE Computer Society Press.
  - [94] J. Waite and W. Welsh. An application of active contour models to head boundary location. In *1<sup>st</sup> British Machine Vision Conference, Oxford, England*, pages 407–412, 1990.
  - [95] T. Yokoyama, Y. Yagi, and M. Yachida. Facial contour extraction model. In *3<sup>rd</sup> International Conference on Automatic Face and Gesture Recognition 1998*, pages 254–259, Nara, Japan, Apr. 1998. IEEE Computer Society.

- [96] K. Yow and R. Cipolla. A probabilistic frame for perceptual grouping of features for human face detection. In *2<sup>nd</sup> International Conference on Automatic Face and Gesture Recognition 1996*, pages 16–21, Los Alamitos, California, Oct. 1996. IEEE Computer Society Press.
- [97] A. L. Yuille, D. S. Cohen, and P. Hallinan. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8(2):99–112, 1992.
- [98] W. Zhao, R. Chelleppa, and A. Krishnaswamy. Discriminant analysis of principal components for face recognition. In *3<sup>rd</sup> International Conference on Automatic Face and Gesture Recognition 1998*, pages 336–341, Los Alamitos, California, Oct. 1998. IEEE Computer Society Press.