Dynamic Face Models: Construction and Applications

A thesis submitted to the University of London
for the degree of Doctor of Philosophy

Yongmin Li
Department of Computer Science
Queen Mary, University of London

2001

To Zhanhong:
For making my life so wonderful.

Abstract

Recognising faces across multiple views is more challenging than that from a fixed view because of the severe non-linearity caused by rotation in depth, self-occlusion, self-shading, and change of illumination. The problem can be related to the problem of modelling the dynamics of moving faces from video input for unconstrained live face recognition. In addition, efficiently extracting the non-linear discriminating features of faces is also a largely under-developed problem. To address these three problems, a comprehensive approach to face modelling, detection, tracking and dynamic recognition is presented in this research.

A multi-view dynamic face model, which consists of a sparse 3D shape model learnt from 2D images, a shape-and-pose-free texture model, and an affine geometrical model, is presented in this work. By temporally estimating the model parameters over an image sequence, the identity and geometrical information of a face is extracted separately. The former is crucial to face recognition and facial analysis. The latter is used to aid tracking and aligning faces.

To address the irregular variation of multi-view face patterns, we proposed a view-based approach for face detection. Several face detectors are constructed on small ranges of views. Support Vector Regression is adopted to estimate the head pose, which is used to choose the appropriate face detector. A hybrid method of Support Vector Machines and eigenface is developed which is capable of achieving the best balance between detection accuracy and speed.

Kernel Discriminant Analysis (KDA) is developed to compute the most significant non-linear basis vectors with an intention of maximising the between-class variance and minimising the within-class variance. We applied KDA to the

problem of multi-view face recognition, and a significant improvement has been achieved in accuracy and reliability.

The identity surface of each face class is constructed in a discriminating feature space from a sparse set of face patterns. Using identity surfaces, face recognition can be performed dynamically by matching an object trajectory constructed from a tracked face over time, and a set of model trajectories constructed on the identity surfaces. Experimental results indicate that this approach achieves a more robust performance since the trajectories encode the spatio-temporal information and contain accumulated evidence about the moving faces in a video input.

Declarations

Some parts of the work presented in this thesis have been published in the following articles:

[1] Y. Li, S. Gong, and H. Liddell. Recognising trajectories of facial identities using Kernel Discriminant Analysis. In *The Twelfth British Machine Vision Conference*, Manchester, UK, September 2001.

[2] Y. Li, S. Gong, and H. Liddell. Modelling faces dynamically across views and over time. In *The Eighth IEEE International Conference on Computer Vision*, pages 554–559, Vancouver, Canada, July 2001.

[3] Y. Li, S. Gong, and H. Liddell. Constructing structures of facial identities on the view sphere using Kernel Discriminant Analysis. In *The Second International Workshop on Statistical and Computational Theories of Vision*, Vancouver, Canada, July 2001.

[4] Y. Li, S. Gong, and H. Liddell. Video-based online face recognition using identity surfaces. In *The Second IEEE International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pages 40–46, Vancouver, Canada, July 2001 (**Best Paper Prize**).

[5] Y. Li, S. Gong, and H. Liddell. Exploiting the dynamics of faces in spatial-temporal context. In *The Sixth International Conference on Control, Automation, Robotics and Vision*, Singapore, December 2000.

[6] Y. Li, S. Gong, and H. Liddell. Recognising the dynamics of faces across multiple views. In *The Eleventh British Machine Vision Conference*, pages

242–251, Bristol, UK, September 2000.

[7] Y. Li, S. Gong, J. Sherrah, and H. Liddell. Multi-view face detection using support vector machines and eigenspace modelling. In *The Fourth International Conference on Knowledge-Based Intelligent Engineering System & Allied Technologies*, pages 241–244, Brighton, UK, August 2000.

[8] Y. Li, S. Gong, and H. Liddell. Support vector regression and classification based multi-view face detection and recognition. In *The Fourth IEEE International Conference on Automatic Face & Gesture Recognition*, pages 300–305, Grenoble, France, March 2000.

Acknowledgements

I am greatly indebted to my supervisor, Prof. Shaogang Gong, for introducing me to computer vision and for his enthusiastic supervision over the last three years. Meanwhile, I would like to express my deep gratitude to my second supervisor, Prof. Heather Liddell, for her continued support and encouragement during the three years. Most of the results in this thesis can be traced back to the productive and inspirational discussions with them. Also thanks to them for agreeing to fund several pleasurable trips to conferences and workshops.

I am grateful to Dr. Dennis Parkinson and Dr. Mounia Lalmas for their penetrative and inspiring comments on the work. Thanks also to them for proof-reading the thesis.

I would like to thank other members of the Machine Vision Group - Jamie Sherrah, Eng-Jon Ong, Ting-Hsun Chang, Paul Verity, Jeffrey Ng, Andrew Anderson, Keith Anderson and Peter McOwan, for three years' supply of friendly, productive working environment and much stimulating discussions.

My thanks also go to technical staff, Keith Clarke, David Hawes, Simon Boggis, Matt Bernstein, Derek Coppen, Colin Barnard, Colin Powell and Thomas King, for solving the innumerable hardware/software problems that cropped up during the course of the work, and the secretaries, Gill Carter, Joan Hunter, Carla Benjamin and Sue White, for their friendly and generous help during the three years.

Most of all, I would like to express my deepest gratitude to my wife, Zhanhong. It has been a difficult time for both of us to pursue the PhD study at the same time. Without her sacrifice, support and encouragement, there would never be any chance for this thesis to happen.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The human face, which is often referred to as "the organ of communication" (Bruce and Young, 1998), provides a bewildering variety of important signals in our social lives, for example, its bearer's identity, gender, age, emotion, and interest.

Despite the fact that human faces are essentially similar, we are very skilled at recognising the identities of people from their faces. We can perform this task very easily and it is a basic and important social act although we are still puzzled with the psychological and physiological nature of the process.

As early as in 1883, Galton (Galton, 1883) expressed this process as follows:

> *The difference in human features must be reckoned great, inasmuch as they enable us to distinguish a single known face among those of thousands of strangers, though they are mostly too minute for measurement. At the same time, they are exceedingly numerous. The general expression of a face is the sum of a multitude of small details, which are viewed in such rapid succession that we seem to perceive them all at a single glance. If any one of them disagrees with the recollected traits of a known face, the eye is quick at observing it, and it dwells upon the difference. One small discordance overweights a multitude of similarities and suggest a general unlikeness; just as a single syllable in a sentence pronounced with a foreign accent makes one cease to look upon the speaker as a countryman.*

Since then, the sources of information used in face recognition have been carefully explored, giving useful insights into how we achieve this feat (Bruce and Young, 1998). The psychological and physiological studies of this subject undoubtedly facilitate the effort on developing computer based face recognition systems.

1.1 Face Recognition

As well as the studies in psychology and physiology, face recognition has been emerging as an active research area in computer vision over the past decade. There are numerous potential applications of this research, for example:

- Biometrics: ATM authentication, access control, criminal identification, immigration control, voter registration, entitlement programs
- Surveillance: bank/store security, intelligent search, suspect tracking, post-event analysis
- Human-computer interaction: system logon, smart desktop
- Video-mediated communication: video conferencing, visually mediated interaction, remote teaching, computer supported cooperative work
- Content based access of images and video database: analysis, indexing, retrieval, classification, summarisation

According to (Gong et al., 2000), given a database consisting of a set, $\mathcal{S}$, of N known people, four tasks of face recognition can be defined as follows:

1. Face classification: to identify a subject under the assumption that the subject is a member of $\mathcal{S}$.
2. Known/Unknown: to decide if the a subject is a member of $\mathcal{S}$.
3. Identity verification: to confirm the information measured from a subject with the identity supplied by other means.

4. Full recognition: to determine if a subject is a member of $\mathcal{S}$, and if so to determine the subject's identity.

Actually, full recognition is a combination of the first two tasks, known/unknown determination and face classification. Meanwhile, identity verification is equivalent to known/unknown determination when $N = 1$. Unless otherwise specifically noted in this thesis, the term face recognition is referred to as the first task, i.e. face classification.

Faces exhibit wide variability due to intrinsic and extrinsic factors. The intrinsic sources of variation include identity, sex, age, expression and speech, while the extrinsic ones may come from the changes of viewing geometry, illumination, imaging process and influence of other objects such as occlusion, shadowing, reflection and refraction (Gong et al., 2000).

It is important to note that there are different perceptual tasks corresponding to each of these sources of variation, e.g. gender classification, age estimation, and expression recognition. Understanding these different sources of variation is useful for performing a specific task.

We mainly focus on the issue of face recognition in this thesis. However, we do not intend to exclude the models and methodology presented here from being applied to other perceptive tasks. It is worth pointing out that analysis of these different sources of variation is also very important to the task of face recognition since a robust recognition can only be achieved by maximising the variation from identities and minimising the variation from other sources.

1.2 Approach

The aim of this research is to address the problem of face detection, tracking and recognition dynamically in a spatial-temporal context, where faces undergo large rotation in depth, change in scale, and transformation in position. In particular, the following issues are emphasised:

1. Rather than recognising faces from static images, the dynamics of face ac-

tion, including the global rigid motion of the whole face and the local non-rigid motion of facial features, is modelled and analysed in a spatial-temporal context.

2. Face recognition is performed on a multi-view basis rather than narrow range of views, e.g. frontal view or near-frontal view.

3. Explore different sources of variation of face patterns, and in particular, discriminate the variation from identities and that from views. Develop an efficient non-linear technique to extract the most discriminating features which can maximise the variation from identities and minimise that from other sources.

4. All the related visual tasks, including selective attention, pose estimation, face detection, tracking and face recognition, are integrated systematically into a uniform framework.

Unless stated otherwise, we take image sequences as the input to our system. However, the system is capable of performing face recognition on static images as well.

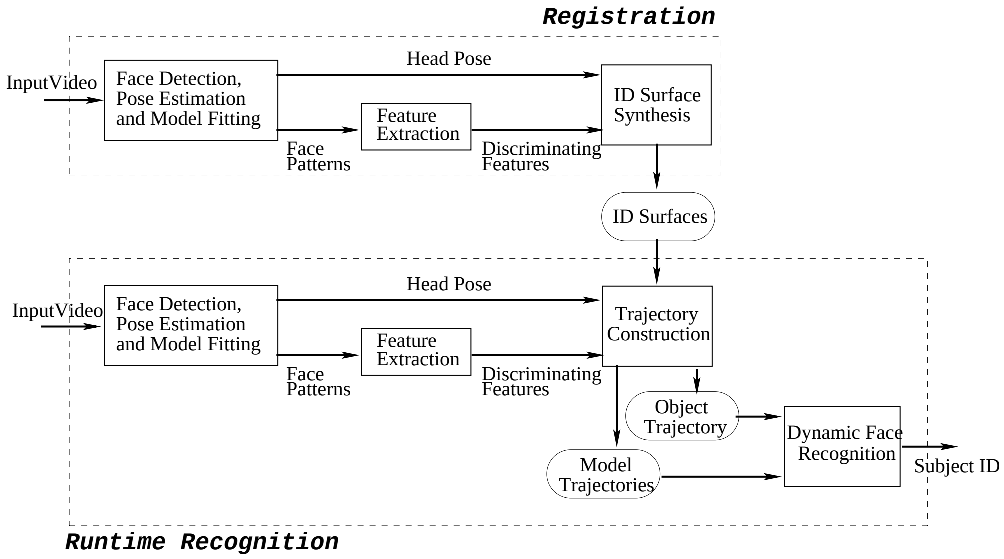


Figure 1.1. The framework for video-based dynamic face recognition.

The proposed framework for video-based dynamic face recognition is illustrated in Figure 1.1. The identities of subjects to be recognised should be first registered to the system. This registration of identities can be carried out as follows: perform pose estimation (Chapter 3), multi-view face detection (Chapter 4) and model fitting (Chapter 5) on the learning image sequences containing the faces of a subject, extract the discriminating features from the fitted model parameters of the faces (Chapter 6), then construct the identity surface of the subject using the extracted discriminating features (Chapter 7).

When performing run-time face recognition, the modules of face detection, pose estimation and model fitting are also applied to a video input containing faces of an unknown identity, followed by the discriminating feature extraction. Then an object trajectory is constructed from the discriminating features and pose information of the face. Meanwhile, a set of model trajectories, one of each face class to be recognised, are synthesised on the identity surfaces using the same pose information and temporal order. Finally face recognition is performed dynamically by matching the object trajectory and the model trajectories.

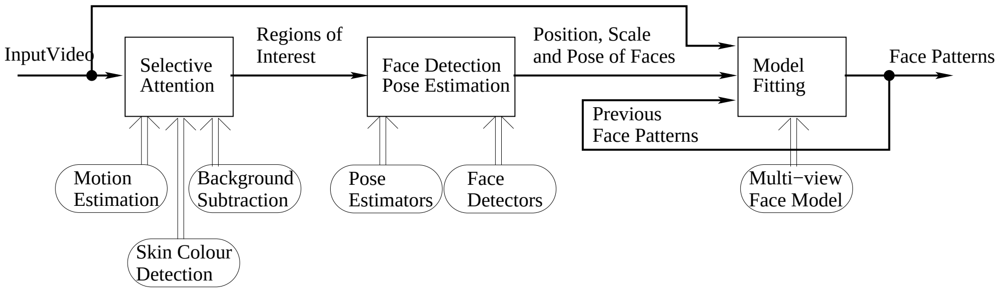


Figure 1.2. Multi-view face detection, pose estimation and model fitting.

The procedure of multi-view face detection, pose estimation and model fitting is illustrated in Figure 1.2.

1. Use motion estimation, skin colour detection, and background subtraction to bootstrap "regions of interest", i.e. sub-images which may contain faces;

2. Detect faces in these regions using multi-view face detectors and pose estimators;

3. Started from the location, scale and pose of the detection, iteratively fit the multi-view dynamic face model.

After successfully fitting the model in a frame of the image sequence, it is not necessary to repeat this whole procedure again for the next frame. For example, one can skip the first two steps, selective attention and face detection and pose estimation, and start model fitting from the parameters obtained in the previous frame. However, the complete procedure is needed when the process is initialised or when tracking fails.

1.2.1 Pose Estimation

The 2D appearance of a face exhibits significant variation when observed from different views. This makes the problem of face detection and recognition more difficult than that from a fixed view, e.g. frontal view. However, if the pose information of a face can be estimated, both the face detection and recognition problems can be simplified to a great extent. The pose information takes a very important role in this work. Firstly, it facilitates the process of multi-view face detection. Secondly, it is used to project the 3D shape model onto a face image, which is essential to model fitting.

Support Vector Regression (Vapnik, 1995; Drucker et al., 1997; Smola et al., 1998) is adopted to learn and estimate head pose. Principal Component Analysis is applied to provide a set of orthogonal basis vectors. The face images are projected to the space spanned by the most significant basis vectors before carrying out pose estimation. This provides a low-dimensional representation for the face images since the number of the significant basis vectors is usually far smaller than the image size.

1.2.2 Multi-View Face Detection

Face detection can be defined as a classification problem - separating face patterns from non-face patterns. For a given image which may contain faces, the exhaustive scan with different sizes of search windows is normally adopted for face detection if no prior knowledge about the faces is available. However, detecting faces across multiple views is more difficult that from a fixed view, e.g. the frontal view. The main reason is that the appearance of a face is significantly different across views. We present a pose estimation based approach in this work.

First, several face detectors are constructed, each on a small range of views. This method proved to be more efficient than using a universal face detector for all views since the face patterns on a small range of views exhibit a more compact distribution and the detector can be constructed with lower complexity.

Second, for a new pattern, either it is a face or not, a coarse pose estimation is performed first to determine which range of views it may fall into. Then the corresponding face detector for this range of views is employed to decide if it is a face. For a non-face pattern, the pose estimate is just a meaningless number, and the corresponding face detector is expected to reject it at the detection stage. It seems that extra computation is applied for pose estimation. However, more computation is saved in detection since only one detector is chosen. Moreover, a fast pose estimator with a coarse precision is sufficient since it is only used to choose a face detector.

Three algorithms for multi-view face detection are implemented in this work: the SVM-based algorithm, the eigenface algorithm, and a hybrid algorithm of SVM and eigenface.

When face detection is performed on image sequences, the motion and skin colour information can be used for selective attention which can significantly improve the real-time performance. Although robust and accurate motion estimation and colour constancy are computationally intensive, the simple techniques of using temporal differencing for motion estimation and mixture of Gaussians for colour detection are sufficient to select and segment the sub-regions of an image which possibly contain faces.

1.2.3 Multi-view Dynamic Face Model

An integrated multi-view dynamic face model is presented in this work which consists of a sparse 3D Point Distribution Model, a *shape-and-pose-free* texture model and an affine geometrical model. The 3D shape model is constructed from a set of 2D multi-view face images with labelled pose and landmarks. To decouple the covariance between shape and texture, the texture model is trained from texture patterns warped to the mean shape at the frontal view.

By fitting this model to a new face, two set of parameters, the identity parameters and the geometrical parameters, are obtained. The former are crucial to face recognition, and the latter are important to face tracking. A stochastic fitting algorithm is developed for the model. The fitting criteria are formulated from the global criterion of the whole face, the local criterion at the landmarks, and the temporal criterion to the landmark appearance in the previous time step.

A Kalman filter based approach provides a temporal estimation of the identity parameters which is more robust and stable over time.

1.2.4 Kernel Discriminant Analysis

The appearance of faces exhibits wide variability due to identities, age, gender, rotation in depth, illumination, etc. A good representation for face recognition should be able to maximise the variation from identities and minimise that from other sources. Usually a feature extraction process is applied on the identity parameters obtained from model fitting before face recognition is carried out.

Kernel Discriminant Analysis, a kernel based non-linear method, is developed to extract the non-linear discriminating features for multi-view face recognition. By applying a non-linear map from the original input space, where face patterns exhibit a severe non-linearity, to a high-dimensional feature space, where the patterns are expected to be linearly separable, one can apply a linear technique, e.g. Linear Discriminant Analysis, in the feature space. Furthermore, the computation can be conducted directly in the original input space through a kernel function which corresponds to the non-linear map. Quadratic Programming techniques can be adopted to compute the problem conveniently and effectively.

1.2.5 Video Based Face Recognition Using Identity Surfaces

Psychological and physiological research showed that a human's ability to recognise animated faces is superior to that on still face images (Knight and Johnston, 1997; Bruce et al., 1998a; Bruce et al., 1998b). For computer based vision systems, when a face is tracked in an image sequence, not only is more information about the face available, but also the dynamics of face action can be captured.

In this work, we propose an approach to modelling face dynamics using *identity surfaces*. Briefly speaking, an *identity surface* is a unique surface in a discriminating feature space for a specific face class. When a face is tracked in a sequence, an object trajectory can be constructed in the feature space. Meanwhile, a set of model trajectories can be synthesised on the *identity surfaces* using the same pose information and temporal order. These trajectories encode the spatio-temporal characteristics of the tracked face, therefore face recognition can be performed dynamically by matching these trajectories.

1.3 Contributions

The original contributions of this thesis include the following:

1. A pose estimation based approach to multi-view face detection has been proposed. A piece-wise multiple model is constructed to separate the problem into a set of sub-problems, each for a small range of views. A pose estimator based on Support Vector Regression is designed to estimate the pose of a candidate face pattern first, then only one of the face detectors is chosen to determine if the pattern is a face.

2. A multi-view dynamic face model has been developed. It consists of a 3D Point Distribution Model, a *shape-and-pose-free* texture model, and an affine

geometrical model. This model can be applied to face images and image sequences where faces undergo large pose variation. A temporal estimation of identity parameters is provided in this model.

3. Kernel Discriminant Analysis, a non-linear method using kernel technique, has been developed to extract the most discriminating non-linear basis vectors for pattern recognition. By using a non-linear map, this method is equivalent to Linear Discriminant Analysis in a high-dimensional feature space. However, computation can be performed conveniently through a kernel function in the original input space.

4. An approach to video based face recognition using *identity surfaces* has been presented in this work. Each face class to be recognised is represented in a discriminating feature space by a unique *identity surface*. The dynamic information of a face tracked in an image sequence can be represented by an object trajectory in this space. Face recognition is then performed by matching this trajectory with a set of model trajectories on the *identity surfaces*.

1.4 Roadmap

This thesis is arranged as follows:

Chapter 2 reviews the previous research on face recognition. The limitations of the previous work and the problems to be addressed in this thesis are also discussed.

Chapter 3 describes an appearance based approach to pose estimation using Support Vector Regression. Principal Component Analysis is adopted to represent multi-view face patterns with a reduced dimensionality.

Chapter 4 discusses the issue of multi-view face detection based on pose estimation. A view based piece-wise detection model is presented, and a coarse pose estimate is used to choose the corresponding face detector.

Chapter 5 introduces a multi-view dynamic face model. This model provides the 3D shape information and the *shape-and-pose-free* texture information to represent facial identity. A temporal estimation of the identity information is also presented in this model.

Chapter 6 introduces the Kernel Discriminant Analysis, a non-linear method for extracting the most discriminating basis vectors for pattern recognition. The results of representing multi-view face patterns using this method are also demonstrated.

Chapter 7 presents an approach to video based face recognition using *identity surfaces*. Each face class is represented by a unique *identity surface* in a discriminating feature space. Face recognition can be performed dynamically by matching an object trajectory obtained from an image sequence and a set of model trajectories on these *identity surfaces*.

Chapter 8 contains the conclusions and a general discussion of the work presented in this thesis.

Chapter 2

Background

The issue of face recognition has been extensively addressed over the past several decades. Various approach have been proposed to solve the problem under different assumptions and conditions. These approach can be approximately categorised into invariant feature based approach, 2D shape based approach, 2D appearance based approach, and 3D structure based approach. These approaches will be reviewed in Section 2.1, Section 2.2, Section 2.3, and Section 2.4 respectively. Video based face recognition has attracted more and more interest in recent years. We will describe the recent work on this issue in Section 2.5. The limitations of previous studies will be discussed in Section 2.6.

2.1 Invariant Feature Based Approach

A classical approach to face recognition is to locate a small set of facial features, such as eyes, nose, mouth, and contour of the face, and then compute the spatial configuration of these features by measurements of positions, distances, angles, and curvatures. These feature are normally chosen to be sufficiently descriptive to the facial characteristics and invariant to some types of variation in facial appearance, for example, rotation, illumination, and expression. Most of the early studies adopted this approach for face representation and recognition.

In one of the early studies, Kanade (Kanade, 1973) presented an automatic feature extraction method based on edge-map projections. Similarly, Kaya and Kobayashi (Kaya and Kobayashi, 1972) used the distances between geometric features, such as eyes, mouth, nose and chin, and their relative positions, to characterise faces.

Craw and Cameron (Craw and Cameron, 1992) represented a face by a set of key points and used the coordinates to normalise the shapes before performing eigen-analysis on the grey level images.

Chen and Huang (Chen and Huang, 1992) extracted the contours of eyes and mouth by a deformable template model and extracted the shapes of eyebrows, nostrils and face using an active contour model in order to carry out the task of frontal-view face recognition.

Brunelli and Poggio (Brunelli and Poggio, 1993) described two algorithms for face recognition, one based on the computation of a set of geometric features, such as nose width and length, mouth position, and chin shape, and the other based on almost-gray-level template matching.

Sinha (Sinha, 1994) addressed the problem of detecting faces under varying illumination. He claimed that the average brightness values of a set of facial regions such as eyes, nose, cheeks, mouth and chin are relatively consistent. Thus faces could be represented by such a collection of relative magnitude values.

Graf *et al.* (Graf et al., 1995) developed a technique to reliably locate facial parts, such as the eyes, nostrils, and lips, and whole faces in images. A band-pass filter is designed to select a range of spatial frequencies, then morphological operations are applied, followed by multi-level thresholding. Possible areas of facial parts are identified by this process. Combinations of such areas are then evaluated with classifiers. The authors claimed that robustness is obtained by combining the evidence of several classifiers.

These approaches provide compact representations (the dimensionality of feature vectors is around 10-50 in most cases) of facial characteristics if the features can be precisely detected. However, they can only provide satisfactory results in high-resolution images and in frontal-view face recognition since they are limited in that:

1. The methodology adopted in this kind of representation is to decompose a complex problem into a set of sub-problems. Unfortunately, these sub-problems, i.e. facial feature detection and analysis, are non-trivial in themselves. Thus these methods may even be less reliable and robust than the simple method of template matching which models the whole face as a single object.

2. The feature points and their geometric configuration are *hard-coded* into the system based on the designers' prior knowledge.

3. Performance can be seriously degraded when dealing with large variation in illumination, pose, cluttered background, or low-resolution images since features cannot be detected robustly or even do not appear in images due to, for example, large rotation or self-occlusion.

2.2 2D Shape Based Approach

2.2.1 Active Contour Models

Early studies on many computer vision problems are dominated by low-level image processing, such as edge or line detection and stereo matching, together with motion tracking, which either seeks to find or are relied upon consistent and significant features in images. It therefore imposes stringent demands on the accuracy and reliability of the recovery of the image-features and grouping of these image-features into structures.

In the seminal paper on Active Contour Models, also commonly referred to as "Snakes", Kass *et al.* (Kass et al., 1987a) proposed the idea of bringing a certain degree of prior knowledge to the problem of image interpretation rather than expecting the desirable properties such as continuity and smoothness to emerge from images by themselves. A Snake is usually defined as an energy-minimising spline guided by external constraint forces and influenced by image forces that pull it toward features such as lines and edges. Scale-space continuation can be used to enlarge the capture region surrounding a feature. The energy functional for fitting a Snake onto an image can be defined as:

$$E^*_{snake} = \int (E_{int}(\mathbf{v}(s)) + E_{image}(\mathbf{v}(s)) + E_{con}(\mathbf{v}(s)))ds \qquad (2.1)$$

where $\mathbf{v}(s) = (x(s), y(s))$ is the feature position in an image, E_{int} represents the internal energy of the spline due to bending, E_{image} gives rise to the image forces, and E_{con} gives rise to the external constraint forces. Briefly speaking, a Snake seeks to converge to the appropriate image features with maximal smoothness and elasticity.

Kass *et al.* applied Snakes to several problems including speaker's lip tracking, where the initial Snake was located manually in the first frame of a video sequence, and after that it tracked the lip movement with high accuracy (Kass et al., 1987a; Kass et al., 1987b). A similar method was reported by Waite and Welsh for head boundary location in images (Waite and Welsh, 1990b; Waite and Welsh, 1990a).

Wu *et al.* (Wu et al., 1996) developed a system to detect human faces, facial features, and face contours from colour images. Multiple Snakes are used to extract face contours, for example, eyebrow, eye or mouth is modelled by 2 contour lines while nose is modelled by 3 contour lines. Colour information based energy terms are defined in this approach for model fitting.

Okubo and Watanabe (Okubo and Watanabe, 1998) presented an approach to lip tracking from video sequences. Optical flow between the present and previous frames is used to determine the initial position of a Snake. A 3-D molding of temporally accumulated lip contours is also formed in this approach for further analysis, for example, lip reading. However, the initial position of the Snake still needs to be manually labelled as in the case of (Kass et al., 1987a; Kass et al., 1987b).

Yokoyama *et al.* (Yokoyama et al., 1998) proposed an improved Active Contour Model for facial contour extraction. The axis-symmetry property is imposed as a global constraint to the energy function. The authors claimed that performance was improved by using different sizes of differential filters and iterative initialisation. However, this approach can only be used to extract facial contours of frontal-view faces.

By defining the energy function appropriately, it is possible for Snakes to incorporate higher-level processes instead of relying solely on low-level image-feature detection. However, there is no problem-specific shape constraints defined in their original form. For example, a Snake designed for face tracking may not converge to a face-like shape since it can be attracted to other edges or lines in an image, especially when the image is taken with cluttered background or significant noise. Another shortcoming of Snakes lies in the fact that the performance crucially depends on the weights given to E_{int}, E_{image}, and E_{con} in the energy function (2.1), for which manual parameter tuning may be inevitable in many applications.

2.2.2 Deformable Shape Models

As stated above, Active Contour Models impose *soft* rather than *problem-specific* constraints to the favoured shapes. By using a parametric shape vector with few degrees of freedom, one can achieve the so-called "Deformable Models", where hard constraints and default shapes of more specific classes of shapes are explicitly defined.

The idea of Deformable Models predates the development of Active Contour Models but has enjoyed a revival inspired by the latter (Blake and Isard, 1998). Please refer to (Fischler and Elschlager, 1973; Burr, 1981; Bookstein, 1989) for details of the early studies.

Among the recent work on Deformable Models, Yuille *et al* (Yuille et al., 1992; Yuille and Hallinan, 1992) proposed a method for detecting and describing features of faces using deformable templates. The features of interest, for example, an eye, is described by a parameterised template with 11 parameters representing the position, scale, rotation, radius of circles and ellipses which are defined to approximate the shape of an eye. An energy function is defined which links edges, peaks, and valleys in the image intensity to corresponding properties of the template. The template then interacts dynamically with the image by altering its parameter values to minimise the energy function, thereby deforming itself to find the best fit. The final parameter values can be used as descriptors for the features.

Craw *et al.* (Craw et al., 1992; Bennett and Craw, 1991) developed a system to locate face and individual face features such as eyes and mouth in grey-level images. This system has two components: modules designed to locate particular face features which are similar to the deformable templates designed in (Yuille et al., 1992; Yuille and Hallinan, 1992), and an overall control strategy which activates modules on the basis of the current solution state, and assesses and integrates the results from each module. Statistical knowledge about the relative positions of 40 pre-specified feature points is obtained by detailed measurements of 1000 example faces. Once an initial location of one feature point has been estimated, predictions about the positions of other features can be obtained from the statistical knowledge. This can lead to a rapid increase in confidence as other features are identified in their predicted positions, or alternatively a quick rejection of the estimated feature location.

Brunelli and Poggio (Brunelli and Poggio, 1993) described a feature based approach to face recognition. Thirty five geometric features are extracted automatically. Integral projections of image edge are used to locate the mouth, nose, eyes and eyebrows, while dynamic programming is adopted for face outline detection. The bilateral symmetry and the layout of human faces serve as constraints for feature location.

Yow and Cipolla (Yow and Cipolla, 1996b) proposed a Bayesian network based probabilistic framework for face detection. Spatial filters are adopted to detect interesting points in images, then face candidates are formed using geometric and grey-level information. Face detection is performed by evaluating a Bayesian network over all face candidates. Besides the six oriented facial features for eyebrows, eyes, nose and mouth, four Partial Face Groups and five horizontal or vertical feature pairs are designed in the shape model, which enable face detection to be performed under scale, orientation, and in particular, viewpoint changes (Yow and Cipolla, 1996a).

The main problem with most of the existing methods for variable object modelling is that they sacrifice model specificity in order to accommodate variability, thereby compromising robustness during image interpretation. To address this problem, Cootes *et al.* (Cootes et al., 1995b) developed the Active Shape Models (ASMs) where the characteristics of a class of objects are learnt from a training set of correctly annotated images. New shapes of the class can be represented by a weighted sum of a small number of significant basis shape vectors. These models can be used for image search through an iterative refinement algorithm analogous to that employed by Active Contour Models. The key difference is that ASMs can only deform to fit the data in ways consistent with the training set (Cootes et al., 1994; Cootes et al., 1995b).

The Active Shape Model has been successfully applied to face recognition. For example, Lanitis *et al.* (Lanitis et al., 1994; Lanitis et al., 1995b; Lanitis et al., 1995c; Lanitis et al., 1995a; Lanitis et al., 1997) introduced a compact parametrised face model which takes into account the variability of individual appearance, 3D pose, facial expression, and lighting. The model is created by performing a statistical analysis over a training set of face images. A robust multi-resolution search algorithm is used to fit the model to faces in new images. This allows the main facial features to be located, and a set of shape and gray-level appearance parameters to be recovered. It has also been used in industrial inspection (Cootes et al., 1995b) and medical image interpretation (Cootes et al., 1995a).

Compared with the classical geometric deformable shape model, the significant advantage of ASMs lies in the fact that the class-specific shape constraints are learnt directly from training examples rather than handcrafted by geometric shapes. However, the intrinsic linear characteristics of ASMs limit their application to small range of shape variation. For example, when faces are undergoing large pose changes, ASMs may be incapable of modelling the resulting severe non-linear variation.

The ASM in its original form corresponds to a single Gaussian density function. When the shape class is too complicated to be modelled as a single Gaussian, a multi-modal density function is needed. Cootes and Taylor (Cootes and Taylor, 1997; Cootes and Taylor, 1999) proposed to estimate the multi-modal density function using mixture of Gaussians. Similar approaches of modelling shape space with severe non-linearity using piece-wise linear sub-models have been developed by Heap and Hogg (Heap and Hogg, 1997; Heap and Hogg, 1998), Bowden *et al.* (Bowden et al., 1998), and Ong and Gong (Ong and Gong, 1999a; Ong and Gong, 1999b).

Edwards *et al.* (Edwards et al., 1996; Edwards et al., 1998a) have described a non-linear ASM built from a Multi-Layer Perceptron. Their experimental results showed that, when the initial placement of the model is bad, image search using the non-linear shape model performs better than that using a linear model. Also, the shape variability of the training images can be explained with as few as half the number of parameters used in the linear model.

Recently, Romdhani *et al.* (Romdhani et al., 1999b; Romdhani et al., 1999a) introduced a non-linear shape model to address the problem of corresponding faces with large pose variation. Kernel Principal Component Analysis (Scholkopf et al., 1996; Scholkopf et al., 1997; Scholkopf et al., 1998b; Scholkopf et al., 1998c; Scholkopf, 1997) is adopted to learn the non-linearity from 2D face images.

2.2.3 Dynamic Shape Models

The Active Contour Models and Deformable Shape Models described above can be fitted in both static images and image sequences. In particular, fitting a shape model to image sequences containing faces is more promising in many applications such as video conferencing, visual interaction and visual surveillance. Under these circumstances, the dynamics of a continuously moving facial shape, for example, the displacement, velocity, acceleration, inertia and viscosity should be modelled as well as the shape parameters and their prior constraints.

Kalman filters (Brammer and Siffling, 1989) have been widely used for this purpose. In brief, a Kalman filter for dynamic shape tracking is a two-phase process comprising “prediction” and “update”. The shape vector of the dynamic model is predicted for the next time-step based on their history values in the first phase, “prediction”. Then the predicted shape is refined using measurement from the image of the new time-step in the second phase, “update”.

Pentland and Horowitz (Pentland and Horowitz, 1991) introduced a physically correct model of elastic non-rigid motion. Because of the small number of parameters involved, this representation is used to obtain accurate overstrained estimates of both rigid and non-rigid global motion. An extended Kalman filter was used to achieve these estimates over time, resulting in stable and accurate estimates of both three-dimensional shape and three-dimensional velocity.

Baumberg and Hogg (Baumberg and Hogg, 1994) demonstrated an approach to articulated non-rigid body tracking using a modal-based flexible shape model and dynamic filtering. An Active Shape Model was generated automatically from real image data and incorporates variability in shape due to orientation as well as object flexibility. A Kalman filter is used to control spatial scale for feature search over successive frames.

Blake *et al.* (Blake et al., 1993) introduced a framework for contour tracking using elastic models and stochastic filtering, where a mechanism is developed for incorporating a shape template into a contour tracker via an affine invariant coupling. In a later study, the authors developed a method for iteratively training of a tracker. Given an ”untrained” tracker, a training motion of an object can be observed over time and stored as an image sequence. The image sequence is used to learn parameters in a stochastic differential equation model. These are used, in turn, to build a tracker whose predictor imitates the motion in the training set (Blake et al., 1995). This method has been used for real-time, unadorned lip tracking (Kaucic and Blake, 1998).

Edwards *et al.* (Edwards et al., 1998c; Edwards et al., 1999) developed a method of identifying and tracking faces in image sequences. Two sets of Kalman filters are used, one for the corrected identity parameters, in which the underlying model of motion is treated as a zeroth order, or constant position model, and another for the residual parameters, where the motion model is assumed to be first order, or constant velocity.

2.3 2D Appearance Based Approach

Modelling faces by their 2D appearances seeks to capture the holistic characteristics of faces rather than a set of individual features. Furthermore, the 3D information can also be modelled implicitly without recovering the 3D structure which is usually computational intensive. Although it can only provide a coarse measurement for the 3D information, real-time efficiency can be achieved which is crucial for many applications. For these reasons, appearance based approaches have been of great interest in face recognition and facial analysis over the last decade. In this section, the recent studies on this issue will be reviewed in three categories: template based models, statistical appearance models, and combined shape and texture models.

2.3.1 Template Based Approach

One of the straightforward approaches is to represent faces by a set of generic templates, then the process of face detection, face recognition, or facial analysis is performed by template matching.

Baron (Baron, 1981) presented an approach to face recognition using raw images as system input. Neural Networks are designed to carry out the processes of encoding visual images into neural patterns, detection of simple facial features, size standardisation, and reduction of the dimensionality of the neural patterns. Recognition is finally performed based on correlation of the resulting sequence of patterns with all model patterns.

Brunelli and Poggio (Brunelli and Poggio, 1993) compared a geometrical-feature-based algorithm with a template-based algorithm. They claimed that the results obtained for the testing sets show about 90% correct recognition using geometric features and perfect recognition using template matching. In their following work, Sung and Poggio (Sung and Poggio, 1994) generated 6 face prototypes and 6 near-face-nonface prototypes as templates to match a new image pattern. A well-tuned Neural Network is employed to synthesise these matching results. Another approach using Support Vector Machines is presented by Osuna and Poggio, where the most representative examples, known as Support Vectors, are extracted automatically (Osuna et al., 1997b; Osuna et al., 1997c).

Template based approaches model faces intuitively without any prior knowledge. Despite its success, its shortcomings are obvious. For example, it is sensitive to variation in illumination, position, rotation, scale and background.

2.3.2 Statistical Approach

Statistical techniques have been widely used in face recognition and facial analysis to extract the abstract features of face patterns. Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and the newly developed Kernel Principal Component Analysis (KPCA) fall into this category. More details of the principles and algorithms of PCA, LDA and KPCA are presented in Appendix A, C and B.

Sirovich and Kirby (Sirovich and Kirby, 1987) first used PCA, also known as the Karhunen-Loeve transform, for face representation. This approach is commonly referred to as the "eigenface" method in the community of face recognition. In principle, the eigenface method seeks to compute a reduced set of orthogonal basis vectors, i.e. eigenfaces, by eigen-decomposing the covariance matrix of the training face patterns. A new face pattern can be approximated by a weighted sum of these eigenfaces. An important property of the eigenface method is that it provides an optimal linear transformation from the original image space to an orthogonal eigen space with reduced dimensionality in the sense of least mean squared reconstruction error.

In their later work, Kirby and Sirovich (Kirby and Sirovich, 1990) used the symmetrical property of human faces to extend the data set and to impose even and odd symmetry on the eigen functions of the covariance matrix without increasing the complexity of computation.

Turk and Pentland (Turk and Pentland, 1989; Turk and Pentland, 1991) used a similar method to code face images and capture face features. Moghaddam and Pentland (Moghaddam and Pentland, 1994; Moghaddam and Pentland, 1995; Moghaddam and Pentland, 1997; Pentland et al., 1994) extended this approach to view-based and modular eigen spaces with an intention of recognising faces under varying views and locating facial features, such as eyes and mouth. In the authors' later work, Moghaddam *et al.* (Moghaddam et al., 1998) modelled two mutually exclusive classes of variation between facial images: intra-personal (variations in appearance of the same individual, due to different expressions or lighting) and extra-personal (variations in appearance due to a difference in identity). The likelihoods for each respective class are learned from training data using eigen space density estimation before used to compute similarity based on the *a posteriori* probability of membership in the intra-personal class, and ultimately used to rank matches in the database. The authors have demonstrated that improved performance is achieved over the eigenface method.

Craw *et al.* (Craw et al., 1995) have comprehensively studied the performance of different PCA based face coding schemes: shape-free normalisation, shape data, and combined shape and texture. They also investigated the performance of using Euclidean distance and Mahalanobis distance. Their experimental results indicated that the eigenface coding of shape-free faces, based on manually coded landmarks, is more effective than the corresponding coding of correctly shaped faces. Also, the use of Mahalanobis distance is clearly more effective than that of Euclidean distance, suggesting that it is not simply the orthogonality properties of the eigenface bases, but their variance properties that aid recognition.

Intrinsically, PCA seeks to find a linear transformation to represent patterns with minimal residual error. This is efficient when the patterns of a problem can be linearly represented or approximated with a uni-modal distribution. However, when severe non-linearity is involved or the distribution of patterns is multi-modal, it would be difficult to obtain a satisfactory solution using PCA. It has been of considerable interest to develop non-linear PCA in an effort to overcome the shortcomings of the linear PCA, for example, the principle curves of Hastie and Stuetzle (Hastie and Stuetzle, 1989) and Tibshirani (Tibshirani, 1992), multi-layer auto-associative neural networks of Kramer (Kramer, 1991), the radially symmetrical kernel approach to non-linear PCA of Webb (Webb, 1996), and the generative topographic mapping of Bishop *et al* (Bishop et al., 1998).

An alternative to this problem, which has been introduced recently, is Kernel Principal Component Analysis (Scholkopf et al., 1996; Scholkopf et al., 1997; Scholkopf et al., 1998b; Scholkopf et al., 1998c; Scholkopf, 1997). The principle of KPCA can be described as follows: By defining a non-linear map from the original space of patterns to a high-dimensional feature space, one expects the projected patterns in the feature space to be sufficiently represented by a linear PCA [1]. However, the computation is not conducted in the feature space where it would be very expensive or even impossible. By using a kernel function, all the computation is performed conveniently in the original space.

When faces are undergoing large pose change out of the image plane, the rotation in depth, self-occlusion and self-shading impose a severe non-linearity to the appearance of the faces. Romdhani *et al.* (Romdhani et al., 1999b; Romdhani et al., 1999a) have applied KPCA to model the non-linearity of facial appearance. Significant performance improvement over the linear PCA based models such as ASMs and AAMs has been reported.

Provided the face images are well-registered, the PCA based approaches can effectively extract the most significant abstract features, i.e. Principal Components or eigenfaces, which capture most of the variance of training face examples and dramatically reduce the dimensionality of face patterns. However, it is important to note that these features are "global" which can be important for applications such as face detection, but less relevant to the task of face recognition since no

[1] It is important to note that no research has reported yet that, to what extent, the linear mechanism is valid in the feature space.

mechanism is provided for the selection of features related to discriminate different facial identities. Swets and Weng (Swets and Weng, 1996) argued that the eigenfaces are only Most Expressive Features which are inefficient for face recognition, and a subsequent discriminant analysis projection is needed to derive the Most Discriminating Features.

LDA (Fisher, 1938; Fukunaga, 1972) seeks to find a linear transformation by maximising the between-class variance and minimising the within-class variance. Computationally, LDA can be solved as an eigen-decomposition problem (see Appendix C for details).

LDA proved to be an appropriate technique for face recognition. Swets and Weng (Swets and Weng, 1996) applied the LDA technique to retrieve the Most Discriminating Features of faces, and compared the performance with that of PCA. Their experiments showed an improved performance with the LDA method when large number of training images of each face class are available. However, with fewer training images, the performance between PCA and LDA is smaller due to the fact that the within-class variation may not be captured sufficiently. In a later work, the authors proposed a self-organising framework for image retrieval. A tree-structured recursive learning mechanism based on the PCA and LDA features is developed in this framework to accelerate retrieval (Swets and Weng, 1999).

To avoid the singular problem which may occur when eigen-decomposing the scatter matrix in LDA, a two-phase strategy can be adopted: face images are first projected onto a face subspace via PCA to obtain PCA vectors whose dimension is adequately small to ensure the scatter matrix non-singular, then LDA is performed on such a set of PCA vectors (Swets and Weng, 1996). A similar approach to frontal-view face recognition using combined PCA and LDA have been reported by Zhao *et al.* (Zhao et al., 1998b; Zhao et al., 1998a).

Edwards *et al.* (Edwards et al., 1996) adopted LDA to select Discriminant Parameters based on Active Appearance Models. They claimed that these parameters can be used to effectively decouple identity variance from pose, lighting and expression variance.

2.3.3 Combined Shape and Texture Approach

Shape and texture are two important parts of information for a face image. Using facial shape alone is clearly insufficient to represent faces comprehensively. On the other hand, the un-registered facial texture may also bring considerable noise to the task of face recognition and facial analysis.

In fact, some degree of registration has been applied to most of the appearance based approaches introduced in Section 2.3 to obtain the normalised texture patterns. For example, the face images are aligned with the location of eyes, nose and mouth in (Moghaddam and Pentland, 1995; Moghaddam and Pentland, 1997). A more precise alignment of face images can be achieved when facial shape information is available. Then the isolated shape and texture information can be combined together for face modelling.

Vetter and Poggio (Vetter, 1996; Vetter and Poggio, 1997; Vetter, 1998) used pixelwise correspondence with optical flow to separate the 2D shape information captured in the correspondence field from the texture information obtained by mapping the pixels onto the reference face. The facial shape or texture patterns are then represented by a linear combination of prototype patterns respectively. New images of the face in different views can be synthesised from a single given image using this method.

However, establishing pixelwise correspondence is computationally intensive. Moreover, it may also be problematic to establish the pixelwise correspondence between different people due to the difference of facial appearance. To address these problems, a faster but less accurate method, establishing correspondence through a sparse set of features, has been adopted by many researchers.

Lades *et al.* (Lades et al., 1993) used a rectangular grid to sparsely sample the facial texture. Gabor wavelet jets, i.e. the filtered local texture with multiple resolutions and orientations, are computed on the grid points. Then recognition is performed by elastic graph matching. However, the shape used to establish correspondence in this work, i.e. the rectangular grid, is not specific to faces. In a later paper, Wiskott *et al.* (Wiskott et al., 1997) extended the work in three respects: phase information is used for accurate node positioning, object-adapted graphs are used to handle large rotations in depth, and image graph extraction is based on a novel data structure, the bunch graph, which is constructed from a small set of sample image graphs.

Lanitis *et al.* (Lanitis et al., 1994; Lanitis et al., 1995b; Lanitis et al., 1995c) developed a face identification system using both facial shape and texture information. The shape information is obtained through an ASM. Two kinds of texture patterns have been used and compared in this work: the local grey-level around landmarks and shape-free grey-level obtained by warping face images to the mean shape of the ASM. The authors demonstrated that the combined shape and texture information achieves the best recognition performance, while both of the texture patterns outperform the shape patterns alone.

Cootes *et al.* (Cootes et al., 1998; Edwards et al., 1996) introduced the Active Appearance Model which combines both the shape and grey-level variation within a single statistical model. This model begins with a Point Distribution Model and uses the landmarks of this model to build the shape-free texture model. By learning about the correlation between the shape and texture parameters using another PCA, a single unified model is obtained. This model also provides a fast, linear algorithm to fit the model on new face images. The relationship between model parameter displacements and the residual errors between a training image and a synthesised model example is learnt offline. This relationship is then employed to predict changes to the current parameters, leading to a better fit.

Both the ASM and AAM have been extended to nonlinear cases across views based on KPCA (Romdhani et al., 1999b; Romdhani et al., 2000a; Romdhani et al., 2000b). These nonlinear models aimed at corresponding dynamic appearances of both shape and texture with large pose variation.

2.4 3D Structure Based Approach

Faces can be modelled by a 3D mesh describing the geometric configuration and a texture map capturing the surface properties. Using 3D scanning devices is an explicit method to obtain the 3D features of faces. Alternatively, these features can be acquired from a set of 2D face images in different views.

Two decades ago, in his pioneering work, Parke (Parke, 1974; Parke, 1975) presented a 3D face model which is constructed of polygonal surfaces and manipulated through a set of parameters to control interpolation, translation, rotation and scale of facial features. The model has been successfully used to produce speech synchronised facial animated sequences with changes in rotation, scale, and facial expression.

DeCarlo and Metaxas (DeCarlo and Metaxas, 1996a; DeCarlo and Metaxas, 1996b; DeCarlo and Metaxas, 2000) presented a 3D deformable face model with a polygon mesh. The model is formed from ten component parts, each with its own set of deformation. The motion of the face model, such as expression change, is modelled using a separated set of deformations. Optical flow and edge information are employed as constraints for face tracking. This model uses a small number of parameters to describe a rich variety of face shapes and facial expressions.

Jebara and Pentland (Jebara and Pentland, 1997) proposed an approach to recover the 3D face structure using Structure from Motion. The estimation of the 3D structure is further constrained for reliable feature tracking by a 3D generic face model which is formed offline from a database of range face data.

Vetter and Blanz (Vetter and Blanz, 1998) introduced a flexible 3D face model learnt from examples of individual 3D face data. A novel 2D face image can be matched to the 3D model in an analysis-by-synthesis way, then images of the novel face in different views, illumination, and expression can be generated by changing the parameters of the matched model.

3D face models have also been used for person-independent face tracking and feature detection (Li et al., 1993; Shakunaga et al., 1998).

Besides the 3D face modes constructed from 3D range data or by handcraft, some researchers have attempted to model the 3D structure of faces through a set of 2D face images in different views.

Akimoto *et al.* (Akimoto et al., 1993) created a generic 3D head model using features extracted from regions and edges of eyes, nose, mouth, hair, and outlines of face. This method has been used for narrow-band visual communication.

Choi *et al.* (Choi et al., 1991) introduced a 3D facial shape model. Based on this model, a particular face image is represented by a weighted sum of facial image bases.

Similarly, Vetter and Poggio (Poggio and Vetter, 1992; Vetter and Poggio, 1997; Vetter, 1998) constructed a single generic 3D model of human head, which is used to solve the correspondence problem between 2D face images in different poses. They claimed that it is possible to synthesise images of a given face in novel views with only a single 2D image of the face.

By using 3D face models, one can track and analyse faces with large pose change. Meanwhile, a fitted face by a 3D model can be represented at any view, which is interesting not only for face recognition and facial analysis, but also for facial animation and visual interaction. However, the shortcomings of 3D face models come from modelling and computational complexity.

2.5 Video Based Face Recognition

Apart from the research on static images, video based face recognition has attracted great interest recently. This is mainly propelled by the demands from applications such as video conferencing, advanced human-computer interface, visual interaction, visual surveillance and access control. In addition, it is also involved in the studies of perception and behaviour which are closely related to face recognition, e.g., gesture recognition, human behaviour interpretation, and Computer Supported Cooperative Work.

Recognising faces from video input presents new difficulties which largely do not exist in the case of dealing with still images. For example, the quality of video images is usually poor, the resolution is low, and the subjects are not necessarily cooperative. Despite these difficulties, video based face recognition has superior advantages over that from static images:

1. More information about faces across multiple views and over time is available to facilitate the task of recognition for an accurate and reliable performance even when the quality and resolution of the video are poor.

2. The information collected over time can be re-used when part of the current data is missing or unavailable due, for example, to occlusion, self-occlusion, and image noise.

3. A more precise representation can be achieved from motion, temporal continuity and identity constancy. For example, the 3D structure of a face can be recovered by Structure from Motion (Hildreth, 1984; Ullman, 1979), or Structure from Shading (Horn and Brooks, 1989; Atick et al., 1996).

4. Apart from the shape and appearance of faces, the person-specific *dynamic* characteristics, for example, a distinct head gesture of an individual, can be captured from image sequences to facilitate the process of identification.

Gong *et al.* (Gong et al., 1994) have addressed the issue of encoding and recognising moving faces. The regions of interest are detected using motion information, then an elliptic active contour is initialised at the bounding box of each region of interest. Kalman filters are adopted to track the elliptic contour through an image sequence over time. The tracked face forms a temporal signature in a multi-view eigen space. Then a partially recurrent neural network is trained from these temporal signatures of different face classes for face recognition. It is reported that the system achieves an over 90% success rate with 40 different test sequences of known and unknown individuals.

Howell and Buxton (Howell and Buxton, 1996) reported a preliminary system for face recognition from image sequences. Radial Basis Function networks are used to tackle the unconstrained face recognition problem from low resolution video input. Two sets of sequences, the primary sequences - a controlled set of data including 180° pose change but with a blank background, and the secondary sequences - totally unrelated data, were tested in the system. Difference of Gaussian and Gabor wavelet filtering were used for image pre-processing. The system was first tested on the primary sequences from a small group of subjects (8 subjects) where sampled frames were used for training examples and the others for test examples. It showed that a high standard of performance is maintained when a confidence measure is used to discard uncertain frames although the initial recognition rate decreases as the sampling interval increases. The authors claimed that the results from the secondary sequences also show considerable promise, especially with the additional use of temporal coherence to improve performance.

McKenna and Gong (McKenna and Gong, 1998c; McKenna and Gong, 1998a) described an integrated system for recognising moving faces in poorly constrained dynamic scenes. Modules for focus of attention, face detection, tracking and recognition are included in this system. Two visual cues based on motion and skin colour are computed for focus of attention. A neural network based face detector is trained to determine the location and scale of faces in probable regions. The face tracking process can provide estimates of face position, scale and pose to improve efficiency and robustness. From image sequences, a large number of face images of a person were obtained, and the probability density function of the subject was estimated using Gaussian mixture models (McKenna et al., 1997). With appropriate order, these mixture models of facial identities can achieve greater accuracy than the typical nearest mean classification.

Similarly, Yamaguchi *et al.* (Yamaguchi et al., 1998) presented an approach to face recognition in temporal image sequences. A subspace is built from the detected face patterns in an image sequence, then face recognition is performed by matching this subspace with prototype subspaces.

Steffens *et al.* (Steffens et al., 1998) presented a real-time face recognition system which is able to capture, track and recognise a person walking toward or passing a pair of stereo cameras. Motion and stereo disparities are used to find regions of interest that are likely to correspond to heads. Elastic graph matching is then adopted to determine the scale of a face and a set of landmarks on a focused region. The final face identification is also performed using elastic graph matching to find the graph in the album gallery with highest similarity. One limitation of the system is that it is restricted to frontal-view faces. However, the authors claimed that preliminary results have shown promise to multi-view faces when combining with the across-view feature transformation scheme reported in (Maurer and Malsburg, 1995) and the pose estimation method reported in (Kruger et al., 1996).

Choudhury *et al.* (Choudhury et al., 1999) proposed a person identification system to recognise and verify people from unconstrained video and audio. The system can detect and compensate for pose variation and changes in the auditory background and can also select the most reliable video frame and audio clip for recognition. Mixture of Gaussians is adopted for skin colour detection, then facial features such as eyes, nose and mouth are located using symmetry transforms and image intensity gradient. The location of these facial features gives an estimate of head pose. Then the detected face is registered to a generic 3D face model and warped to frontal-view using the pose information. A single eigen space is constructed from these warped face patterns, and recognition is performed using the eigenface method. In this system, the 3D depth information of a human head is also used to detect the presence of an actual person as opposed to an image of that person. It is reported that the system achieves 100% recognition and verification rates on natural real-time input with 26 registered clients.

Edwards *et al.* (Edwards et al., 1998c; Edwards et al., 1999) proposed an approach to learning the class-specific correction of identity parameters from image sequences. The Active Appearance Model is adopted for locating faces from images. The fitted model parameters may contain variation from expression, illumination and pose changes, thus LDA is applied to these parameters to extract the most discriminating features. The authors claimed that the features obtained from LDA are optimal for all face classes, but not necessarily optimal for a specific individual. To extract the class-specific projection of the features, an online learning scheme is developed. A zeroth order Kalman filter is also designed to obtain an optimal estimation over a whole image sequence temporally. Experimental results showed that more stable identity estimates have been achieved with this correction mechanism.

In general, most of the existing studies on video-based face recognition have the following characteristics:

1. Motion and skin colour have been used as efficient tools for selective attention to find regions of interest;
2. Most of the systems integrate the processes of selective attention, face detection, tracking and recognition;
3. The techniques used for recognition are still largely based on those used for static images.

Nevertheless, it is important to note that some researchers have been exploiting the dynamic characteristics of continuously moving faces in image sequences, for example, the temporal facial signatures (Gong et al., 1994) and online class-specific discriminating feature correction (Edwards et al., 1998c).

2.6 Limitations of Previous Studies

Although the issue of face recognition has been extensively addressed for three decades, there still exist many limitations. In this thesis, we mainly aim to address three challenging problems: modelling faces with large pose variation, using nonlinear techniques for an improved representation, and modelling faces in a dynamic context.

2.6.1 Modelling Faces with Large Pose Variation

The problem of frontal-view face recognition has been extensively addressed by many researchers. In most of the previous studies, a single model can usually provide a satisfactory solution to the problem since the patterns of frontal-view faces share a similar shape and appearance configuration. However, for faces with large pose variation, i.e. multi-view faces, the techniques which have been successful in frontal-view may fail. One of the most significant difficulties comes from the fact that *the similarity between two images of the same person in different poses is less than the similarity between images of two different persons in the same pose.*

Figure 2.1. Face images from two persons with large pose variation. the similarity between two images of the same person in different poses is less than the similarity between images of two different persons in the same pose.

This problem is illustrated in Figure 2.1 where the faces from two subjects rotate from left profile to right profile. It is interesting to note that this does not challenge our biological vision system since we can distinguish the two set of faces from each other with great ease. However, from the viewpoint of image processing and machine based vision system, the simple and effective techniques used in the frontal-view counterpart may not work. For example, we applied the eigenface technique to these face patterns. The distribution of the patterns in the first two dimensions is shown in Figure 2.2. It is clear that the distance between patterns of a same face class is not necessarily smaller than that between patterns of different face classes. Therefore the widely used nearest mean or nearest neighbour method would give poor recognition.

2.6.2 Non-linear Techniques

The multi-view problem is related to the second problem we will discuss in this thesis: non-linear techniques are needed to represent and model multi-view face

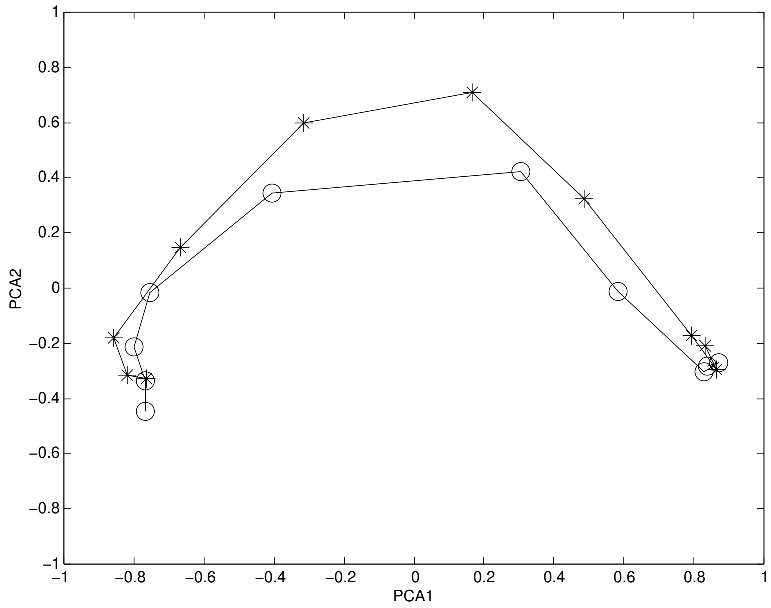


Figure 2.2. Representing the face images in Figure 2.1 in the first two eigenface dimensions. The distance between patterns of one person from different views is not necessarily smaller than that between patterns of different face classes.

patterns due to the severe non-linearity caused from rotation in depth, self-shading and self-occlusion.

The techniques used most often in the previous studies are linear techniques. For example, an optimal linear transform based on least mean squared reconstruction error is computed in the eigenface method. Similar examples include Linear Discriminant Analysis and the Active Shape Models. When applied to the frontal-view or near frontal-view problems, these linear techniques can provide fast and satisfactory results. However, for the multi-view problem as shown in Figure 2.2, where the patterns from each face class exhibit a significant non-linear distribution, these linear techniques are ill-suited.

2.6.3 Modelling Faces in a Dynamic Context

As stated in Section 2.5, perceiving moving faces is more than static image matching. The motion of faces over time, including the rigid motion of the overall head and the local non-rigid motion of the facial feature, encodes dynamic information of both person-specific characteristics and generic human behaviour characteristics. The former is related to identification of an individual, while the latter is important for human behaviour recognition.

Unfortunately, although considerable effort has been made in the research of video-based face recognition and facial analysis, the methodology adopted in the previous studies is largely based on static image matching. However, some researchers have been working on the issue of dynamic face recognition. For example, Gong *et al.* (Gong et al., 1994) presented the concept of temporal signatures of facial identities and Edwards *et al.* (Edwards et al., 1998c) developed a method for online correction of identity parameters in image sequences.

In this work, we present a different approach to the problem of dynamic face recognition using *identity surfaces*. The issues of modelling faces with large pose variation and using non-linear techniques for discriminating feature extraction are also addressed.

Chapter 3

Estimating Head Pose

For appearance based methods, dealing with face changing across multiple views [1] is one of the most challenging problems because of the severe nonlinearity caused by rotation in depth, self-occlusion, self-shading, and change of illumination.

A straight-forward method of solving this problem is to model the multi-view face patterns using multi-modal techniques such as mixture of Gaussians. Unfortunately these techniques usually require a large number of training examples which are difficult, if not impossible, to collect. However, if the pose information of a face is available, the problem can be solved more efficiently.

The basic ideas of explicitly using pose information for face modelling, detection, and recognition can be expressed as follows:

1. Pose information can be used to decompose the problem of multi-view face detection into several sub-problems of face detection in a fixed range of views, which are much easier. For example, if a face in an image is known in frontal view, one can use a specific face detector tuned to frontal view to perform the task.

2. Align face images across views. The performance of face recognition crucially depends on the efficiency of face image alignment. Given the pose

[1] Strictly speaking, pose is referred to as the *subject-initialised* position and angle, while view is defined *objectively* from the stand of an observer. However, we use the two terms indiscriminatingly in this context.

information and the 3D shape of a face, the appearance of the face can be normalised to be *shape-and-pose-free*, which considerably reduces the non-linear variation caused by pose change.

3. Given pose information, the computation in detecting faces across views can be simplified. Previous research indicated that a universal detector is usually not sufficient to deal with variations across views. Instead a set of detectors, each for a specific range of views, should normally be designed to carry out the task (Moghaddam and Pentland, 1997; Ng and Gong, 1999b). If no pose information is available, all the detectors have to be computed. Also, extra computation is needed to determine if the pattern is a face from the outputs of all the detectors. However, if the pose of faces can be estimated, only the detector at the estimated pose needs to be computed.

4. Pose information provides a useful cue for face tracking, which is very important in alleviating the burden of computation and understanding the dynamics of faces in a spatial-temporal context.

5. As well as gaze, gesture and gait, the head pose of a subject is also closely related to visual behaviour exhibited in visual interaction and human-computer interface.

In the rest of this chapter, we first briefly review the previous studies on pose estimation in Section 3.1. A system for multi-view face image acquisition is discussed in Section 3.2. Section 3.3 describes an approach to representing face patterns using PCA. An SVM regression based algorithm for pose estimation is presented in Section 3.4. Section 3.5 is a summary of this chapter.

3.1 Background

In early approaches to head pose estimation, contacted sensing was widely adopted. Although these methods are still used in many situations now, the issue of using non-contact and passive methods to estimate pose and gaze has been of great interest in recent years. In the rest of this section, we review these approaches in four categories: contacted-sensing based approach, facial feature based approach, image feature based approach, and appearance based approach.

3.1.1 Contacted Sensing Based Approach

In early research, pose and gaze estimation was usually conducted intrusively by using active and contacted sensing. For example, Hutchinson *et al.* (Hutchinson et al., 1989) demonstrated a human-computer interaction system in which the eyes of a user are illuminated with infrared light, and the gaze orientation of the user is calculated from the relative position of the *bright-eye* and the *glint* from the cornea.

3.1.2 Facial Feature Approach

Recently, many researchers have sought to adopt non-contact and passive methods for this problem. Geometrical approaches based on facial features such as eyes, nose and mouth have been widely used.

Gee and Cipolla (Gee and Cipolla, 1994) discussed the issue of head pose and gaze estimation from a single, monocular view of face. Two approaches, the planar approach and 3D approach, were presented. It was claimed that the former is more accurate for near-profile views of the face while the latter is superior for near-frontal views. A switch was also developed to automatically choose between the two alternative methods in their work.

Horprasert *et al.* (Horprasert et al., 1996) used five facial points, four at the eye corners and one at nose tip, to estimate the orientation of face from a single image. The yaw and roll components of the pose are computed using perspective projection, and the pitch component is estimated using an anthropometric model.

For these type of pose estimation methods, the accuracy and robustness of facial feature detection are crucial to the ultimate performance. To address the problem of feature detection and tracking, some researchers proposed to use stereo input.

Xu and Adatsuka (Xu and Adatsuka, 1998) proposed an approach to head pose estimation from stereo images. By tracking four facial points, pupils and mouth corners, in both input images, the 3D coordinates of these points can be estimated. The head pose is then computed as the normal of the plane covering the facial points.

Similarly, Matsumoto and Zelinsky (Matsumoto and Zelinsky, 2000) presented a stereo face tracking and gaze detection system to measure head pose and gaze direction. In their system, facial features such as eye corners and mouth corners are tracked from stereo video input, then pose estimation is performed by determining the rotation matrix from the previous facial feature positions using least squares method.

3.1.3 Image Feature Based Approach

The feature points chosen for modelling pose do not necessarily have to be physiologically meaningful. In other words, from the standpoint of image processing, these feature points should be selected as those with the most significant pictorial characteristics such as edge, valley and ridge. Moreover, these features are usually preprocessed by filtering before being tracked.

Maurer and Malsburg (Maurer and von der Malsburg, 1996) demonstrated a system capable of tracking face graphs and estimating head pose in natural image sequences. Gabor wavelet jets were taken as features for tracking. Based on the assumption that all the nodes of a facial graph are on a flat surface, the head pose is estimated by determining the optimal affine transformation.

Kruger *et al.* (Kruger et al., 1997) presented a system for the automatic determination of face position and head pose. Elastic Graph based on a set of wavelet jets is employed to represent face patterns. Pose estimation is performed by matching a number of bunch graphs that model different poses to the image and choosing the one with maximal similarity. This method has been extended by Elagin *et al.* (Elagin et al., 1998) using graph-flipping and similarity thresholding techniques. However, these approaches only provide a sparse set of discrete estimation, e.g. frontal, half profile left, profile left, half profile right, and profile right, rather than a continuous-valued estimation.

Wu and Toyama (Wu and Toyama, 2000) built an ellipsoidal head model encapsulating a set of regularly distributed points. Image filters such as rotation-invariant Gabor wavelets, Gaussians and Laplacians are applied on these points. In training, the density function of each model point on each pose angle is estimated with a single Gaussian. Run-time pose estimation is performed by maximising a *posteriori*. Although it is claimed that the system is considerably robust to real-world visual perturbations, the modelling and computation involved, which include filtering, probability density function estimation and posterior maximisation on each of the model points, are very expensive.

Shimizu *et al.* (Shimizu et al., 1998) used a method of Iterative Closest Curve matching to estimate head pose. A generic 3D head model with a set of curves to represent the contours of eyes, lips and eyebrows is employed in their approach. The pose of a face in a 2D image is recovered by iteratively minimising the distances between the projected model curves and their closest image edges. Although the curves contain more information than the isolated feature points, the performance of this approach heavily depends on the results of image edge detection.

3.1.4 Appearance Based Approach

For both facial feature based and image feature based approaches, the performance of pose estimation crucially depends on the successful tracking of features. In other words, these approaches try to solve a single face tracking and pose estimation problem using several feature tracking problems. Unfortunately, these feature tracking problems are not necessarily less trivial than the task of face tracking. As opposed to the feature based approach, some researchers tried to solve the problem using holistic facial appearance matching.

Beymer *et al.* (Beymer et al., 1993) proposed an approach to example based face analysis and synthesis in which a Radial Basis Function network is trained to estimate face rotation and the degree of smiling. However, intensive computation is required as pixel-wise correspondence between an image and its reference image needs to be established through computing optical flow.

Rowley *et al.* (Rowley et al., 1998b) presented a neural network based system which is capable of detecting faces with rotation in the image plane. In this system, a "router" network is designed to estimate the orientation of a search window in an image, then one or more "detector" networks are employed to determine if it is a face based on the orientation estimation. However, this approach was only proposed to deal with rotation in the image plane, which involves mainly linear geometrical transformation, which is less challenging than the rotation *out of* the image plane.

Gong *et al.* (Gong et al., 1996) investigated the distribution of multi-view face patterns in a pose eigenspace. It was demonstrated that the pose change of a continuous face rotation in depth forms a smooth curve in the pose eigenspace. In particular, by representing the face patterns with a composite Gabor wavelet transform and PCA, a highly linear pose distribution is achieved. This kind of representation was then employed for person-specific pose estimation based on nearest template matching (McKenna and Gong, 1998b) and person-independent pose recognition using similarities to prototypes (Gong et al., 1998b; Sherrah et al., 2001).

3.2 Multi-View Face Image Acquisition

We refer to the pose of a face as including both the yaw and tilt angles from its frontal-upright position. Rotation in the image plane is not taken into account, assuming that most faces appear upright in the images. This assumption is acceptable in most real-world scenarios in visual surveillance, visual conferencing and visual interaction.

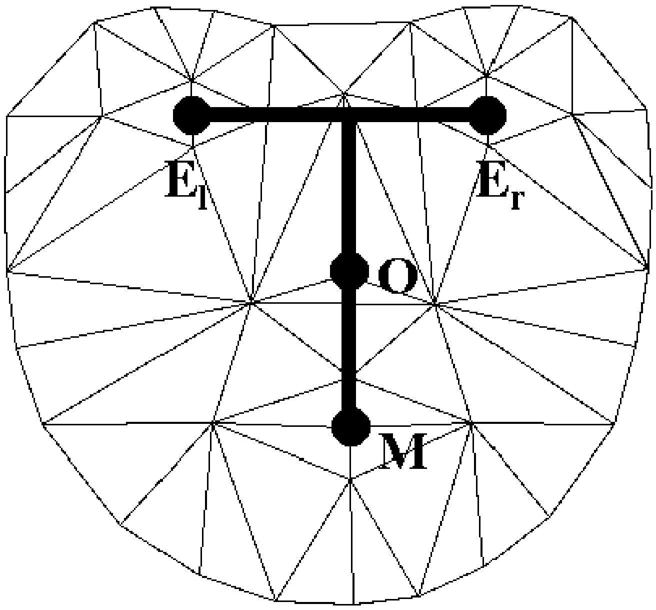


Figure 3.1. Rotation centre (O) used for measuring pose angles.

3.2.1 Definition of Pose

As shown in Figure 3.1, we assume the rotation centre O for measuring pose is the middle point between the eye centre and the mouth centre, i.e.

$$O = \frac{1}{2}(M + \frac{1}{2}(E_l + E_r)) \tag{3.1}$$

where M, E_l and E_r are the 3D positions of mouth centre, left eye and right eye respectively.

The rotation centre O is set as the origin of the object coordinate system of a face. The z axis is assumed to coincide with the vector from O to the observer, while the x and y axes point to the horizontal and vertical directions respectively.

We define the pose vector $\mathbf{n}$ of a face as the fixed vector on the face surface starting from O. It points towards the observer when the face is in frontal-upright position. The tilt α is defined as the angle between $\mathbf{n}$ and the $x - z$ plane, and the yaw β the angle between the z-axis and $\mathbf{n}'$, the projected vector of $\mathbf{n}$ in the $x - z$ plane.

It is important to note that:

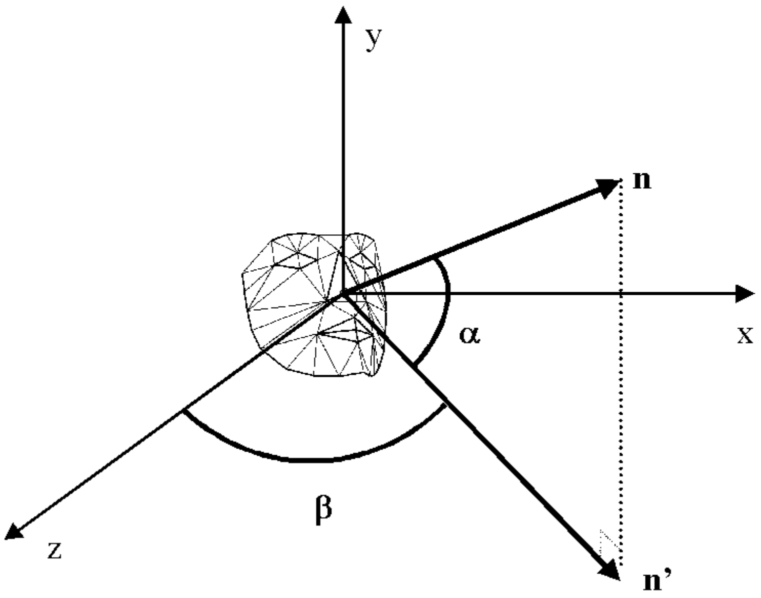


Figure 3.2. The object coordinate system and the definitions of tilt and yaw. The origin O is assumed to be the rotation centre of a face. The tilt α is defined as the angle between $\mathbf{n}$ *and the $x - z$ plane, and the yaw β the angle between the z-axis and* $\mathbf{n}'$*, the projected vector of* $\mathbf{n}$ *in the $x - z$ plane.*

1. The position of O is relatively stationary with respect to the local movement of facial parts, especially when a face undergoes significant expression changes. Moreover, unlike some person-dependent feature points, such as the nose tip which is related to the size of the nose, the position of O is less sensitive to variation of facial layout from different people. Therefore, it is more appropriate for registering faces across different views and from different people.

2. The rotation centre O is selected differently from that in the multi-view face acquisition system (Gong et al., 2000) where it is chosen as the centre of the head. This is because the 3D transformation and rotation can be computed more efficiently with the rotation centre defined in this work. The different settings of the rotation centre would introduce error to pose estimation. However, when the distance between the head and the camera is adequately

α	tilt
β	yaw
(x_{le}, y_{le})	position of the left eye
(x_{re}, y_{re})	position of the right eye
(x_m, y_m)	position of mouth

Table 3.1. Parameters obtained from the acquisition system

larger than the size of the head, the error can be ignored.

3.2.2 Data Acquisition and Alignment

In our previous work, a system was designed to capture the multi-view face images with labelled pose and positions of eyes and mouth. The system utilises a magnetic sensor rigidly attached to a subject's head and a camera calibrated to the sensor's transmitter. The sensor provides the 3D coordinates and orientation relative to its transmitter. In the initialisation stage, the positions of mouth and eyes are manually located on the screen. These positions are usually adjusted at different views to make sure they are rigidly "attached" to the facial features. More details about the multi-view face acquisition system is described in (Gong et al., 2000). The system provides the parameters listed in Table 3.1 as well as the multi-view face images.

Figure 3.3 shows a sample image with the pose and feature positions measured by the system. A squared subimage is cropped from the original image about the face centre $o(x_o, y_o)$ and with size $r \times r$.

$$x_o = 0.5(x_m + 0.5(x_{le} + x_{re})) \tag{3.2}$$

$$y_o = 0.5(y_m + 0.5(y_{le} + y_{re})) \tag{3.3}$$

$$r = 2.4Max(|x_{re} - x_{le}|, |y_m - 0.5(y_{le} + y_{re})|) \tag{3.4}$$

It is important to point out that

1. o is actually the projection of 3D rotation centre O defined in (3.1) in the image plane;

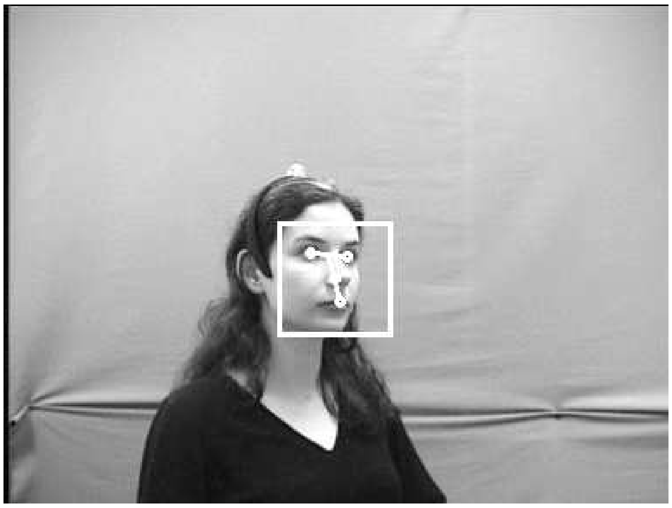


Figure 3.3. A sample image obtained from the multi-view face acquisition system. The locations of features including eyes and mouth are marked in the image. The sub-image centred at $o(x_o, y_o)$ and with size $r \times r$ is cropped to train the pose estimators and face detectors.

2. the cropped images provide a rough correspondence between the captured faces in terms of pose, position, and size.

The size of the original images are 384×288 pixels where the faces take about $50 \sim 60$ pixels. However, we try to develop an approach to face modelling and analysis in a low resolution image or video input, thus the cropped face images are scaled to 20×20 pixels. The range of pose of these face images is $[-90^\circ, +90^\circ]$ in yaw and $[-30^\circ, +30^\circ]$ in tilt. We have collected a set of multi-view face images from 31 subjects. Figure 3.4 shows sample face images from one of the subjects.

3.3 Representation of Face Patterns

The multi-view face images are preprocessed using background subtraction and intensity normalisation. Principal Component Analysis (PCA) is then employed for a low-dimensional representation.

Figure 3.4. Sample face images of one subject from the multi-view face database. Those images are taken with yaw angle changing in $[-90°, +90°]$ *and tilt angle in* $[-30°, +30°]$.

3.3.1 Preprocessing

To reduce the variation in illumination, skin tones and background, face images are preprocessed before PCA.

An inconsistent background may impose a bad influence on pose estimation, face detection and recognition. To alleviate this influence, background is subtracted from the face images.

For images with homogeneous background such as those in the database, a simple Gaussian model can be constructed from the background colour pixels.

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{3/2}|\mathbf{\Sigma}|^{1/2}} exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right) \tag{3.5}$$

where $\mathbf{x}$ is the colour vector in RGB, $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$ are the mean vector and covariance matrix which can be estimated by Maximum Likelihood method (Bishop, 1995).

If the value of Equation (3.5) for a pixel is above a preset threshold p_0, then the pixel is regarded as background colour. Otherwise, it belongs to the foreground. The threshold p_0 is chosen as the value of minimal misclassification against a set of foreground pixels as discussed in (Bishop, 1995).

When the background cannot be modelled as a homogeneous one, more complicated techniques are needed. However, this issue is beyond the topic of this thesis. More details can be found in (Toyama et al., 1999; Stauffer and Grimson, 1999; Comaniciu and Meer, 1999).

Meanwhile, the illumination change and skin tones may also cause significant variation to the appearance of face images. To reduce the variation stated above, we simply normalise the intensity $\mathcal{I}_0$ of a face image by its norm

$$\mathcal{I} = \frac{\mathcal{I}_0}{\|\mathcal{I}_0\|} \tag{3.6}$$

3.3.2 PCA for Feature Extraction and Dimension Reduction

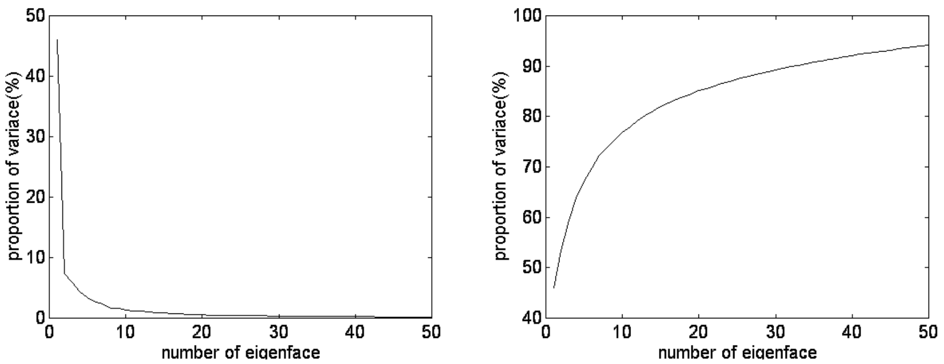
PCA seeks to minimise the mean square error by constructing a low-dimensional orthogonal space from the first few eigenvectors of an eigen-composition. This technique has been widely applied in many pattern recognition problems, especially in face detection and recognition where it is well-known as the "eigenface" method. Appendix A gives more details for performing PCA.

We trained PCA on 2660 multi-view face images from 20 subjects with pose changes between $[-90°, +90°]$ in yaw and $[-30°, +30°]$ in tilt. Figure 3.5 shows the proportions of variance in percentage of the first n eigenfaces. The first 20 eigenfaces are shown in Figure 3.6.

Sample images from a new subject which are not used for training the PCA and their reconstruction from the first 10, 20, 30, 40, and 50 eigenfaces are illustrated in Figure 3.7. One notices that the general characteristics of multi-view faces can be expressed by a relative small number of eigenfaces, for example, the first 20 eigenfaces. It is indicated that the first few eigenfaces capture the person-independent characteristics of faces which is important for face detection and pose estimation, while the rest of the eigenfaces may carry the person-dependent characteristics which are useful for face recognition.

After training the PCA, a preprocessed face image can be projected into the feature space spanned by the first M eigenfaces to obtain a feature vector

$$\mathbf{x} = \mathbf{U}^{\mathrm{T}}(\mathcal{I} - \mu) \tag{3.7}$$



(a) proportions of variance of eigenfaces

(b) accumulated proportions of variance of eigenfaces

Figure 3.5. Distribution of variance of PCA on multi-view face images with respect to the number of eigenfaces

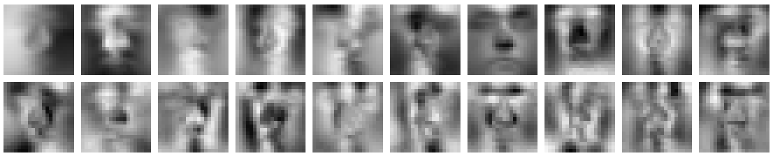
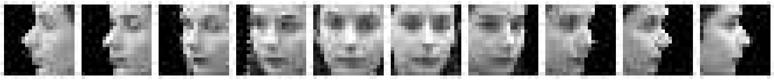


Figure 3.6. The first 20 eigen faces trained from 2660 multi-view face images of 20 subjects.

where $\mathcal{I}$ is the preprocessed image, $\boldsymbol{\mu}$ is the mean vector of all training images, and $\mathbf{U}$ is a matrix comprising the first M eigenfaces. $\mathbf{x}$ is the input to the pose estimator and multi-view face detectors.

3.4 Estimating Head Pose Using SVM Regression

We adopted an appearance based method to estimate the head pose. The input to the pose estimators is the PCA vectors of face images, and the output is the



(a) Face images of a new subject which are not used to train PCA

(b) Reconstructed face images from the first 10, 20, 30, 40 and 50 eigenfaces

Figure 3.7. The PCA representation for multi-view face detection.

pose angles in tilt and yaw.

3.4.1 Algorithm

In this research, SVM regression is employed for pose estimation. Compared with other learning methods, the SVM-based method has distinguishing properties such as:

1. No model structure design is needed. The final decision function can be expressed by a set of "important examples" called Support Vectors (SVs).

2. By introducing a kernel function, the decision function is implicitly defined by a linear combination of training examples in a high-dimensional feature space.

3. The problem can be solved as a Quadratic Programming problem, which is guaranteed to converge to the global optimum of the given training set.

After transformed to its dual problem, the SVM regression problem can be solved by maximising

$$
\begin{aligned}
W(\alpha^*, \alpha) &= -\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)k(\mathbf{x}_i, \mathbf{x}_j) \\
&\quad -\varepsilon \sum_{i=1}^{l}(\alpha_i^* + \alpha_i) + \sum_{i=1}^{l} y_i(\alpha_i^* - \alpha_i) && (3.8) \\
\text{subject to} \quad & \sum_{i=1}^{l}(\alpha_i^* - \alpha_i) = 0 && (3.9) \\
& 0 \leq \alpha_i^*, \alpha_i \leq C && (3.10)
\end{aligned}
$$

which provides the solution

$$
f(\mathbf{x}) = \sum_{i=1}^{l}(\alpha_i^* - \alpha_i)k(\mathbf{x}, \mathbf{x}_i) + b \tag{3.11}
$$

where $\mathbf{x}$ is the PCA feature vector of a face image from Equation (3.7), k is the kernel function used in the SVM pose estimator, y_i is the pose angle in yaw or tilt of pattern $\mathbf{x}$. More details about SVMs are presented in Appendix D.

Two SVM regression based pose estimators, one for yaw and the other for tilt, are constructed. The Quadratic Programming problem is solved by a decomposition algorithm based on the LOQO algorithm (Vanderbei, 1994) [2].

3.4.2 Tuning the Parameters

Two kinds of parameters need to be considered in pose estimation: parameters of the SVM and the dimension of PCA.

SVM Parameters

In order to construct an SVM based pose estimator, one needs to choose an appropriate kernel function and set the parameters of the kernel function. We have tested the performance of the SVM based pose estimator on different types of kernels and different parameter settings. The experimental results indicate: [3]

[2]The code of the LOQO algorithm is kindly provided by Alex Smolar.

[3]It is noted that similar results about kernel selection and parameter setting have been achieved in other experiments including SVM based frontal-view face detection (Section 4.2.3), multi-view face detection (Section 4.4.2), KDA training (Section 6.3.2) and 2D classification/regression (not reported in this thesis).

1. Different kernel functions, such as Gaussian, polynomial and sigmoid kernels, provide similar performance, nevertheless, the best results are usually achieved when a Gaussian kernel is adopted;

2. The performance is not very sensitive to small changes of the SVM parameters. This indicates that it is not necessary to design a very complicated algorithm to adjust the parameters. In our experiments, it is found that the Gaussian kernel usually provides an acceptable performance when its parameter is set as $2\sigma^2 = 1$ and the patterns are normalised to unit vectors. Although a sound mathematical proof is lacking, this setting proves to be very useful and effective in practise.

In this experiment, a Gaussian kernel

$$k(\mathbf{x}_1, \mathbf{x}_2) = exp(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{2\sigma^2}) \tag{3.12}$$

is used to build the SVM. The parameters selected in the algorithm are listed in Table 3.2.

Kernel	Gaussian
$2\sigma^2$	1
C	1000
Image dimension	400 (20×20)
Range of tilt	$[-30° , +30°]$
Range of yaw	$[-90° , +90°]$
Number of subjects used for training	10
Number of subjects used for test	10
Images of each subject	133
Total number of training images	1330
Total number of test images	1330

Table 3.2. Parameters of the SVM based algorithm for pose estimation

Tolerance Coefficient ε

The tolerance coefficient ε is used to define the ε-insensitive loss function (Vapnik, 1995) in SVM regression.

$$|f(\mathbf{x}) - y| = \begin{cases} 0, & \text{if } |f(\mathbf{x}) - y| \leq \varepsilon \\ |f(\mathbf{x}) - y|, & \text{otherwise} \end{cases} \tag{3.13}$$

where f is the regressed function, and y is the known label of pattern $\mathbf{x}$. By introducing the loss function defined in (3.13), the SVM can provide a sparse solution to a regression problem, i.e. the number of SVs can be far less than the number of the training examples.

Normally, ε can be used to control the accuracy of a SVM regressor. A large value of ε may lead to a regression function with poor accuracy and good real-time performance since a larger error is acceptable by the loss function (3.13) and a smaller number of SVs can be obtained from training. However, it is important to point out that one cannot expect to achieve a perfect result by setting ε to 0 or near 0. The maximal accuracy of a regression problem is determined by its VC-dimension (Vapnik, 1995). Too small a value of ε may lead to overfitting, i.e. the results are perfect on the training set but deteriorate on the test set. Scholkopf *et al.* (Scholkopf et al., 1998a; Smola et al., 1999; Scholkopf et al., 2000)have discussed this problem extensively. They also presented a method for automatic accuracy control in SVM regression.

Pose information is needed for face detection, tracking and recognition. However, the requirement for pose estimation can be different for different visual tasks.

1. For face detection (more details will be discussed in Chapter 4), pose information is used to choose an appropriate detector, or to quickly discard the extraordinary patterns, for example, the ones with pose significantly different from the detected face in the previous frame of a sequence. Pose estimation in this case does not need to be very accurate. On the other hand, as face detection is based on exhaustively scanning the images, the burden of computation is quite heavy. Thus a relatively simpler pose estimator which is less accurate but faster is appropriate in this case.

2. When fitting the dynamic model to a face image, which will be addressed in Chapter 5, and recognising a face using *identity surfaces*, which will be described in Chapter 7, a more precise estimation of pose is needed.

To investigate the influence of ε on the performance of a pose estimator, we designed the following experiments where the value of ε changes from 2 to 20. The PCA dimension is fixed to 20, and other parameters are chosen as listed in Table 3.2. Figure 3.8 shows the results of SV numbers, errors in tilt and yaw, and test time.

The experimental results indicate:

1. The number of SVs increases steeply with the decrease of ε. The number when $\varepsilon = 2$ is over 8 times as big as that when $\varepsilon = 20$.

2. Lowering the value of ε does not always improve the accuracies. Actually, the optimal accuracies are obtained when ε is chosen around 10, which may reflect the intrinsic precision of the training examples.

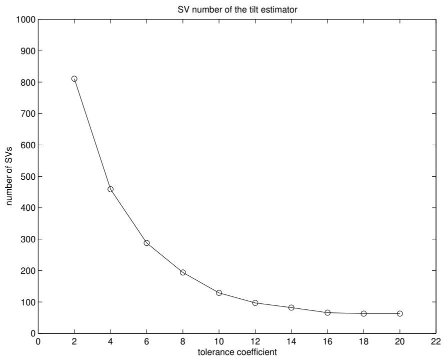
3. Better real-time performance is achieved when increasing ε therefore less SVs obtained. This is because the estimation speed is determined by the number of SVs.

PCA Dimension

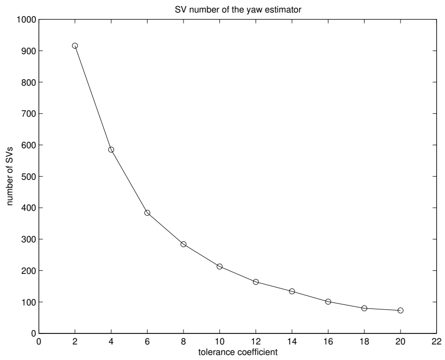
We designed the following experiment to evaluate the performance of the SVM based pose estimators with different PCA dimensions. ε is fixed to 10, and other parameters are chosen as listed in Table 3.2 in this experiment. The performance is evaluated in terms of the number of SVs, the estimation error on the test set, and test time. The results are shown in Figure 3.9.

From the experimental results, we have the following observations:

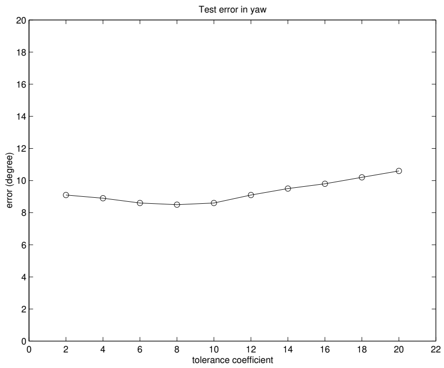
1. Except for the very low dimensional case, the number of SVs remains constant with the increase of the PCA dimension. This reflects the underlying characteristics of SVMs since the number of SVs is corresponding to the



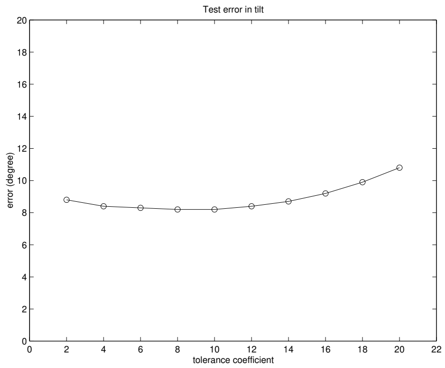
(a) SV number for the yaw estimator



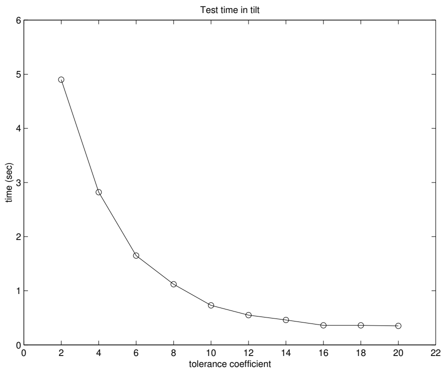
(b) SV number for the tilt estimator



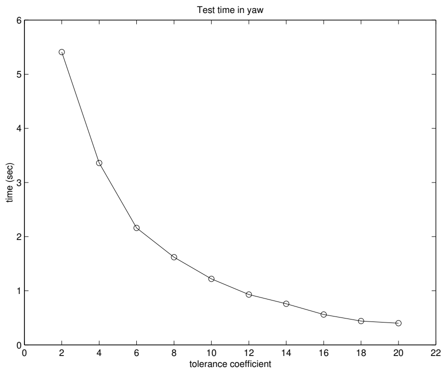
(c) test error in yaw estimation



(d) test error in tilt estimation



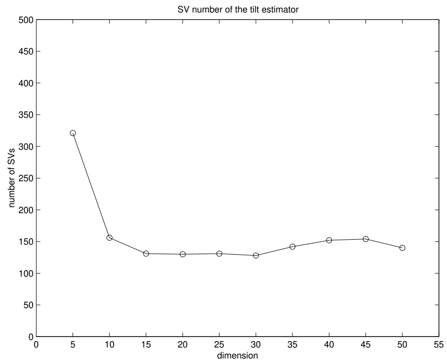
(e) test time in tilt estimation



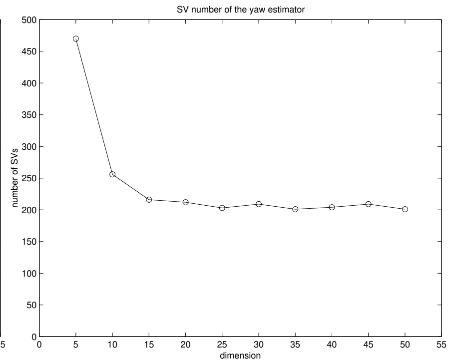
(f) test time in yaw estimation

Figure 3.8. Pose estimation performance vs. tolerance coefficient ε

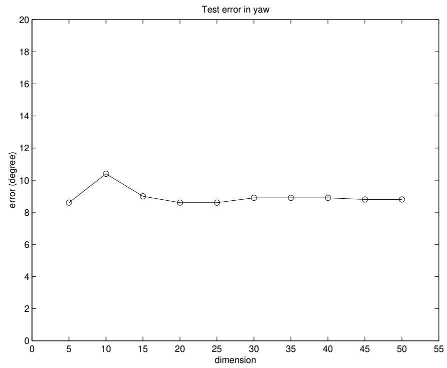
VC-dimension of the problem. A very high PCA dimension does not provide further improvement to the performance. When the PCA dimension is



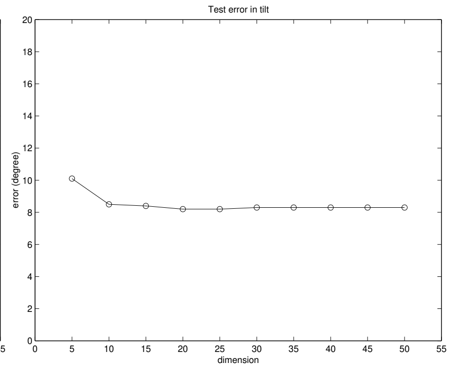
(a) SV number of the yaw estimator



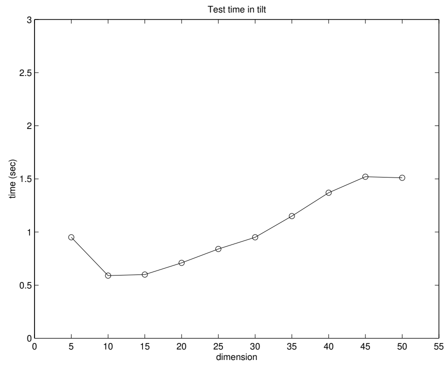
(b) SV number of the tilt estimator



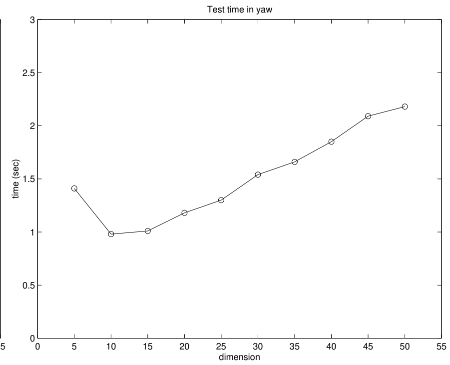
(c) test error in yaw



(d) test error in tilt



(e) test time in tilt



(f) test time in yaw

Figure 3.9. Pose estimation performance vs. PCA dimension

below 15, the number of SVs is considerably high since the representation with very low dimension is incapable to capture sufficient information for pose estimation.

2. Estimation errors are approximately stable except for the case of very low dimensions. This indicates that a relatively low PCA dimension can provide sufficient accuracy. The stable error rates also indicate that the SVM based pose estimators correctly reflect the intrinsic precision of the training examples and they are not over-fitted even when the PCA dimension is high.

3. The estimation speed is related to the number of SVs and the PCA dimension. When the PCA dimension is below 15, a poor real-time performance is observed due to the large number of SVs. Above that, the test time increases nearly linearly with the increase of dimension.

These experiments result indicate that a relatively low dimensional representation, for example, 20 in PCA dimension, can provide satisfactory performance in terms of accuracy and run-time speed in pose estimation.

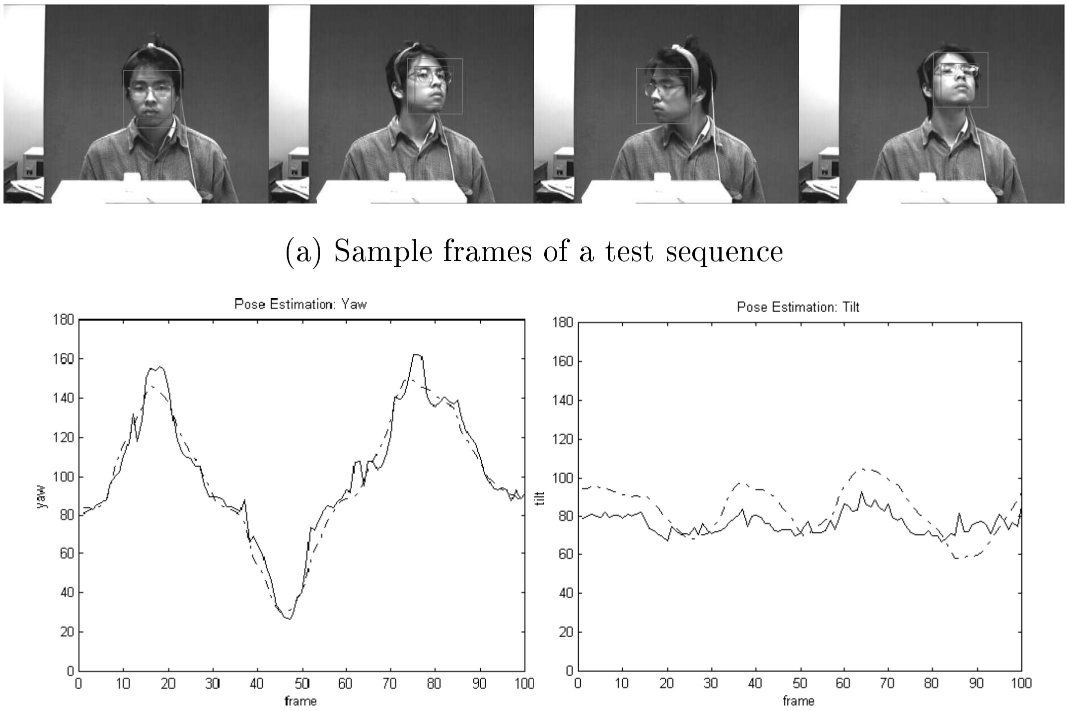
3.4.3 Test Results

Figure 3.10 shows estimated pose from a test sequence. The parameters used for SVM training are listed in Table 3.2. The dimension of the PCA vector of face patterns is chosen as 20. Over the whole sequence, the estimation errors in both yaw and tilt are around 10° , which are sufficiently accurate for the purpose of multi-view face detection and recognition in this work.

3.5 Summary

Modelling the appearance of faces with large pose variation may be problematic due to the severe nonlinearity caused by rotation in depth, self-occlusion and self-shading. However, if pose information is explicitly used, the problem can be simplified to some extent.

We have presented an appearance based approach to pose estimation in this chapter. PCA is adopted to represent multi-view face patterns in a low-dimensional



(a) Sample frames of a test sequence

(b) Yaw estimation

(c) Tilt estimation

Figure 3.10. Pose estimation on a test sequence. In (b) and (c), the solid curves are the estimated pose in yaw and tilt and the dotted curves are the ground-truth pose which is measured by the data acquisition system.

orthogonal feature space, and SVM regression is employed to construct the pose estimators.

It is important to point out that we adopted very low-resolution images as input, e.g. all the face images used in the experiments are in the size of 20×20. The reasons are twofold:

1. In many real-world scenarios such as visual surveillance and video conferencing, only low-resolution images and video are available;

2. An improved real-time performance can be achieved on low-resolution input.

The dimension of raw images with the size of 20×20 is still high [4]. Meanwhile,

[4]In our experiments, it is hard to obtain satisfactory results of pose estimation and face

the pixel-ordered vector built directly from an image is not very efficient as a representation since it contains a lot of redundancy and does not provide sufficient information for similarity discrimination on its own. For example, shifting an image by a couple of pixels may only cause a slight difference in visual effect; however, this shift will probably bring a significant variation on the Euclidean distance measured in the vector space. To achieve a low-dimensional and more effective representation, we applied PCA to the face images.

On the other hand, the SVM has been of considerable interest in the area of pattern recognition and machine learning recently. Theoretically, it is based on Structural Risk Minimisation which deals with not only the empirical error in the training data, but also the capacity of a learning machine. Moreover, it can be trained directly from the data with little requirement of the prior knowledge about the data and it is guaranteed to converge.

However, it is important to point out that our approach to pose estimation presented in this chapter is made under the assumption that multi-view faces have been located in or cropped from the original images. In fact, face detection and pose estimation are inter-dependent which cannot be simply separated. How can we estimate the pose of a face without knowing the position of the face, and, how can we detect the face without knowing its pose? We leave these question to the next chapter where the problem of multi-view face detection and the relationship between face detection and pose estimation will be addressed in detail.

detection when a size of faces is below 20×20. Baker and Kanade (Baker and Kanade, 2000) have reported a similar case with image size of 12×16.

Chapter 4

Detecting Faces across Multi-Views

For most artificial systems, face detection is usually regarded as a necessary process before recognition. However, in biological vision systems, the relationship between face detection and recognition may not be so clear, especially under the circumstances where faces are moving continuously. It has been reported that the ability of human's vision system to recognise familiar faces is much better than that on unfamiliar faces (Knight and Johnston, 1997; Bruce et al., 1998b). We can quickly recognise a familiar face, sometimes subconsciously, from a group of people. In this case, it would be more plausible that we "see" the face before beginning to locate and track it, instead of detecting it first before recognising it.

Furthermore, if the movement and temporal information of faces are taken into consideration, the recognition result from the previous time steps will considerably facilitate the detection and tracking of faces at present time.

Another related problem to face detection is pose estimation, which has been discussed in Chapter 3. It is difficult to detect faces with large pose variation using a unimodal detector since the appearance of faces can change dramatically with the pose change. If the pose information of a face is available, the problem of multi-view face detection can be eased to some degree. However, this leads to the following dilemma: one needs to know the position of a face to estimate its pose, and one needs to know the pose of a face before detection.

In this chapter, we will discuss in detail the problem of face detection, especially multi-view face detection, in a systematic manner. First, previous studies are reviewed in Section 4.1. A relatively simpler problem, frontal-view face detection, is discussed in Section 4.2. An appearance based approach using SVMs is adopted to iteratively train a classifier which separates the face patterns from non-face patterns. The more challenging problem, multi-view face detection, is addressed in Section 4.3. We propose a novel approach to this problem by explicitly using the pose information of a possible face, i.e. one estimates the pose of a pattern either it is a face or not, then a view-specific face detector is chosen to determine if it is a face. Detection speed and accuracy are often two conflicting demands in face detection. We implement and evaluate three algorithms, the eigenface method, the SVM based method, and a hybrid method, for face detection. The eigenface method provides fast but less accurate detection, the SVM based method is accurate but slow, and the hybrid method of eigenface and SVM achieves the best balance between accuracy and speed. These methods, as well as the performance evaluation, are presented in Section 4.4. Another important issue, video based face detection, is discussed in Section 4.5. Section 4.6 summaries this chapter.

4.1 Background

Face detection aims to locate the positions and obtain the scales of faces in an input image or video sequence. According to the mechanisms of constructing and manipulating a face model, the previous work on face detection falls into the following categories: Neural Network (NN) based approach, statistical approach, knowledge-based approach, and feature-based approach.

4.1.1 Neural Network Based Approach

NNs have been widely used for face detection and recognition. The most frequently adopted architectures include Multi-Layer Perceptrons (MLPs) and Radial Basis Functions (RBFs).

Soulie *et al.* (Soulie et al., 1993) described a system using neural networks for face detection. They implemented a multi-modular architecture where various rejection criteria are employed to trade-off false recognition against false rejection.

Another NN-based face detection system was presented by Sung and Poggio. They designed 6 positive prototypes (faces) and 6 negative prototypes (non-faces) in the hidden layer. Supervised learning is performed to determine the weights of these prototypes to the output node (Sung and Poggio, 1994).

Rowley *et al.* (Rowley et al., 1996) introduced a neural network based upright frontal face detection system. A retinally connected neural network examines small windows of an image and decides whether each window contains a face. The system arbitrates between multiple networks to improve performance over a single network. Later on, they extended the work to rotation invariant face detection by designing an extra network to estimate the rotation of face in the image plane (Rowley et al., 1997; Rowley et al., 1998a).

McKenna *et al.* (McKenna et al., 1996; McKenna and Gong, 1996) presented an integrated face detection-tracking system with a closed-loop in which a motion-based tracker is used to reduce the search space for an NN-based face detection whilst the latter is used to aid motion tracking and resolve ambiguities in grouping of visual motion. An MLP was trained using back-propagation with iterative selection of false-positive non-face patterns in this system.

One of the promising properties of NNs is that they implicitly define a *non-linear mapping* from the observable feature space to the interpretation of patterns. For example, an MLP with three layers of weights and sigmoidal activation functions can approximate any smooth mapping to arbitrary accuracy (Lapedes and Faber, 1987; Bishop, 1995). In spite of the existence of solutions, implementation is still problematic. For example, it is difficult to determine the number of nodes in the hidden layer if only limited prior knowledge of a problem is available.

4.1.2 Statistical Approach

Moghaddam and Pentland (Moghaddam and Pentland, 1994; Pentland et al., 1994; Moghaddam and Pentland, 1997; Moghaddam et al., 1998) introduced the eigenface method, where the probability of face patterns is modelled by the "distance-in-feature-space" (DIFS) and "distance-from-feature-space" (DFFS) criteria. When detecting faces, a Maximum Likelihood principle is applied as non-face patterns are not modelled in their approach. For example, if only one face exists in a given image, the image patch with maximal value of probability is taken as final detection.

Gong *et al.* (Gong et al., 1998a) used general and modified Hyper Basis Function (HBF) networks with Gaussian mixture models to estimate the density function of face space with large pose variation. As a result, face recognition can be performed more successfully than either of the linear models. Three types of HBF networks with radial Gaussians, diagonal Gaussians and full covariance Gaussians were experimented in this work.

Osuna *et al.* (Osuna et al., 1997b; Osuna et al., 1997c) presented an SVM-based approach to frontal-view face detection. Unlike the eigenface method where only the positive density is estimated, this approach seeks to learn the boundary between face and non-face patterns. After learning, only the "important" examples located on the boundary are selected to build the decision function. The other training examples, which are irrelevant to separate the two classes, are abandoned. Since the SVM method is based on Structural Risk Minimisation, it can automatically control the generalisation performance to avoid over-fitting.

Ng and Gong (Ng and Gong, 1999b; Ng and Gong, 1999a) extended SVMs to model the appearance of faces with large pose change. The view sphere was divided into five smaller, more localised yaw segments. On each of the segments, an SVM based face detector was trained. For face detection across the view sphere, a composite classifier was constructed from all the component SVM detectors.

4.1.3 Knowledge-based Approach

Yang and Huang (Yang and Huang, 1994) presented a hierarchical knowledge-based system to locate human faces in a complex background. The system consists of three levels: the higher two levels are based on mosaic images at different resolutions, and in the lower level, an improved edge detection method is employed.

4.1.4 Feature-based Approach

Brul and Perona (Burl and Perona, 1996) proposed a framework for recognising planar object classes, such as recognising faces from images, based on local feature detectors and a probabilistic model of the spatial configuration of the features.

Yow and Cipolla (Yow and Cipolla, 1996b) proposed a face detection framework that groups image features into meaningful entities using perceptual organisation, assigns probabilities to each of them, and reinforces the probabilities using Bayesian reasoning. They claimed that the framework can be applied to face detection under scale, orientation and viewpoint variations (Yow and Cipolla, 1996a).

4.1.5 Discussions

Face detection can be defined as a classification problem to separate face patterns from non-face patterns. Traditional statistical methods seek to solve this problem by estimating the probability density functions of both face and non-face classes. In practice, there are mainly three obstacles for these methods:

1. the high dimensionality of patterns;
2. the possible number of non-face patterns is extremely large and their distribution is very much irregular;
3. it may also be difficult to model the probability distribution of face patterns, especially the multi-view face patterns, with a unimodal function.

Moghaddam and Poggio (Moghaddam and Pentland, 1994; Pentland et al., 1994; Moghaddam and Pentland, 1997; Moghaddam et al., 1998) tried to address the problem of high dimensionality using PCA to linearly extract the most significant modes of face patterns. They successfully established a statistical density model based on these abstract features (eigenfaces). However, non-face patterns are not modelled in their approach.

Sung and Poggio (Sung and Poggio, 1994) modelled the non-face patterns by 6 prototypes in their approach. A bootstrapping method is employed for the collection of near-face negative patterns. However, the model of the non-face class is built in a predetermined way.

The SVM-based approach (Osuna et al., 1997b; Osuna et al., 1997c) seems to be a promising method to solve this problem. Instead of estimating the densities of face and non-face classes, it seeks to model the boundary of the two classes. Moreover, the generalisation performance, or the capacity of the learning machine, is automatically maintained by the principle of Structural Risk Minimisation.

Most of the previous work is limited to the frontal view. The problem of dealing with rotation in depth and hence being able to detect faces across multiple views remains difficult. The most adopted approach to dealing with the problem of multi-view face detection is to build multiple view-based face detectors, i.e. dividing the view sphere into several small segments and construct one detector on each of the segments (Moghaddam and Pentland, 1997; Gong et al., 1998a; Ng and Gong, 1999b; Ng and Gong, 1999a). Nevertheless, a new problem is normally introduced in these view-based approaches: since the pose of a face is unknown before detection, which detector should we choose to determine if it is a face? One possible solution is to use the *maximum-likelihood* principle, which requires to compute all the view-based detectors. Undoubtedly, it is computationally inefficient.

In this research, an approach to multi-view face detection based on pose estimation is presented. We also decompose the problem of face detection across large range of views into a set of sub-problems, each of them for a small range of views. However, by using the pose information, only one of the view-based detectors is chosen to determine if a pattern is a face. In addition, motion and skin colour detection is employed for selective attention, which results in some sub-images that may contain faces. Face detection is then performed by scanning these sub-images with different sizes of searching windows. Three methods for multi-view face detection are designed: the eigenface and SVM-based methods extended to the multi-view case, and a novel combination of the two methods aiming to improve the overall performance in terms of speed and accuracy.

4.2 Frontal-View Face Detection

Before moving on to multi-view face detection, we first discuss a simple case: frontal-view face detection. We employed the SVM to construct a classifier which separate face patterns from non-face patterns. The method is adopted from (Osuna et al., 1997a; Osuna et al., 1997b; Osuna et al., 1997c) except that we used PCA to represent face patterns in a low-dimensional feature space for fast run-time performance.

4.2.1 Preprocessing of Face Patterns

The frontal-view face database used in this research contains 4040 face images from more than 40 people. The size of these images are scaled to 20×20 since we intend to achieve a fast real-time performance and to construct a low-resolution oriented system which is very promising in many applications such as visual surveillance and video conferencing. The face images are taken under various illumination conditions and the database has been expanded with slightly tilted images from the original ones. Figure 4.1 shows some example images from the database.

We designed a 20×20 mask (as shown in Figure 4.2) to crop out the pixels near the four corners of a face image to diminish the influence of background. Then the masked image is normalised using histogram equalisation (Sonka et al., 1996) in order to compensate for illumination change. Figure 4.2 shows an original face

Figure 4.1. Sample face images from the frontal-view face database. The size of these face images is 20×20. *The images are taken under various illumination conditions and the database has been expanded with images which are slightly tilted from the original ones.*

image, the mask, the masked image, and the normalised image.

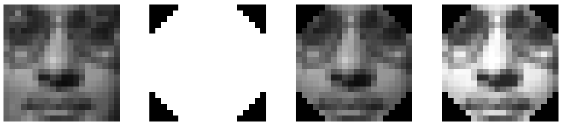


Figure 4.2. Preprocessing of face images. From left to right, the original face image, the mask designed to diminish the influence of background, the masked image, and the histogram-equalised image (for illumination compensation).

4.2.2 Representing Face Images Using PCA

To extract the statistically significant features of face images and to reduce the dimensionality, PCA is applied to the 4040 training face images. The first 20 eigen faces obtained from PCA is shown in Figure 4.3, and the first 5 modes, which change from the mean face by $[-5, +5]$ standard deviation, are shown in Figure 4.4.

It is noted that the training set contains face images only. Since the distributions of non-face patterns could be very wide and irregular, it is not appropriate to model the non-face patterns with a unimodal technique like PCA.

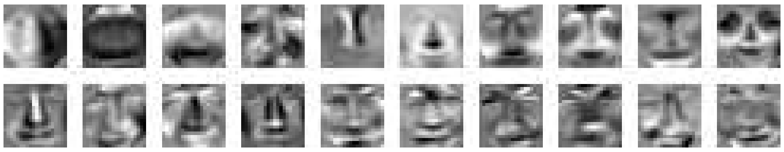


Figure 4.3. The first 20 eigen faces obtained by performing PCA on 4040 training face images. The elements of each eigenface vector have been normalised to $[0, 255]$ *for visualisation.*

Figure 4.4. The first 5 modes of PCA change from the mean face (the middle column) by $[-5, +5]$ *standard deviation.*

4.2.3 Iteratively Training the SVM-based Face Detector

An SVM based face detector can be constructed by solving the following Quadratic Programming problem:

$$\text{maximise} \qquad W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \tag{4.1}$$

$$\text{subject to} \qquad \sum_{i=1}^{l} \alpha_i y_i = 0 \tag{4.2}$$

$$0 \leq \alpha_i \leq C, \quad i = 1, 2, \ldots, l \tag{4.3}$$

where y_i is the label of a training example $\mathbf{x}_i$ which takes value 1 for face and -1 for non-face, k is a kernel function, and C is the upper bound of the Lagrange multiplier α_i.

For a new pattern $\mathbf{x}$, the trained face detector gives an output

$$f(x) \;=\; \sum_{i=1}^{l} y_i \alpha_i k(\mathbf{x}, \mathbf{x}_i) + b \tag{4.4}$$

where b is the bias. More details about the SVM and its algorithms are given in Appendix D.

Since the number of non-face patterns is very large, it is impossible to collect all non-face patterns before training the SVM based face detector. A boot-strapping method (Osuna et al., 1997b) is adopted for iterative training. An arbitrary collection of non-face patterns, which can be cropped randomly from scenic pictures, are chosen as the first set of negative examples for training. Then one applies the resulting detector on some scenic pictures which do not contain any faces. If the detector reports positive output (false positive), save these detections for further training. This process can be repeated iteratively until satisfactory results are achieved. Figure 4.5 shows the intermediate results of boot-strapping on a picture.

Figure 4.5. The boot-strapping method for training the SVM based face detector. The false positive detections are marked with white boxes.

A decomposition algorithm based on the LOQO (Vanderbei, 1994; Vanderbei, 1997) algorithm is developed to train the SVM based face detector. The parameters used in the experiments are list in Table 4.1.

C (upper bound of the Lagrange multipliers)	100
Image dimension	400 (20×20)
PCA dimension	20
Number of face examples	4040
Number of nonface examples (final)	8362

Table 4.1. Parameters used to train the SVM based face detector.

In addition to the parameters listed in Table 4.1, the choice of the kernel function is an important issue for construction of an SVM face detector. We trained four detectors on the same set of examples using different kernel functions:

Linear kernel:	$k(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y})$
Gaussian kernel:	$k(\mathbf{x}, \mathbf{y}) = exp(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{(2\sigma^2)})$
Polynomial kernel (1):	$k(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y})^d$
Polynomial kernel (2):	$k(\mathbf{x}, \mathbf{y}) = (1 + \mathbf{x} \cdot \mathbf{y})^d$

Receiver Operating Characteristic (ROC) curves (Swets and Pickett, 1982) are used to assess the performance of the detectors. Given a test set containing POS face patterns and NEG non-face patterns, one obtains TP correctly detected face patterns (true positive) and FP wrongly detected non-face patterns (false positive). The true positive rate (r_{tp}) and false positive rate (r_{fp}) are defined as:

$$r_{tp} = \frac{TP}{POS} \tag{4.5}$$

$$r_{fp} = \frac{FP}{NEG} \tag{4.6}$$

The ROC curves demonstrates the relation between r_{tp} and r_{fp} by generating a curve with a continuously varying threshold. As shown in Figure 4.6, assuming that r_{fp} is the horizontal coordinate and r_{tp} the vertical coordinate, an optimal detector should have a curve that goes steeply from the bottom left corner to the top left corner, and then to the top right corner. The diagonal from the bottom left corner to the top right corner corresponds to a random detector with equal probabilities for face and non-face. Usually the integral under the curve is used as a performance measure. In other words, the closer a curve to the top left corner, the better performance it achieves.

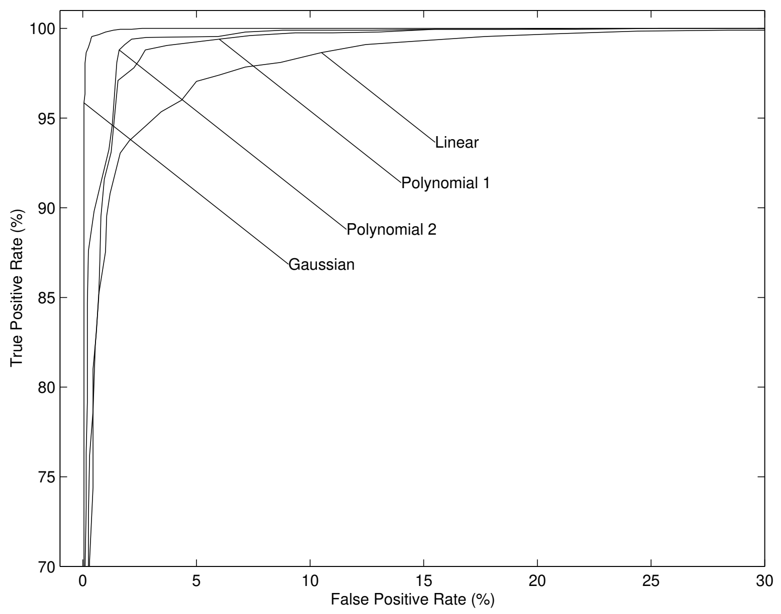


Figure 4.6. The ROC curves of four SVM based face detectors with Gaussian ($2\sigma^2 = 1$), linear, polynomial 1 and polynomial 2 ($d = 2$) kernels respectively. Gaussian kernel achieves the best performance among all.

The ROC curves of the four detectors with different kernels are shown in Figure 4.6. In these experiments, 2000 face patterns and 2000 non-face patterns which do not appear in the training set are used as test examples. It is noted that the Gaussian kernel achieves the best performance while the linear kernel achieves the worst. Figure 4.7 shows the variation of r_{tp}, r_{fp} and the overall error rate of the face detector with Gaussian kernel with respect to the threshold change. Some sample SVs of this detector are shown in Figure 4.8.

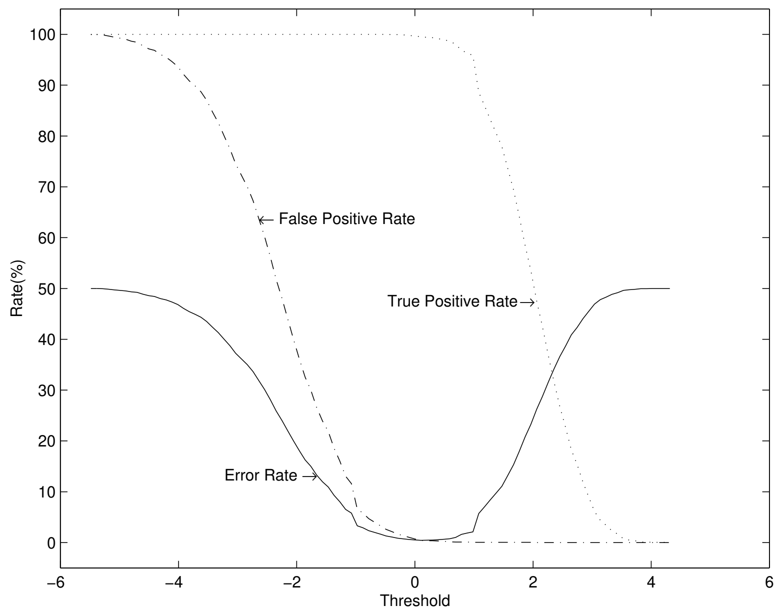


Figure 4.7. Variation of False Negative Rate, False Positive Rate, and overall Error Rate with respect to threshold.

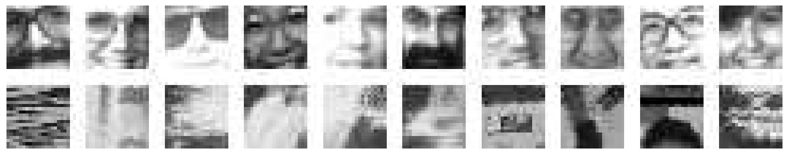


Figure 4.8. Sample SVs of the frontal-view face detector using Gaussian kernel. Top row: positive (faces) SVs, bottom row: negative (nonfaces) SVs.

4.2.4 Face Detecting

On static images, face detection is normally carried out by exhaustively scanning the images with different scales. Each sub-image obtained from the scan is fed into a face detector. If the output of the detector is above a preset threshold, a detection is reported. Figure 4.9 shows the typical results of face detection on a static image.

If colour images are available, the colour information can be used to segment the skin-colour "blobs" which may contain faces. Face detection can then be per-

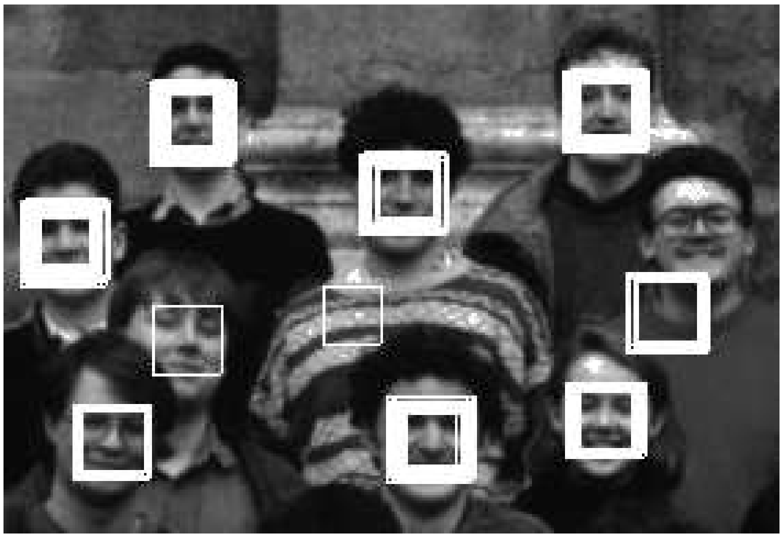


Figure 4.9. Face detection on a static image. Multiple detections may be found for a single face.

formed on the segmented sub-images. This kind of segmentation can significantly reduce the computation in exhaustive scanning. Moreover, if face detection is carried out on video input, the motion of a face, as well as the skin-colour information, can be used for selective attention in order to improve the performance in terms of both speed and accuracy. We will discuss this issue in Section 4.5.

4.3 Multi-View Face Detection Based on Pose Estimation

The appearance of faces can change significantly when observed from different views. This change makes the distribution of multi-view face patterns very irregular, and furthermore, makes the task of estimating the density of these patterns or the boundary between face and non-face patterns more difficult.

A straightforward way for multi-view face detection is to build a single detector which deals with all views of faces. An alternative approach is to build several detectors, each of them corresponding to a specific view. In run-time, if one or more of the detectors give positive output for a given pattern, a face is considered as detected. Previous studies showed that the first method led to poor performance as it fails to deal with the irregular variations of faces across multi-views (Moghaddam and Pentland, 1997; Gong et al., 1998b). The second approach usually performs better than the first one but the computation is expensive since all the multi-view face detectors need to be computed for a given pattern. A novel approach to the problem is presented in this research, which estimates the “pose” of a given pattern before choosing only one of the view-based face detectors to determine whether the targeted image pattern is a face.

4.3.1 Problem Decomposition

As stated above, the problem of multi-view face detection can be decomposed into a set of sub-problems, each for a specific range of views. In this work, we divide the view space into 8 segments: left profile, left frontal, right frontal, right profile in the horizontal direction (yaw), and upper and lower in the vertical direction (tilt), as shown in Figure 4.10. When constructing the view based piece-wise face detectors, we adopt the following strategies.

1. Faces are symmetrical along the vertical line across the nose-bone, so the right view faces can be converted to left view without losing the general face characteristics. Based on this, one only needs to model the multi-view faces either in the left or the right view. As illustrated in Figure 4.10, only four detectors are constructed.

2. Soft boundaries between segments, i.e. overlap by 10° between neighbouring segments, are introduced to provide seamless detection.

3. The vertical separating angle is 0° in tilt, and horizontal $\pm 45^\circ$ in yaw with an intention to separate one-eye faces (profile) from two-eye faces (frontal) effectively.

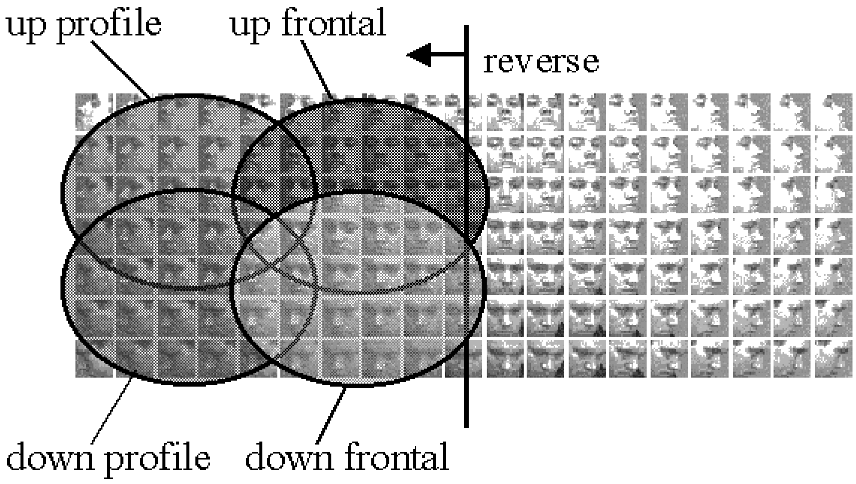


Figure 4.10. Modelling multi-view faces. Four detectors are modelled based on the symmetry property of human face: up profile, up frontal, down profile, down frontal.

4.3.2 Multi-View Face Detection Based on Pose Estimation

When detecting multi-view faces in an image or sub-images obtained from selective attention, the following procedure can be adopted:

1. Estimate the pose in yaw and tilt for each image patch from exhaustive search.

2. Choose an appropriate face detector using the pose information to determine if the image patch is a face.

3. After the scan, there may be several positive detections obtained. The optimal detection can be chosen using, for example, maximum likelihood method.

It seems that extra computation is applied in pose estimation for each image patch. However, the estimated pose can be used to choose the appropriate face detector, so further computation is only applied to one of the detectors, therefore more computation is actually saved in determining if the image pattern is a face. Otherwise, one has to compute all the detectors and to synthesise the outputs of these detectors for a final detection.

Also, the pose information needed in this circumstance is only for the purpose of choosing a detector, therefore a coarse pose estimator can be trained and used to reduce the computation. Nevertheless, when the final detection is obtained, a more precise pose estimator can be employed to re-estimate the pose if more accurate pose estimation is needed for further face tracking and recognition.

It is important to point out that when face detection is performed from video input, it is not necessary to follow the procedure described above strictly. Firstly, motion estimation, skin colour detection, and background subtraction can be used to reduce the searching region to small sub-images that may contain faces. Secondly, the pose information in the previous frame can be conveniently used for the detector selection in the current frame. However, when the pose information is not available or not reliable, for example, in the first frame, or when detection failure occurs, the whole procedure is needed to recover detection. The issue of video-based face detection will be described in Section 4.5.

4.4 Algorithms

In this work, three methods for multi-view face detection, the eigenface method, the SVM-based method, and a hybrid method of eigenface and SVM, are designed and compared with each other.

4.4.1 Eigenface Method

Moghaddam and Pentland introduced the eigenface method, where the confidence $P(\mathbf{x})$ of a pattern $\mathbf{x}$ being a face is modelled by the "distance-in-feature-space" (DIFS) and "distance-from-feature-space" (DFFS) criteria,

$$P(\mathbf{x}) = \left[\frac{exp(-\frac{1}{2}\sum_{i=1}^{M}\frac{u_i^2}{\lambda_i})}{(2\pi)^{M/2}\sum_{i=1}^{M}\lambda_i^{1/2}} \right] \left[\frac{exp(-\frac{\epsilon^2(\mathbf{x})}{2\rho})}{(2\pi\rho)^{(N-M)/2}} \right] \tag{4.7}$$

where λ_i is the ith eigenvalue, u_i is the projection onto the ith eigenvector, N is the total number of eigenvectors, M is the number of significant eigenvectors selected in the model, and ρ is an approximation factor (Moghaddam and Pentland, 1997; Moghaddam et al., 1998).

When detecting faces, a *maximum likelihood* strategy is used which takes an image patch with the maximal value of $P(\mathbf{x})$ as the final detection. A more general statistical strategy for the classification problem can be described as finding the separating threshold (an optimal confidence value) for the two classes, for example, face and non-face. This strategy is illustrated in Figure 4.11. The two curves stand for the distributions of confidence values of the two classes. In a two-class classification problem such as face detection, since we are only interested in discriminating positive patterns from negative ones, the eigen-decomposition is performed on the positive class only. Thus the value of a positive pattern is more likely to be higher and the positive curve is usually located to the right of the negative curve. One can choose the optimal separating threshold t_o as the confidence value of the intersection point of the two curves, assuming equal priors for the two classes.

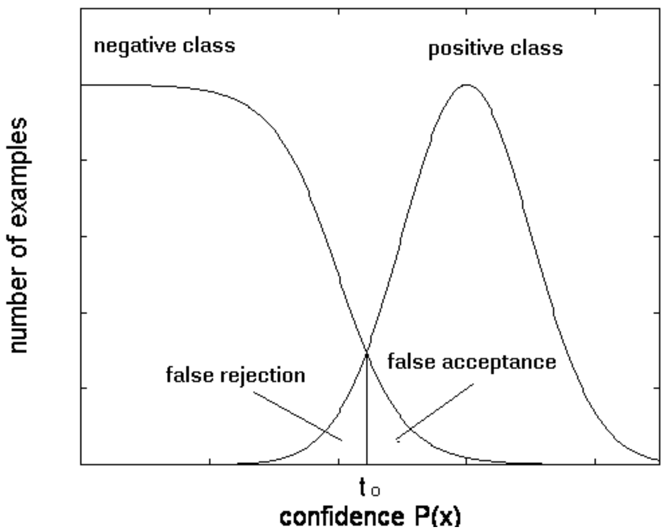


Figure 4.11. Eigenface method for classification.

4.4.2 SVM-Based Method

Alternatively, an SVM-based face detector can be designed. In this work, the approach reported in (Osuna et al., 1997b; Osuna et al., 1997c) is extended to the multi-view situation based on pose estimation. It is interesting to note that:

1. while the eigenface method models the probability density of face patterns, the SVM-based method only models the boundary between faces and non-faces;

2. by solving a Quadratic Programming problem, the SVM-based method is guaranteed to converge to the global optimum;

3. the solution is expressed directly by a subset of "important" training examples called Support Vectors.

4.4.3 A Hybrid Learning Approach of eigenface and SVM

From the description above, one notices that the SVM method seeks to model the boundary of classes, thus it is more accurate when detecting faces. However, as the boundary is constructed by a subset of examples appearing in the training set, which are not necessarily the optimal ones, the computation on these examples, i.e. SVs, may be expensive. On the other hand, the eigenface method is faster than the SVM method, but less accurate since the false negative and false acceptance regions may be larger.

Aiming to achieve improved overall performance in terms of both speed and accuracy, a novel approach which combines the eigenface and the SVM methods together is presented. A schematic illustration of the classification criterion of the hybrid method is given in Figure 4.12.

The whole process consists of a coarse detection phase by the eigenface method followed by a fine SVM phase. In the first phase, the probability density of each class is estimated as simply as possible. Unlike the eigenface model shown in Figure 4.11, two thresholds, a rejection threshold (t_r) and an acceptance threshold

(t_a), are defined. For a test sample $\mathbf{x}$, if the value of $P(\mathbf{x})$ given by Equation (4.7) is less than t_r, it is rejected as a negative example. If the value is larger than t_a, it is accepted as positive. Otherwise, if the value falls between t_r and t_a, it is considered as ambiguous and left to the SVM classifier in the next phase.

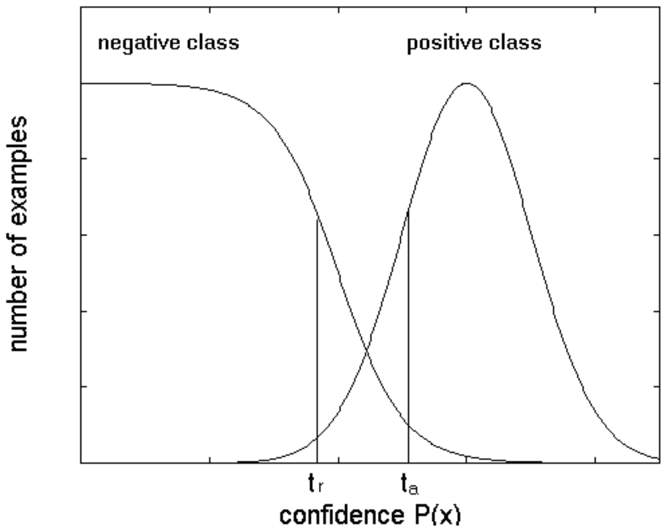


Figure 4.12. A hybrid model of eigenface and SVM.

An SVM-based classifier is trained using the examples in the middle region of Figure 4.12. The classifier is only activated when an ambiguous pattern emerges. In most cases, the SVM-based classifier is computationally more expensive than the eigenface method, but more accurate. However, since the proportion of the examples in the ambiguous region is relatively small, a significant improvement of the classification speed can be achieved.

Furthermore, due to the fact that the SVM classifier is trained only on the examples in the ambiguous region and not on the whole training set, the SVM classification problem is simplified to some degree. A more precise and compact set of SVs are obtained.

Suppose $\mathcal{C}_1$ and $\mathcal{C}_2$ are the positive and negative classes to be classified, and n_1 and n_2 are the number of examples of the two classes, as illustrated in Figure 4.12. The two classification criteria, t_r and t_a, define the false negative rate r_{fn} and false positive rate r_{fp} respectively in the training example set:

$$r_{fn} = \frac{|\{\mathbf{x} : P(\mathbf{x}) < t_r, \mathbf{x} \in \mathcal{C}_1\}|}{n_1 + n_2} \tag{4.8}$$

$$r_{fp} = \frac{|\{\mathbf{x} : P(\mathbf{x}) > t_a, \mathbf{x} \in \mathcal{C}_2\}|}{n_1 + n_2} \tag{4.9}$$

One can determine the expected values of r_{fn} and r_{fp} *a priori* through the training data set. For example, one can train the SVM on a small sample from the training data set and yield r_{fn} and r_{fp}. Then the expected r_{fn} and r_{fp} can be set correspondingly. A conservative approach to determining r_{fn} and r_{fp} is to make them small enough so that more examples are handled by the SVM-based classifier. In this case, the final performance in terms of error rate and speed is close to the SVM method.

After calculating the expected number of false negatives $n_{fr} = r_{fn} \cdot (n_1 + n_2)$ and the number of false positives $n_{fa} = r_{fp} \cdot (n_1 + n_2)$, t_r is set to the n_{fr}th smallest value of $P(\mathbf{x})$ in $\mathcal{C}_1$, and respectively, t_a the n_{fa}th largest value of $P(\mathbf{x})$ in $\mathcal{C}_2$.

4.4.4 Experiments and Analysis

When training each multi-view face detector, the face images corresponding to the specific view range are selected as positive examples. Negative examples (non-faces) are collected by a boot-strapping technique (Sung and Poggio, 1994) from a set of scenic pictures. All example images are scaled to 20×20 pixels.

In the experiments, 2660 face images of 20 subjects were selected as positive examples (faces) from the same database for pose estimation (see Section 3.3), with pose changing from -90° to 90° in yaw and from -30° to 30° in tilt. The non-face images were collected as the “false positive” detections when bootstrapping the SVM face detector on a set of scenery pictures which do not contain any face. All images were scaled to 20×20 pixels. The method described in Section 3.3 is used for image preprocessing.

The results from the three methods on a test sequence are illustrated in Figure 4.14, while some sample frames of the sequence is shown in Figure 4.13. The ground-truth position of the face on each frame is obtained from the multi-view

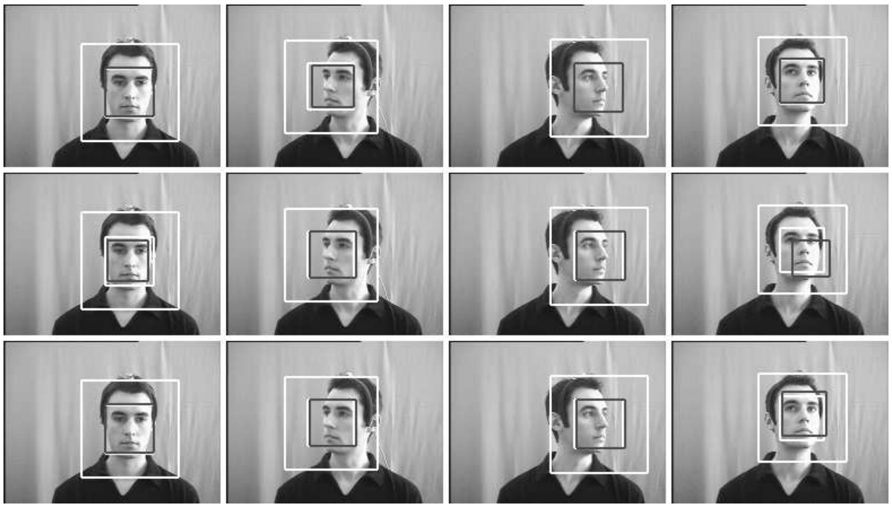


Figure 4.13. Sample frames from a test sequence. From top to bottom are the face detection results of the SVM, eigenface and hybrid methods. For each frame, detection is performed within the outer box. The small white box is the ground-truth position of the face, and the dark box is the detected face pattern.

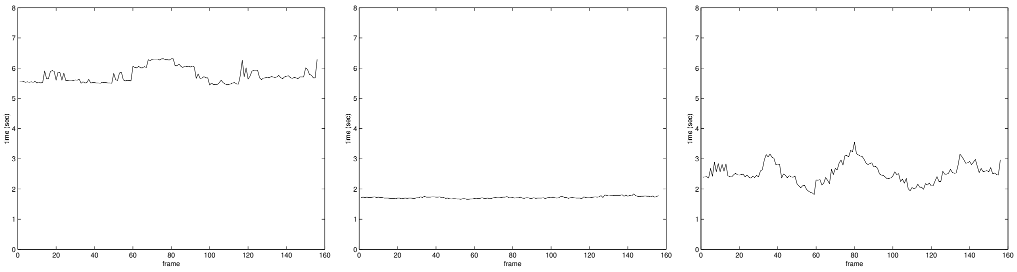
face acquisition system described in Section 3.2.2. Face detection is performed within the outer box with a doubled size of the ground-truth box. The reason of using this bounding box is only to ensure that the number of computation on each frame is equal so that the results are comparable through the whole sequence. We will demonstrate in the next section that motion and skin-colour can be used effectively to determine the bounding boxes on which face detection is performed.

The experimental results indicate that:

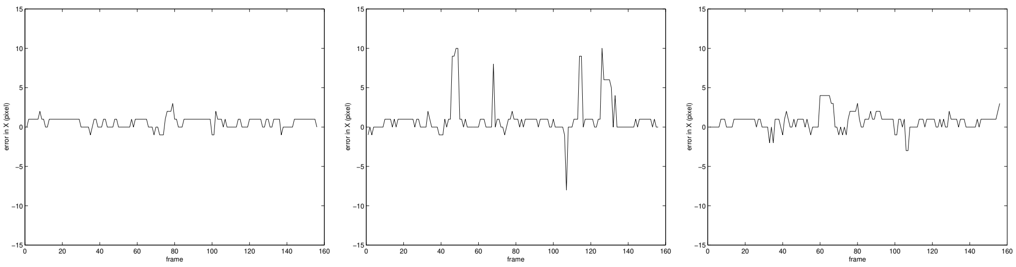
1. The SVM method is the most accurate in terms of error in detection scale and location, but also the slowest;

2. The eigenface method is the fastest, but less accurate in certain frames;

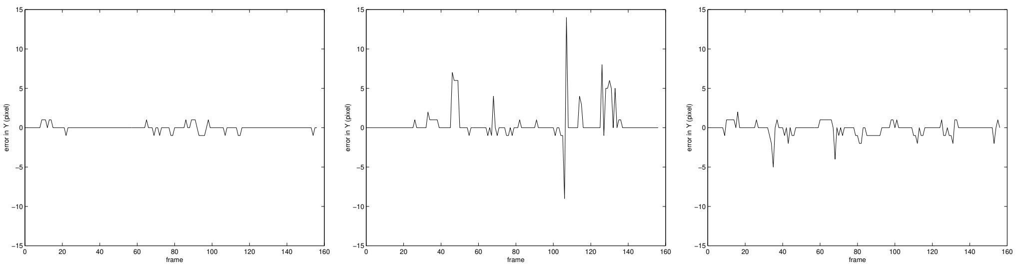
3. The hybrid method demonstrates the best balance between accuracy and speed; it is almost as accurate as the SVM method and not significantly slower than the eigenface method in most frames.



(a) Detection time



(b) Error in X



(c) Error in Y

Figure 4.14. Comparison results of, from left to right, the SVM, eigenface and hybrid methods for multi-view face detection on a test sequence. (a) shows the detection time in second on each frame. (b) and (c) are the position errors in pixels from the ground-truth position in horizontal (X) and vertical (Y) direction respectively.

4.5 Detecting Faces Dynamically from Video

So far we have discussed the issue of face detection purely from the viewpoint of pattern recognition, i.e. iteratively collecting face patterns and non-face patterns, then constructing a pattern classifier based on these data. However, for a realistic system where video input is available, there are two very important factors which can be used to efficiently improve the performance of face detection: motion (McKenna and Gong, 1996; McKenna et al., 1996) and skin colour (Kjeldsen and Kender, 1996; McKenna et al., 1997; Raja et al., 1998b; McKenna et al., 1998; Raja et al., 1998a). In other words, the motion and skin colour cues can be used to segment in images the regions of interest which most likely contain faces, then detection can be carried out on the segmented sub-images only. This achieves a considerable reduction in computation. [1]

4.5.1 Motion Detection

Human faces are usually undergoing continuous movement. Our biological vision system is very sensitive to this movement and uses it as a cue to effectively focus attention. Moreover, we can initiatively make this kind of movement to attract attention and convey useful information. For computer vision systems, although robust estimation and grouping of visual motion consistently over time can be problematic, qualitative and partial estimation of motion is usually sufficient to select and segment the sub-regions of a whole image which possibly contain faces (McKenna and Gong, 1996; McKenna et al., 1996).

Assuming the image intensity change is mainly from motion, the temporal difference of two successive frames is computed by:

$$\frac{\partial I(x,y,t)}{\partial t} = I(x,y,t) - I(x,y,t-1) \tag{4.10}$$

where I is the image intensity, x, y are the pixel position in the images, and t is the time. For colour image, I can be computed by averaging the three chromatic components: red, green and blue (RGB). If the value of Equation (4.10) is above a preset threshold, the pixel (x, y) is regarded as on a moving object.

[1] The code on selective attention using motion and skin-colour detection is kindly provided by Jamie Sherrah.

4.5.2 Skin Colour Detection

The skin colour of a face can be affected by various factors, which include the extrinsic factors such as illumination, camera characteristics and shading, and intrinsic factors such as skin tone of different ethnic groups and physiological changes of face (e.g. blushing and paling). Despite of these potential sources of variability, previous studies indicated that the face colours occupy a relatively compact distribution in a colour space, e.g. the Hue-Saturation (HS) space (Hunke and Waibel, 1994; Raja et al., 1998b; McKenna et al., 1998). In many situations, the skin colour can be sufficiently modelled by a single Gaussian, or more precisely, by mixture of Gaussians, where the probability density of a pixel $\boldsymbol{\xi}$ belonging to the skin colour is given by:

$$p(\boldsymbol{\xi}) = \sum_{j=1}^{m} p(\boldsymbol{\xi}|j)P(j) \tag{4.11}$$

where $P(j)$ is the mixing parameter of component j, $\boldsymbol{\xi}$ is the pixel colour vector comprised of HS or normalised RGB values, and $p(\boldsymbol{\xi}|j)$, the density of component j, is constructed with mean $\boldsymbol{\mu}_j$ and covariance matrix $\boldsymbol{\Sigma}_j$:

$$p(\boldsymbol{\xi}|j) = \frac{1}{2\pi|\boldsymbol{\Sigma}_j|^{\frac{1}{2}}} exp\left\{-\frac{1}{2}(\boldsymbol{\xi} - \boldsymbol{\mu}_j)^{\mathrm{T}}\boldsymbol{\Sigma}_j^{-1}(\boldsymbol{\xi} - \boldsymbol{\mu}_j)\right\} \tag{4.12}$$

We adopted the approach introduced by (Raja et al., 1998b; McKenna et al., 1998) to construct the colour model. Once the model has been constructed, a look-up table can be created off-line for fast real-time performance.

4.5.3 Grouping Motion and Colour for Selective Attention

The motion information, i.e. the temporal difference of successive frames in our case, normally sketches the *contour* of a moving object which corresponds to the entire body of a person [2]. Skin colour, on the other hand, typically provides *regions* of pixels which are usually located on faces, hands, and arms. The motion and skin colour cues can be used complementally for selective attention.

[2] The local texture variation, for example, the pixels around the eyebrows, eyes, nose, and mouth, may also be reflected.

In this work, we firstly carry out motion detection on *sub-sampled* images, then perform skin-colour detection on the resulting pixels of motion detection $\mathcal{M}$, obtaining pixel set $\mathcal{S}$. The number of pixels in $\mathcal{M}$ is considerably smaller compared to the original size of images. The pixels in $\mathcal{S}$ are clustered into several "blobs" based on spatially connected histogram bins. Since people usually stand upright in the images, we can obtain a bounding box for each of these "blobs" with its sides parallel to the image axes.

The process of grouping motion and skin colour is described in Table 4.2.

1	sub-sample the image
2	detect motion using (4.10) and store thresholded pixels into set $\mathcal{M}$
3	detect skin colour using (4.11) on pixels in $\mathcal{M}$, then save thresholded pixels to $\mathcal{S}$
4	cluster $\mathcal{S}$, obtaining the regions of interest
5	detect faces on each of the regions of interest

Table 4.2. Group motion and skin colour for selective attention.

It is important to note that three issues have been considered for computation reduction and real-time performance:

1. image sub-sampling;
2. look-up table for skin-colour detection;
3. motion detection, where the computation is slightly lighter, is performed first, then colour detection is carried out on the "moving" pixels only.

Sample frames of a sequence with the results of motion-colour based selective attention (big boxes) and face detection (small boxes) are shown in Figure 4.15.

4.6 Summary

In this chapter, we have discussed the problem of face detection, from the simple frontal-view problem to the more challenging multi-view problem, from static to

Figure 4.15. Face detection using SVM classifier on a video sequence. The bigger boxes are obtained by motion-colour based selective attention. Face detection is then performed on these bounding boxes only. The final detections are labelled with the smaller boxes inside the bigger ones.

dynamic, and from the eigenface algorithm, the SVM based algorithm, to a hybrid algorithm.

Frontal-view face detection has been intensively addressed previously. We implemented an appearance based face detection system mainly based on the method proposed in (Osuna et al., 1997b; Osuna et al., 1997c). In principle, face detection is a classification problem of separating face patterns from non-face patterns. Therefore, estimating a boundary which robustly separates the two classes of patterns is more promising than other methods such as estimating the density function of the face patterns or the density functions of both face and non-face patterns since we only need to compute the face and non-face patterns located around the boundary. This is actually the underlying characteristic of SVMs. By iteratively collecting near-face negative patterns using a prototype face detector, we can gradually refine the detector, making it well fitted on the boundary between face and non-face patterns.

Unlike the frontal-view problem, detecting faces with large pose variation is more challenging since the severe nonlinearity caused by rotation in depth, self-shading and self-occlusion yields an extraordinarily irregular distribution of face patterns. The straightforward method, constructing a single universal detector, proved to be inefficient. Some researchers tried to build view-based piece-wise multiple models to address this problem (Moghaddam and Pentland, 1994; Ng and Gong, 1999b; Ng and Gong, 1999a). However, computation is intensified since a pattern needs to be evaluated on more models. In this work, we presented a novel approach to this problem by explicitly using the pose information. By determining firstly the possible pose of a pattern, only the classifier for this specific pose is needed for evaluation. Moreover, the computation on pose estimation, which may be intuitively regarded as an extra burden, does not impose a significant influence on the real-time performance of a system since a "cheap" pose estimator, which provides a coarse estimation, is sufficient in this case.

To improve the overall performance of face detection in terms of both speed and accuracy, three methods for multi-view face detection are implemented and compared with each other. The eigenface method and the SVM method were extended to the case of multi-view face detection. Experimental results show that the eigenface method is faster but less accurate as there is a relatively big overlap between the confidence distributions of face and non-face classes, while the SVM method is more accurate but slower since the number of SVs cannot be efficiently controlled at a low level. By combining the two methods together, a novel method is proposed which keeps the advantages and suppresses the disadvantages of both methods. The properties of the hybrid method lie in:

1. most "obvious" patterns are determined by the eigenface method which is fast;

2. the ambiguous patterns are classified by the SVM method which is accurate;

3. the acceptance and rejection thresholds are calculated based on a preset detection accuracy which guarantees the final accuracy is in an acceptable level;

4. the SVM classifier is trained only on a small set of ambiguous patterns, thus it is more accurate and faster.

This hybrid method can also be applied to other classification problems.

Another interesting issue is dynamic face detection from video input. In this situation, the motion and skin colour of faces provide an enriched information for detection. Although robust motion estimation and colour constancy over time can be problematic, a relatively simple method, which adopts temporal differencing for motion estimation, mixture of Gaussians for skin colour modelling, and grouping motion and skin colour for selective attention, has proved to be sufficient to improve the real-time performance. In this approach, we adopted several strategies for performance improvement: sub-sampling images, using look-up table for skin colour detection, and applying colour detection on "moving" pixels only.

Chapter 5

A Multi-View Dynamic Face Model

Human faces contains various information such as shape, texture, pose, illumination, and expression. A good face model should be capable of separating these different types of information and expressing them quantitatively with a set of model parameters. In most systems, model fitting is usually performed after the approximate location of a face is obtained by, for example, face detection or selective attention using motion/colour.

5.1 Background

Modelling faces across multiple views is one of the most challenging problems because of the rotation in depth, self-occlusion, self-shading, and the consequent non-linearity in both shape and texture. To address this problem, three categories of approaches have been proposed in the previous research.

Some researchers adopted view-based 2D models to solve this problem. Moghaddam and Pentland (Moghaddam and Pentland, 1994) presented a view-based and modular eigenspace method. Ng and Gong (Ng and Gong, 1999b; Ng and Gong, 1999a) introduced a view-based piece-wise SVM model of the face space. Cootes et al (Cootes et al., 2000) proposed the view-based Active Appearance Models which employ three models for profile, half-profile and frontal views. The limitation of these approaches lies in the fact that the division of the face space is often arbitrary, coarse, and ad hoc.

The second type of methods is to use non-linear 2D models. For example, Romdhani *et al.* (Romdhani et al., 1999b) developed a multi-view appearance model using KPCA. The non-linearity of KPCA enables the model to deal with large pose variation, but at the price of intensive computation.

Another kind of methods to address the multi-view problem is to use 3D models. DeCarlo and Metaxas (DeCarlo and Metaxas, 1996a) presented a 3D deformable face model in which optical flow and edge information are combined. Their model successfully tracked faces in sequences with significant expression change and pose change. Jebara and Pentland (Jebara and Pentland, 1997) proposed an approach to recover the 3D face structure using Structure from Motion. The estimation of the 3D structure is further constrained for reliable feature tracking by a 3D range data model of an average human face. Vetter and Blanz (Vetter and Blanz, 1998) introduced a flexible 3D face model learnt from examples of 3D range face data. A novel 2D face image can be matched to the 3D model in an *analysis-by-synthesis* manner. Then images of the novel face in different views, illumination, and expression can be synthesised by changing the parameters of the matched model. 3D face models have also been used for person-independent face tracking and feature detection (Li et al., 1993; Shakunaga et al., 1998).

It is important to point out that face recognition is more than static image matching. When a moving face is observed continuously, the spatial and, in particular, the temporal characteristics about the face can provide far enriched information for recognition (Gong et al., 2000). Exploring the dynamics of a moving face from video input is another challenging problem for face recognition. In the previous research, some researchers have presented some preliminary approaches to address the problem. Gong et al. (Gong et al., 1994) introduced an approach that uses Partially Recurrent Neural Networks to recognise temporal signatures of faces. Edwards et al. (Edwards et al., 1998b) proposed an integrated approach to decouple the identity variance from the residual variance of pose, lighting and expression. By learning the correlation between the two parts of variance online, a class-specific refinement for the identity covariance can be achieved. Yamaguchi et al. (Yamaguchi et al., 1998) presented a method for face recognition from sequences by building a subspace for the detected faces from a given sequence and then matching the subspace with prototype subspaces.

To comprehensively address the two problems stated above, we present an integrated multi-view dynamic face model. It consists of three parts: a sparse 3D shape model trained from 2D images labelled with pose and landmarks, a *shape-and-pose-free* texture model, and an affine geometrical model.

We propose to address the non-linear problem of multi-view faces by modelling and representing faces with a sparse 3D shape model. Meanwhile, by warping the facial texture to the mean shape at the frontal view, the normalised *shape-and-pose-free* texture can be obtained.

The second problem, exploring the dynamics of faces, is addressed in the following aspects: When fitting the model to a moving face, the temporal information of the face over time, together with the global appearance of the face and the local appearance of the facial landmarks, is taken into consideration. Also, we present a Kalman filter based method for temporal model parameter estimation. Furthermore, face recognition is performed by matching the temporal trajectories constructed from the discriminating features of a moving face over time. The last issue will be discussed in Chapter 7.

The remaining part of this chapter is arranged as follows: Section 5.2 gives the details of model components and model construction. A model fitting algorithm is presented in Section 5.3 which is formulated by optimising the global fitting criterion of the overall face appearance, the local fitting criterion on a set of 2D landmarks, and the temporal fitting criterion between the information on successive frames of a sequence. Section 5.4 describes the issue of temporal model fitting, i.e. obtaining a robust estimation of model parameters dynamically from sequences where faces are undergoing large pose changes. Section 5.5 summaries the research presented in this chapter.

5.2 Multi-View Dynamic Model

Our multi-view dynamic face model consists of a sparse 3D Point Distribution Model (PDM) (Cootes et al., 1995b) learnt from 2D images in different views, a *shape-and-pose-free* texture model, and an affine geometrical model which controls the rotation, scale and translation of faces. The first two parts of the model aim to represent the identities of faces to be analysed, while the latter is used for alignment and tracking.

5.2.1 Constructing 3D Shape from Labelled 2D Images

Modelling the appearance of faces with large pose variation is non-trivial for 2D models due to the severe non-linearity. But if 3D geometrical information is available, this situation can be alleviated to some extent. A straight-forward way to collect 3D information about faces is to use sensors such as a 3D laser scanner. However, the huge amount of 3D range data involved may bring a heavy burden to the computation. Another difficulty comes from establishing the correspondence between dense 3D data. In this work, we learn a 3D face shape model containing only a sparse set of feature points from 2D face images at different views.

Our database includes 2D face images from 12 subjects, 133 poses of each subject. All face images were chosen without spectacles on (see (Gong et al., 2000) for more details of the data acquisition process). As described in Chapter 3, the pose of a face is defined by two parameters: tilt and yaw (α, β), the rotation angles about horizontal and vertical axes respectively. The rotation in the image plane is not taken into account on the basis that human heads are assumed to be mostly upright. A sparse set of 44 landmarks locating the mouth, nose, eyes, and face contour were semi-automatically labelled on each face image. Figure 5.1 illustrates the landmarks used in this work and the triangulation formed from these landmarks which can be used to warp multi-view faces onto the frontal

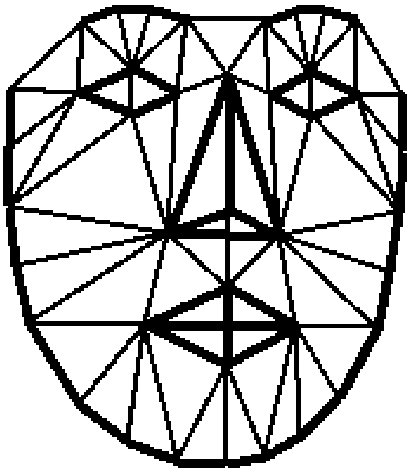


Figure 5.1. Landmarks and triangulation of the face model.

view (details will be discussed in Section 5.2.3). Figure 5.2 shows the sample face images used to construct the model and also the landmarks labelled on each image.

Figure 5.2. Sample training face images (first row) and the landmarks labelled on the images (second row).

Given a set of 2D face images with known positions of the landmarks and pose, the 3D positions of the landmarks can be estimated using linear regression. The rotation centre used to measure the pose angles is assumed at the centre of the eye centres and the mouth centre (see Section 3.2.1). We set this point as the origin of the object coordinate system.

Orthographic projection is adopted for simplicity. Suppose the 3D coordinates of a landmark in the object coordinate system is (X, Y, Z), the position of this landmark in the 2D image with pose (α, β) is given by:

$$(x, y)^{\mathrm{T}} = \mathbf{R}(\alpha, \beta)\ (X, Y, Z)^{\mathrm{T}} \tag{5.1}$$

where $\mathbf{R}(\alpha, \beta)$ is the rotation matrix for pose (α, β) obtained by rotating about the horizontal axis first by α and then about the vertical axis by β.

$$\mathbf{R}(\alpha, \beta) = \begin{bmatrix} cos(\beta) & 0 & sin(\beta) \\ sin(\alpha)sin(\beta) & cos(\alpha) & -sin(\alpha)cos(\beta) \end{bmatrix} \tag{5.2}$$

Note that the results are only slightly different if rotating in the reverse order, i.e. first β, then α.

If $M (M \geq 2)$ face images in different poses are available, one can estimate the 3D coordinates (X, Y, Z) of a landmark using linear regression by

$$\text{Minimise} \quad \sum_{i=1}^{M} \left((x - x_i)^2 + (y - y_i)^2\right) \tag{5.3}$$

subject to :

$$(x_i, y_i)^{\mathrm{T}} = \mathbf{R}(\alpha_i, \beta_i)\ (X, Y, Z)^{\mathrm{T}}, \quad i = 1, 2, ..., M \tag{5.4}$$

where (x_i, y_i) is the known 2D position of the landmark and (α_i, β_i) is the pose of the landmark in the ith face image. Then the 3D shape vector $\mathbf{p}$ is obtained as:

$$\mathbf{p} = (X_1, Y_1, Z_1, X_2, Y_2, Z_2, ..., X_{N_l}, Y_{N_l}, Z_{N_l})^{\mathrm{T}} \tag{5.5}$$

where N_l is the number of landmarks.

Figure 5.3 shows a 3D shape pattern with tilt fixed on 0° and yaw changing from −40° to +40° . This shape pattern is estimated from the labelled face images in Figure 5.2.

Ideally, the larger the range of poses covered by the training images, the more accurate the 3D position. However, when a face rotates to nearly profile view, some of the landmarks are invisible in the image. Therefore, for each subject, 45 of the 133 face images with poses between [−20° , 20°] in tilt and [−40° , 40°] in yaw are selected for training. Also, the training set M should be adequately large. In our experiments, a random selection of 20 out of 45 face images from each

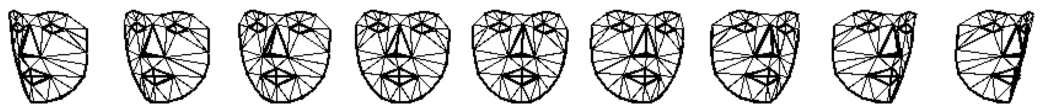


Figure 5.3. A 3D shape rotates from $-40°$ *to* $+40°$ *in yaw (tilt fixed on* $0°$ *). The shape vector is estimated from the face images shown in Figure 5.2.*

subject is used to learn the 3D shape vector of all landmarks. For each subject, 50 shape vectors are estimated in this manner in order to learn the statistical 3D PDM of faces. This will be further discussed in Section 5.2.2.

5.2.2 A Sparse 3D PDM of Faces

Although only a sparse set of 44 landmarks are chosen to represent the 3D shape of faces, the dimensionality is still too high to fit the shape model. However, the shapes of human faces are able to be represented in an even lower dimensional shape space since they share a very similar structure. The PDM is adopted to construct this low dimensional shape space.

Performing PCA on N given 3D face shape vectors $\{\mathbf{p}_i, i = 1, 2, ..., N\}$, which are estimated using the method described in Section 5.2.1, one obtains the mean shape $\bar{\mathbf{p}}$ and the eigen matrix $\mathbf{U}$ which is comprised of the first N_s significant eigen vectors

$$\mathbf{U} = [\mathbf{u}_1 \mathbf{u}_2 ... \mathbf{u}_{N_s}] \tag{5.6}$$

A shape pattern $\mathbf{p}$ can then be represented by a vector in the PDM space

$$\mathbf{s} = \mathbf{U}^{\mathrm{T}}(\mathbf{p} - \bar{\mathbf{p}}) \tag{5.7}$$

whose dimension is N_s. The reconstructed 3D shape from $\mathbf{s}$ is obtained from

$$\mathbf{p}_r = \mathbf{U}\mathbf{s} + \bar{\mathbf{p}} \tag{5.8}$$

We trained the PDM on a set of 600 3D shape patterns from 12 different subjects (50 of each subject) with pose changes between $[-20^\circ, 20^\circ]$ in tilt and $[-40^\circ, 40^\circ]$ in yaw. Each 3D shape pattern was estimated from a random selection of 20 of 45 face images of the same subject as stated in Section 5.2.1.

It is important to point out that the reason for using the small range of pose *in the training stage* is to make sure all landmarks are visible in the image. Otherwise, if some landmarks are invisible, it would be difficult to label the positions of these landmarks. However, this constraint is not imposed when fitting the model onto a novel image or sequence. It will be shown later that the model can be fitted successfully even in the profile view where nearly half of a face is invisible in a 2D image.

Figure 5.4 shows the projection, on $\{-40^\circ, -20^\circ, 0^\circ, +20^\circ, +40^\circ\}$ in yaw (from left to right), of the first shape mode changing from the mean shape by $\{-3, 0, 3\}$ of standard deviation (from top to bottom). The first 10 eigenshapes account for 95.5% of all variance.

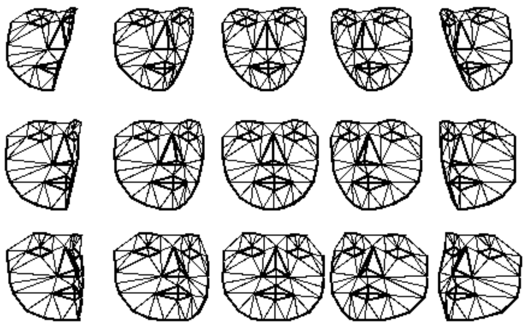


Figure 5.4. The first mode of variation of the 3D PDM rotating by $\{-40^\circ, -20^\circ, 0^\circ, +20^\circ, +40^\circ\}$ in yaw (from left to right), and changing by $\{-3, 0, 3\}$ of standard deviation from mean shape (from top to bottom).

5.2.3 A Shape-and-Pose-Free Texture Model

There is no doubt that texture carries as important information as shape. However, accurately modelling facial texture is non-trivial due to its sensitivity to changes in illumination, pose, and expression. In this work, we focus mainly on the problem of modelling facial texture variation arising from pose change. Explicitly modelling surface reflection and shading properties provides one possible solution to this problem (Atick et al., 1996; Zhao and Chellappa, 2000). As an alternative, we present here a statistical approach to model face textures by extracting *shape-and-pose-free* texture information.

To decouple the covariance between facial texture and shape, the facial texture is warped to the mean shape at frontal view (with $0°$ in both tilt and yaw). This is implemented by forming a triangulation from the landmarks and employing a piece-wise affine transformation between each of the triangle pairs (see Figure 5.1). By warping to the mean shape, one obtains the shape-free texture of a given face image. Furthermore, by warping to the frontal view, a pose-free texture representation is achieved. Figure 5.5 illustrates the *shape-and-pose-free* texture patterns of the face images shown in Figure 5.2.

It is noted that when one side of a face becomes partially invisible, the texture pattern is constructed from the visible side using the bilateral symmetry of faces.

Figure 5.5. Extracted shape-and-pose-free *texture patterns of the face images shown in Figure 5.2.*

We applied PCA to a set of 540 *shape-and-pose-free* face textures from 12 subjects with pose changes between $[-20°, 20°]$ in tilt and $[-40°, 40°]$ in yaw (45 from each subject). The first 12 eigen modes account for 96.4% of all variance. Figure 5.6 shows the first three texture modes varying from the mean texture by $[-4, +4]$ standard deviation.

Figure 5.6. The first three eigen textures changing from the mean texture by $[-4, +4]$ *standard deviation .*

During the fitting process, a *shape-and-pose-free* texture pattern $\mathbf{q}$ of a face image, which is already warped to the mean shape in the frontal view, can be represented by

$$\mathbf{t} = \mathbf{V}^{\mathrm{T}}(\mathbf{q} - \bar{\mathbf{q}}) \tag{5.9}$$

where $\bar{\mathbf{q}}$ is the mean texture, and $\mathbf{V}$ is constructed by the first N_t significant eigen vectors of the texture PCA

$$\mathbf{V} = [\mathbf{v}_1\mathbf{v}_2 ... \mathbf{v}_{N_s}] \tag{5.10}$$

The reconstruction of the texture pattern is given by

$$\mathbf{q}_r = \mathbf{V}\mathbf{t} + \bar{\mathbf{q}} \tag{5.11}$$

5.2.4 Representing Face Patterns

Based on the analysis above, a face pattern can be represented in the following way. First, a 3D shape model is fitted to a given image or video sequence containing faces. The shape parameters of the fitted face is given by Equation (5.7). The face texture is warped onto the mean shape of the 3D PDM model at the frontal view. Then the texture parameters of the face are computed using Equation (5.9). Finally, by adding parameters controlling pose, shift and scale, the complete parameter set of the dynamic model for a given face pattern is

$$\mathbf{c} = (\mathbf{s}, \mathbf{t}, \alpha, \beta, dx, dy, r)^{\mathrm{T}} \tag{5.12}$$

where (α, β) is pose in tilt and yaw, (dx, dy) is the translation of the centroid of the face, and r is its scale.

The parameter set consists of two parts: the identity information $(\mathbf{s}, \mathbf{t})$ which is crucial to face recognition and facial analysis, and the geometrical information $(\alpha, \beta, dx, dy, r)$ which is important for face alignment and tracking.

5.3 Model Fitting

Model fitting in this context is the problem of searching for the optimal parameters of the model for an unknown face image to be interpreted, and it is given by:

$$\mathbf{c}^* = argmin(L(\mathbf{c})) \tag{5.13}$$

where $L(\mathbf{c})$ is a loss function which evaluates how well the model fits onto the image.

5.3.1 Loss Function for Fitting

We formulate the loss function as

$$\begin{aligned} L(\mathbf{c}) \;=\; & \|\mathbf{q}_r(\mathbf{c}) - \mathbf{q}\| + \\ & \xi \sum_{i=1}^{N_l} w_i \mathcal{M}(\hat{\mathbf{F}}_i(\mathbf{c}), \mathbf{F}_{i0}) + \\ & \eta \sum_{i=1}^{N_l} w_i \mathcal{M}(\hat{\mathbf{F}}_i(\mathbf{c}), \hat{\mathbf{F}}_i(\mathbf{c}_{-1})) \end{aligned} \tag{5.14}$$

The first term on the right-hand side evaluates the difference between the image appearance and the model synthesised appearance, where $\mathbf{q}_r(\mathbf{c})$ is the reconstructed texture given by (5.11), and $\mathbf{q}$ is the original texture warped onto the mean shape at frontal view. This is based on the principle of *analysis-by-synthesis* (Ezzat and Poggio, 1996; Cootes et al., 1998; Vetter and Blanz, 1998). The better the model fits, the smaller the difference.

The second term, which is measured in Mahalanobis distance, describes the local texture similarity of each landmark to the template of this specific landmark estimated from training images, where $\hat{\mathbf{F}}_i(\mathbf{c})$ is the response of Gabor wavelet filters (Lades et al., 1993) or derivatives of Gaussian, on the current position of the ith landmark. The same filters have been applied to the training face images described in Section 5.2.3. A set of templates, one for each landmark, is obtained using PCA. $\mathbf{F}_{i0}$ denotes the template centroid. The Mahalanobis distance $\mathcal{M}(\hat{\mathbf{F}}_i(\mathbf{c}), \mathbf{F}_{i0})$ is calculated using the notion of distance-in-feature-space (DIFS) (Moghaddam and Pentland, 1997). Each $\mathcal{M}(\hat{\mathbf{F}}_i(\mathbf{c}), \mathbf{F}_{i0})$ is weighted by w_i, which measures the visibility of the ith landmark. The value of w_i is computed from the normal of the landmark on the 3D shape. ξ is a normalisation coefficient, and N_l is the number of landmarks. It was noted in our experiments that the Gabor wavelet filter does not outperform the simpler derivatives of Gaussian.

The last term, which is only enabled when the input is a video sequence, compares the difference between the filtered local texture around each landmark $\hat{\mathbf{F}}_i(\mathbf{c})$ and that in the previous frame $\hat{\mathbf{F}}_i(\mathbf{c}_{-1})$. The Mahalanobis distance $\mathcal{M}(\hat{\mathbf{F}}_i(\mathbf{c}), \hat{\mathbf{F}}_i(\mathbf{c}_{-1}))$ is also calculated using DIFS. η is a normalisation coefficient.

The loss function defined in (5.14) can be interpreted as follows: it is a weighted summation of the fitting criterion of the *global* appearance to the model synthesised appearance, the *local* fitting criterion around each landmark, and the *temporal* fitting criterion to the pattern in the previous frame.

5.3.2 A Fitting Algorithm

Based on stochastic search, the fitting algorithm of the multi-view face model is described in Table 5.1. The evaluation of the loss function used in step 4 is carried out as described in Table 5.2.

The Support Vector Machine based method described in Section 3.4 was used for real-time pose estimation in Step 1. Figure 5.7 illustrates the process of applying the above algorithm to a face image.

1	assume initial parameter $\mathbf{c}_0 = (\mathbf{s}, \alpha, \beta, r, dx, dy)$
2	randomly sample n parameter points around the initial $\mathbf{c}_0$
3	randomly sample m parameter points around each of the n points
4	evaluate the values of the loss function $L(\mathbf{c})$ for each of the $m \times n$ parameters
5	sort the loss function values in ascending order
6	if no improvement from the top value, stop
7	otherwise, save the first n parameters, then go to 3

Table 5.1. Fitting Algorithm

Figure 5.7. Fit the multi-view face model to a face image. The first row shows the original face image and the fitted pattern warped on the original image. The second row lists the fitting results in 10 iterations.

5.4 Fitting the Model to Sequences Over Time

By fitting the multi-view face model to face images, one extracts and separates the identity parameters and geometrical parameters from the raw images. A solution to this problem can be greatly improved when a continuous video input is available. From video sequences, not only can more information across views and over time be used for model fitting, but also, the temporal continuity provides the possibility to exploit the facial dynamics encoded in the input stream.

1 perform pose estimation using $(\mathbf{s}, r, dx, dy)$

2 restore 2D shape using $(\mathbf{s}, \alpha, \beta, r, dx, dy)$

- reconstruct 3D shape $\mathbf{p}_r$ from $\mathbf{s}$ using (5.8)
- project $\mathbf{p}_r$ to (α, β)
- scale to r and translate to (dx, dy)

3 evaluate the global fitting criterion given as the first term in (5.14)

- warp the texture enclosed by the 2D shape to the mean shape at frontal view to obtain the *shape-and-pose-free* texture $\mathbf{q}$
- compute the texture parameter $\mathbf{t}$ by projecting $\mathbf{q}$ using (5.9)
- reconstruct $\mathbf{q}_r$ using (5.11)
- calculate the similarity

4 sample and filter the local texture around each landmark

5 evaluate the local fitting criterion of landmarks given by the second term in (5.14)

6 evaluate the temporal fitting criterion of landmarks, if necessary, given by the third term in (5.14)

7 compute the overall loss in (5.14)

Table 5.2. Evaluation of $L(\mathbf{c})$

5.4.1 Temporal Estimation of Model Parameters

Suppose an input sequence contains one subject whose identity is unchanged throughout the sequence. Fitting the model onto a sequence frame by frame independently is likely to yield fluctuate estimation of the model parameters for the following reasons:

1. There is no identity constancy constraint imposed on the fitting process. Instead, in each frame, it only tries to minimise the loss function given in

(5.14).

2. The fitting algorithm may be attracted to local optima and image noise.

3. Expression and illumination changes may also affect the estimation of model parameters.

This problem is illustrated in Figure 5.8 where the dotted curves show the results when fitting the model frame by frame independently. Only the first two dimensions of the shape vector $\mathbf{s}$ are shown here for clarity.

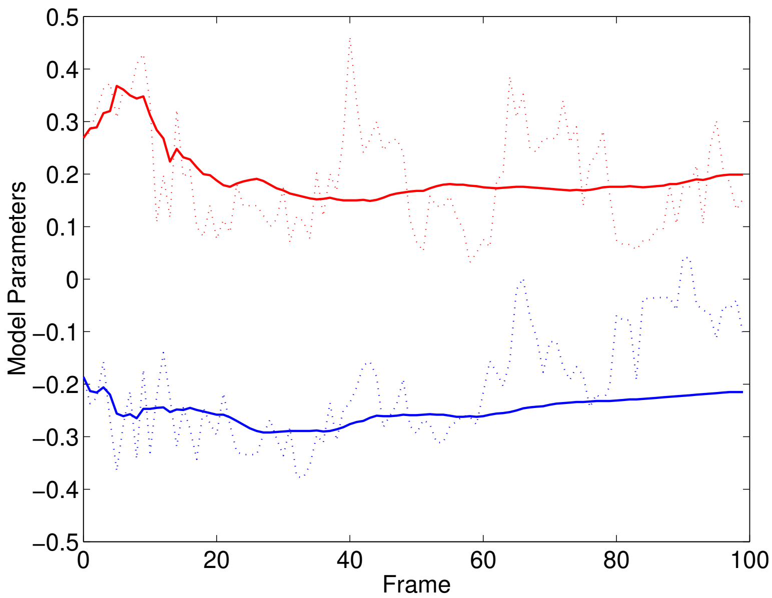


Figure 5.8. Model parameter estimation. The dotted curves are obtained by applying the fitting algorithm frame by frame independently on a sequence, while the solid curves are computed using Kalman filter based temporal estimation.

Under these circumstances, the model fitting problem should be regarded as dynamic parameter estimation of an underlying stochastic process where the identity parameters $(\mathbf{s}, \mathbf{t})$ are kept constant and the geometrical parameters change freely. In the following discussions, we assume that the purpose of model fitting is face recognition, i.e. temporally estimating the identity parameters of faces.

A straightforward approach to estimate the identity parameters temporally is performing Gaussian estimation (Brammer and Siffling, 1989) based on the least squares principle. However, this method computes all the information accumulated in a batch way which is not appropriate for dynamic model fitting. Alternatively, a temporal model such as Kalman filters (Brammer and Siffling, 1989) provides a recursive solution to this problem.

The problem of estimating the identity parameters of the model using Kalman filters can be formulated as follows. For the shape vector, the state transition equation is

$$\mathbf{s}(k) = \mathbf{s}(k-1) \tag{5.15}$$

The observation is taken from the 2D projection of the 3D shape since this is the only visible part of the 3D shape from the images.

$$\mathbf{o}'(k) = \mathbf{R_t}(k)(\mathbf{U}\ \mathbf{s}(k) + \bar{\mathbf{p}}) + \mathbf{w}(k) \tag{5.16}$$

where $\mathbf{w}(k)$ denotes a zero-mean, white observation noise, and $\mathbf{R_t}(k)$ is the rotation and projection matrix extended by $\mathbf{R}$ in (5.2),

$$\mathbf{R_t}(k) = \begin{bmatrix} \mathbf{R} & 0 & ... & 0 \\ 0 & \mathbf{R} & ... & 0 \\ & & ... & \\ 0 & 0 & ... & \mathbf{R} \end{bmatrix} \tag{5.17}$$

Defining

$$\mathbf{H}(k) = \mathbf{R_t}(k)\mathbf{U} \tag{5.18}$$

$$\mathbf{o}(k) = \mathbf{o}'(k) - \mathbf{R_t}(k)\bar{\mathbf{p}} \tag{5.19}$$

the observation equation is then given by

$$\mathbf{o}(k) = \mathbf{H}(k)\mathbf{s}(k) + \mathbf{w}(k) \tag{5.20}$$

Hence, temporal estimation of the model identity parameters can be performed using a Kalman filter as follows:

$$\hat{\mathbf{s}}(k) = \hat{\mathbf{s}}(k-1) + \mathbf{K}(k)[\mathbf{o}(k) - \mathbf{H}(k)\hat{\mathbf{s}}(k-1)] \tag{5.21}$$

$$\mathbf{P}(k) = \mathbf{P}(k-1) - \mathbf{K}(k)\mathbf{H}(k)\mathbf{P}(k-1) \tag{5.22}$$

$$\mathbf{K}(k) = \mathbf{P}(k-1)\mathbf{H}^{\mathrm{T}}(k)[\mathbf{H}(k)\mathbf{P}(k-1)\mathbf{H}^{\mathrm{T}}(k) + \mathbf{Q}]^{-1} \tag{5.23}$$

where $\mathbf{K}$ is Kalman gain, $\mathbf{P}$ is the error covariance matrix, and $\mathbf{Q}$ is the covariance matrix of $\mathbf{w}(k)$ which can be estimated from the training data.

A Kalman filter can also be designed for the texture vector in a similar way. However, unlike the one for the shape vector, where the observation vector is formulated from the 2D projection of the 3D shape, the state vector, i.e. the texture parameters, is fully observable, thus the observation vector and the state vector can be identical.

Applying the Kalman filter based temporal model to the example sequence of Figure 5.8, a stable estimate of the identity parameters shown as the solid curves in Figure 5.8 can be extracted.

5.4.2 Model Generalisation for Tracking Out-of-Range Poses

As stated in Section 5.2.2, the 3D shape PDM is trained from 2D images with limited pose range where all landmarks are visible. To verify if the model generalises well to out-of-range poses, we applied the model to sequences where faces underwent large pose change. The pose range in these sequences were from profile to profile. Figure 5.9 displays the results of temporal model fitting on one of these sequences.

The results depict that the model is capable of coping with large variation of pose even though it is trained on a limited range of views. This can be explained for two reasons. First, the shape information is represented in 3D, so the model can be rotated and projected to 2D for any given pose. Second, in the loss function (5.14), the local and temporal fitting criteria are defined in a pose-specific way since they are weighted by a visibility measure which depends on pose. In all

Figure 5.9. Tracking faces undergoing large pose change. The first rows are original images from sample frames with 8 frame interval, and the second row shows the reconstructed face patterns overlapped on the original images. The length of this sequence is 81 frames.

the experiments, the model has demonstrated a reliable performance between $[-70^\circ, 70^\circ]$ in yaw. However, when the pose is nearly $\pm 90^\circ$, tracking may fail since little information is available in these views.

5.4.3 Tracking Faces with Expression Changes

We also fitted the multi-view dynamic face model on sequences containing faces undergoing significant expression changes. The results from one of these sequences is shown in Figure 5.10. It is noted that the fitting is less good and the expressions are not fully re-constructed. This is mainly due to two reasons: first, all the face images used for training are taken in neutral expression, and second, the number of landmarks is insufficient to model the variation arising from expression change. However, it is important to point out that, due to the averaging and smoothing effect of Kalman filtering, the fitting process still converged to a stable estimation of the subject identity and was shown to be very robust over time despite errors in individual frames.

At this stage of our work, the model has been developed for estimating the identity parameters and is not best suited for modelling expressions. This is partly reflected by the fact that a state-invariant model defined by (5.15) is used on the basis of subject constancy. For accurate expression recognition, a time-variant state equation is needed to model more detailed dynamics of facial shape and texture.

Figure 5.10. Tracking faces with significant expression change. Images are sampled with 5 frame interval. The length of this sequence is 47 frames.

5.5 Summary

In this chapter, we have presented an integrated multi-view dynamic face model which includes a sparse 3D PDM shape model, a *shape-and-pose-free* texture model and an affine geometrical model. This model is aimed at two important issues in face recognition and facial analysis: modelling face appearance with large pose variation and modelling faces dynamically over time. The characteristics of the model include:

1. A 3D PDM shape model is learnt from 2D images labelled with 3D poses and 2D landmarks. Instead of using dense 3D range data, this model consists of a sparse set of landmarks only.

2. A *shape-and-pose-free* texture model is built to decouple the covariance between shape and texture.

3. Although only face images from limited pose range are used in the training stage to ensure all landmarks are visible in the images, this limitation of pose range is never imposed when applying the model for tracking. Experimental results indicate that the model is able to cope with pose variation almost from profile to profile.

4. By applying the model, two sets of information, the identity parameters and geometrical parameters, are obtained. The former is crucial to face recognition and facial analysis, and the latter is important for face alignment and tracking.

5. Fitting criteria are formulated from the global fitting criterion of the entire face, the local fitting criterion of the landmarks and the temporal fitting criterion to previous patterns.

6. Temporal estimation of model parameters is employed to provide a more robust and stable fitting results over time. Where face recognition is involved, Kalman filters with constant state equations can be implemented for temporal estimation.

Chapter 6

Identity Feature Extraction Using Kernel Discriminant Analysis

In classification, the patterns normally exhibit two types of variance: within-class variance and between-class variance. Specifically, the problem of face recognition involves the variance from different people, and the variance caused from pose, expression, illumination changes. In this case, extracting the significant features to maximise the between-class variance and minimise the within-class variance is crucial, especially for multi-view face recognition problem where severe non-linearity is induced by rotation in depth, self-shading, self-occlusion and illumination change.

6.1 Background

PCA has been widely adopted for abstract feature extraction in face recognition. Sirovich and Kirby (Sirovich and Kirby, 1987) first used PCA for face representation. Turk and Pentland (Turk and Pentland, 1991) proposed the eigenface approach which uses a similar method to code face images and capture face features. PCA has also been used extensively in other face models such as the the Active Shape Model (ASM) and Active Appearance Model (AAM) (Cootes et al., 1995b; Cootes et al., 1998).

However, the features extracted by PCA are "global" features for all face classes, thus they are not necessarily representative for discriminating one face class from others. Linear Discriminant Analysis (LDA), which seeks to find a linear transformation by maximising the between-class variance and minimising the within-class variance (Fukunaga, 1972), proved to be a more suitable technique for class separation. Computationally, LDA can be solved as an eigen-decomposition problem similar to PCA. Swets and Weng (Swets and Weng, 1996) applied a subsequent LDA projection followed by PCA to derive the Most Discriminating Features. Zhao *et al.* (Zhao et al., 1998b) used LDA as a representation for frontal-view face recognition. Edwards *et al.* (Edwards et al., 1996) adopted LDA to select discriminating parameters based on Active Appearance Models. They argued that these parameters can be used to decouple identity variance from pose, lighting and expression variance.

Although LDA can provide a significant discriminating improvement to the task of face recognition, it is still a linear technique by its very nature. When severe non-linearity is involved, this method is intrinsically poor. Another shortcoming of LDA lies in the fact that the number of basis vectors is limited by the number of face classes, therefore it would be less representative when small set of subjects are concerned. To extract the non-linear principal components, Kernel PCA (KPCA) was developed (Scholkopf et al., 1997). Romdhani *et al.* (Romdhani et al., 2000b) adopted KPCA to construct a nonlinear model aiming at corresponding dynamic appearances of both shape and texture across views. However, as with PCA, KPCA captures the *overall* variance of all patterns which is inadequate for discriminating purpose.

In this work, Kernel Discriminant Analysis (KDA), a novel non-linear approach which employs the kernel technique to maximise the between-class variance and minimise the within-class variance, is developed to compute the non-linear discriminating basis vectors for multi-view facial feature extraction. In this chapter, the KDA method is introduced in Section 6.2, then the issue of modelling faces using KDA is discussed in Section 6.3 preceded by the summary of this chapter in Section 6.4.

6.2 Kernel Discriminant Analysis

As stated in the previous section, both PCA and LDA are limited to linear problems, and KPCA is designed to deal with the *overall* rather than the *discriminating* variance. In this work, the Kernel Discriminant Analysis, a nonlinear discriminating approach based on the kernel technique (Scholkopf et al., 1997; Vapnik, 1995; Vapnik, 1998), is developed to extract the nonlinear discriminating features for face recognition across multiple views [1].

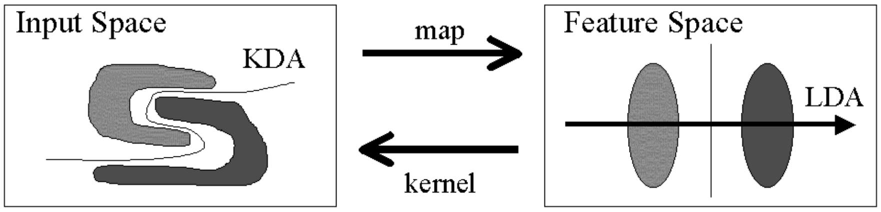


Figure 6.1. Kernel Discriminant Analysis.

The principle of KDA can be illustrated in Figure 6.1. It is difficult to directly compute the discriminating features between the two classes of patterns because of the severe non-linearity. By defining a non-linear map from the input space to a high-dimensional feature space, one obtains a linearly separable distribution in the feature space. Then LDA, the linear technique, can be performed in the feature space to extract the most significant discriminating features. However, the computation may be problematic or even impossible in the feature space due to the high dimension. By introducing a kernel function which corresponds to the non-linear map, all the computation can be carried out in the input space conveniently.

[1] We developed this method independently. However, we have noticed similar approaches have been published by Mika *et al.* (Mika et al., 1999) and Baudat and Anouar (Baudat and Anouar, 2000). Although the basic principle is similar, our approach is different from the others both in problem formulation and algorithm.

6.2.1 Centred Data

For a set of training patterns $\{\boldsymbol{x}\}$ which are categorised into C classes, $\boldsymbol{\phi}$ is defined as a non-linear map from the input space to a high-dimensional feature space. By this map one assumes that an original nonlinear problem in the input space can be transformed to a linear problem in the high-dimensional feature space and solved using regular linear techniques. However, computing $\boldsymbol{\phi}$ explicitly may be problematic or even impossible. The kernel technique provides a subtle solution to this problem. If the map $\boldsymbol{\phi}$ satisfies Mercer's condition (Vapnik, 1995; Vapnik, 1998), then the inner product of two vectors in the feature space can be calculated through a kernel function

$$k(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{\phi}(\mathrm{x}) \cdot \boldsymbol{\phi}(\mathrm{y})) \tag{6.1}$$

which can be conveniently computed in the input space.

Let us first consider the centred data set in the feature space, i.e.

$$\sum_{i=1}^{N} \boldsymbol{\phi}_i = 0 \tag{6.2}$$

where N is the total number of training patterns. Define a between-class scatter matrix $\mathbf{S}_b$ and a within-class scatter matrix $\mathbf{S}_w$ in the feature space as

$$\begin{aligned} \boldsymbol{S}_b &= \frac{1}{C}\sum_{c=1}^{C} \boldsymbol{\mu}_c \boldsymbol{\mu}_c^{\mathsf{T}} && (6.3) \\ \boldsymbol{S}_w &= \frac{1}{C}\sum_{c=1}^{C} \frac{1}{N_c}\sum_{i=1}^{N_c} \boldsymbol{\phi}_i \boldsymbol{\phi}_i^{\mathsf{T}} && (6.4) \end{aligned}$$

where N_c is the number of patters in the cth class, and $\boldsymbol{\mu}_c$ is the mean vector of class c,

$$\boldsymbol{\mu}_c = \frac{1}{N_c}\sum_{i=1}^{N_c} \boldsymbol{\phi}_i \tag{6.5}$$

Since the data are centred, one does not need to subtract the global mean vector from $\boldsymbol{\mu}_c$ and $\boldsymbol{\phi}_i$. Thus the scatter matrices $\mathbf{S}_b$ and $\mathbf{S}_w$ can be expressed in a simplistic form. We will return to the general case where the data are not centred in the next section.

Assuming $\boldsymbol{S}_w$ is not singular, one can maximise the between-class variance and minimise the within-class variance of vectors $\boldsymbol{\phi}_i$ in the feature space by performing eigen-decomposition on matrix

$$\boldsymbol{S} = \boldsymbol{S}_w^{-1}\boldsymbol{S}_b \tag{6.6}$$

Assuming $\boldsymbol{v}$ is one of the eigenvectors of matrix $\boldsymbol{S}$, and λ is its corresponding eigenvalue, i.e.

$$\boldsymbol{S}\boldsymbol{v} = \lambda\boldsymbol{v} \tag{6.7}$$

Combining (6.6) and (6.7), one obtains

$$\boldsymbol{S}_b\boldsymbol{v} = \lambda\boldsymbol{S}_w\boldsymbol{v} \tag{6.8}$$

Then taking the inner product with vector $\boldsymbol{\phi}_m$ on both sides of equation (6.8) yields

$$(\boldsymbol{S}_b\boldsymbol{v}\cdot\boldsymbol{\phi}_m) = \lambda(\boldsymbol{S}_w\boldsymbol{v}\cdot\boldsymbol{\phi}_m), m = 1, 2, ..., N \tag{6.9}$$

A coefficient vector exists

$$\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_N)^{\mathtt{T}} \tag{6.10}$$

that satisfies

$$\boldsymbol{v} = \sum_{n=1}^{N}\alpha_n\boldsymbol{\phi}_n \tag{6.11}$$

Substituting (6.3), (6.4) and (6.11) in (6.9) yields

$$\sum_{n=1}^{N}\alpha_n\sum_{c=1}^{C}\frac{1}{N_c^2}\sum_{i=1}^{Nc}\sum_{j=1}^{Nc}(\boldsymbol{\phi}_{ci}\cdot\boldsymbol{\phi}_m)(\boldsymbol{\phi}_{cj}\cdot\boldsymbol{\phi}_n) = \lambda\sum_{n=1}^{N}\alpha_n\sum_{c=1}^{C}\frac{1}{N_c}\sum_{i=1}^{N_c}(\boldsymbol{\phi}_{ci}\cdot\boldsymbol{\phi}_m)(\boldsymbol{\phi}_{ci}\cdot\boldsymbol{\phi}_n) \tag{6.12}$$

Defining a $N \times N_c$ matrix $\boldsymbol{K}_c$ as

$$(\boldsymbol{K}_c)_{ij} := (\boldsymbol{\phi}_i\cdot\boldsymbol{\phi}_i) = k_{ij} \tag{6.13}$$

and a $N_c \times N_c$ matrix $\mathbf{1}_{Nc}$ as

$$(\mathbf{1}_{Nc})_{ij} := 1 \tag{6.14}$$

one obtains

$$\left(\sum_{c=1}^{C} \frac{1}{N_c^2} \boldsymbol{K}_c \mathbf{1}_{Nc} \boldsymbol{K}_c^{\mathrm{T}}\right) \boldsymbol{\alpha} = \lambda \left(\sum_{c=1}^{C} \frac{1}{N_c} \boldsymbol{K}_c \boldsymbol{K}_c^{\mathrm{T}}\right) \boldsymbol{\alpha} \tag{6.15}$$

Defining $N \times N$ matrix as

$$\boldsymbol{A} = \left(\sum_{c=1}^{C} \frac{1}{N_c} \boldsymbol{K}_c \boldsymbol{K}_c^{\mathrm{T}}\right)^{-1} \left(\sum_{c=1}^{C} \frac{1}{N_c^2} \boldsymbol{K}_c \mathbf{1}_{Nc} \boldsymbol{K}_c^{\mathrm{T}}\right) \tag{6.16}$$

one derives

$$\boldsymbol{A}\boldsymbol{\alpha} = \lambda \boldsymbol{\alpha} \tag{6.17}$$

By eigen-decomposing matrix $\boldsymbol{A}$, one obtains the coefficient vector $\boldsymbol{\alpha}$. Therefore, for a new pattern $\boldsymbol{x}$ in the original input space, one can calculate its projection onto $\boldsymbol{v}$ in the high-dimensional feature space by

$$(\boldsymbol{\phi}(x) \cdot \boldsymbol{v}) = \sum_{i=1}^{N} \alpha_i (\boldsymbol{\phi}_i \cdot \boldsymbol{\phi}(x)) = \sum_{i=1}^{N} \alpha_i k(\boldsymbol{x}, \boldsymbol{x}_i) = \boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{k}_x \tag{6.18}$$

where

$$\boldsymbol{k}_x = (k(\boldsymbol{x}, \boldsymbol{x}_1), k(\boldsymbol{x}, \boldsymbol{x}_2), ..., k(\boldsymbol{x}, \boldsymbol{x}_N))^{\mathrm{T}} \tag{6.19}$$

Constructing the eigen matrix

$$\boldsymbol{U} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_M] \tag{6.20}$$

from the first M significant eigenvectors of $\boldsymbol{A}$, the projection of $\boldsymbol{x}$ in the M-dimensional KDA space is given by

$$\boldsymbol{y} = \boldsymbol{U}^{\mathrm{T}} \boldsymbol{k}_x \tag{6.21}$$

6.2.2 Non-centred Data

In the general case, $\{\boldsymbol{\phi}(\boldsymbol{x}_i)\}, i = 1, 2, ..., N$, are not centred in the feature space. A similar method to (Scholkopf, 1997) is adopted here. By defining

$$\tilde{\boldsymbol{\phi}}_i := \boldsymbol{\phi}_i - \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\phi}_n \tag{6.22}$$

one can use the method stated above since $\{\tilde{\boldsymbol{\phi}}_i\}, i = 1, 2, ..., N$ are now centred. The kernel matrix $\tilde{\boldsymbol{K}}_c$ can then be expressed by its non-centred counterpart $\boldsymbol{K}$ as follows:

$$\begin{aligned}
(\tilde{\boldsymbol{K}}_c)_{ij} &= (\tilde{\boldsymbol{\phi}}_i \cdot \tilde{\boldsymbol{\phi}}_j) \\
&= \left(\boldsymbol{\phi}_i - \frac{1}{N}\sum_{m=1}^{N}\boldsymbol{\phi}_m\right) \cdot \left(\boldsymbol{\phi}_j - \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\phi}_n\right) \\
&= (\boldsymbol{\phi}_i \cdot \boldsymbol{\phi}_j) - \frac{1}{N}\sum_{m=1}^{N}(\boldsymbol{\phi}_m \cdot \boldsymbol{\phi}_j) - \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{\phi}_i \cdot \boldsymbol{\phi}_n) \\
&\quad + \frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N}(\boldsymbol{\phi}_m \cdot \boldsymbol{\phi}_n) \\
&= k_{ij} - \frac{1}{N}\sum_{m=1}^{N}k_{mj} - \frac{1}{N}\sum_{n=1}^{N}k_{in} + \frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N}k_{mn}
\end{aligned} \tag{6.23}$$

Using $N \times N$ matrix $(\boldsymbol{K})_{ij} := k_{ij}$ and $\mathbf{1}_N$, one obtains

$$\tilde{\boldsymbol{K}} = \boldsymbol{K} - \frac{1}{N}\mathbf{1}_N\boldsymbol{K} - \boldsymbol{K}\frac{1}{N}\mathbf{1}_N + \frac{1}{N^2}\mathbf{1}_N\boldsymbol{K}\mathbf{1}_N \tag{6.24}$$

Therefore $\tilde{\boldsymbol{K}}_c$ can be obtained as a sub-matrix of $\tilde{\boldsymbol{K}}$. Then substituting $\boldsymbol{K}_c$ with $\tilde{\boldsymbol{K}}_c$ in (6.16) and eigen-decomposing $\boldsymbol{A}$, one obtains the matrix $\boldsymbol{U}$ in (6.20).

Similar to the centred case given in (6.18), the projection of a new pattern $\boldsymbol{x}$ onto an eigenvector $\tilde{\boldsymbol{v}}$ in the feature space is given by

$$(\tilde{\boldsymbol{\phi}}(\boldsymbol{x}) \cdot \tilde{\boldsymbol{v}}) = \sum_{i=1}^{N}\alpha_i(\tilde{\boldsymbol{\phi}}(\boldsymbol{x}) \cdot \tilde{\boldsymbol{\phi}}(\boldsymbol{x}_i)) = \tilde{\boldsymbol{k}}_x\boldsymbol{\alpha} \tag{6.25}$$

where

$$
\begin{aligned}
(\tilde{\boldsymbol{k}}_x)_i &= \left(\boldsymbol{\phi}(\boldsymbol{x}) - \frac{1}{N}\sum_{m=1}^{N}\boldsymbol{\phi}(\boldsymbol{x}_m)\right)\left(\boldsymbol{\phi}(\boldsymbol{x}_i) - \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\phi}(\boldsymbol{x}_n)\right) \\
&= k(\boldsymbol{x}, \boldsymbol{x}_i) - \frac{1}{N}\sum_{m=1}^{N} k(\boldsymbol{x}_i, \boldsymbol{x}_m) - \frac{1}{N}\sum_{n=1}^{N} k(\boldsymbol{x}, \boldsymbol{x}_n) \\
&\quad + \frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N} k(\boldsymbol{x}_m, \boldsymbol{x}_n) \qquad (6.26)
\end{aligned}
$$

Defining an $N \times 1$ vector $\mathbf{1}'$ with all entries equal to 1, one obtains

$$\tilde{\boldsymbol{k}}_x = \boldsymbol{k}_x - \frac{1}{N}\boldsymbol{K}\mathbf{1}' - \frac{1}{N}\boldsymbol{k}_x\mathbf{1}_N + \frac{1}{N^2}\mathbf{1}'\boldsymbol{K}\mathbf{1}_N \qquad (6.27)$$

Finally, the projection of $\boldsymbol{x}$ in the M-dimensional KDA space is given by

$$\boldsymbol{y} = \boldsymbol{U}^{\mathrm{T}}\tilde{\boldsymbol{k}}_x \qquad (6.28)$$

6.2.3 A Toy Problem

We use a toy problem to illustrate the characteristics of KDA as shown in Figure 6.2. Two classes of patterns denoted by circles and crosses respectively have a significant non-linear distribution. To make the results comparable, we try to separate them with a *one dimensional* feature, i.e. the most significant mode of PCA, LDA, KPCA or KDA. The left column shows the patterns and the discriminating boundaries computed by the four different methods. The right column illustrates the intensity of the one-dimensional features given by PCA, LDA, KPCA and KDA on the region covered by the training patterns.

In this experiment, the discriminating boundary is determined by the value of the discriminating feature which minimises the misclassification from the given patterns (Bishop, 1995). As shown in Figure 6.3, the histogram distribution of the two classes of patterns are illustrated as functions of the discriminating feature x. The shaded region denotes the classification errors. The separating boundary is placed at the point where the area of the shaded region is minimised.

It can be seen clearly that PCA and LDA are incapable of providing correct classification because of their linear nature. Neither does KPCA do so since it is

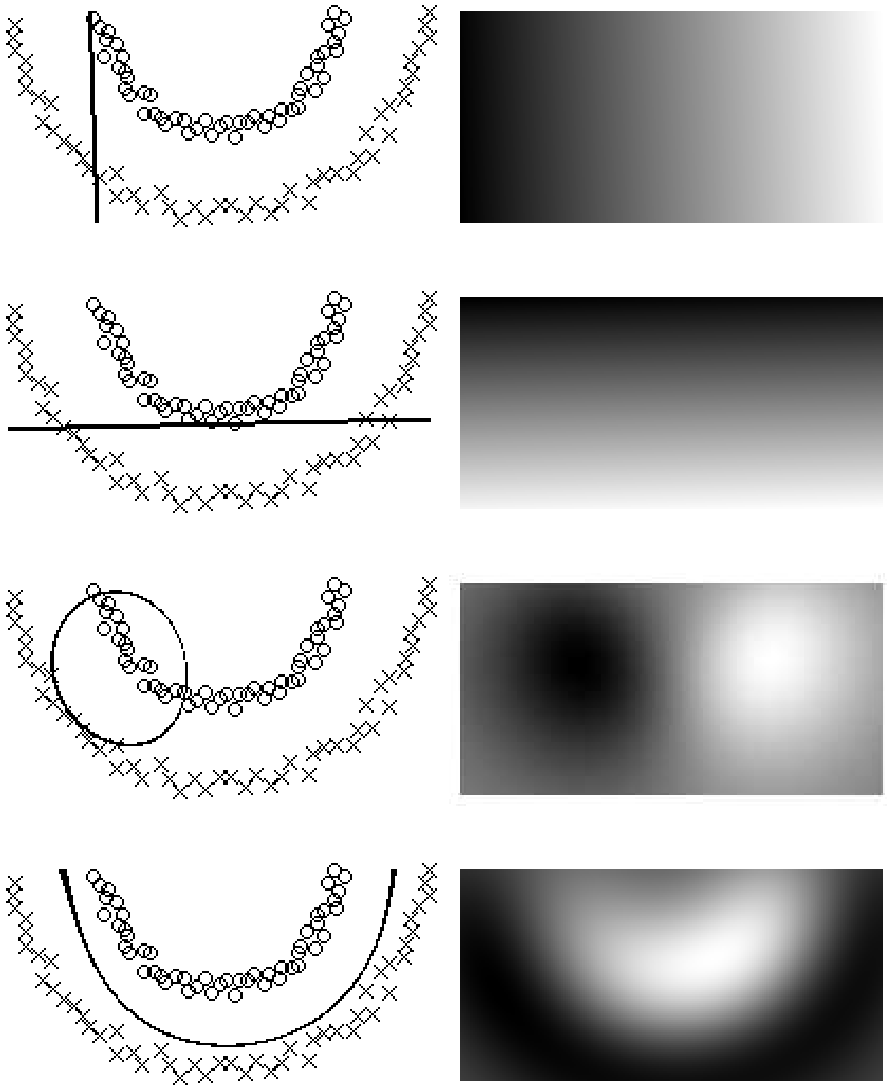


Figure 6.2. Solving a nonlinear classification problem with, from top to bottom, PCA, LDA, KPCA and KDA. The left column shows the patterns and the discriminating boundaries computed by the four methods. The right column illustrates the intensity of the one-dimensional features computed using the four methods.

designed to extract the overall rather than the discriminating variation though it is nonlinear in principle. KDA gives the correct classification boundary, and the feature intensity correctly reflects the actual pattern distribution.

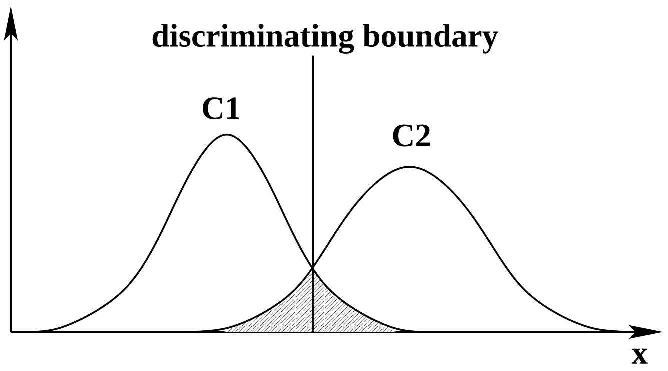


Figure 6.3. Determining the discriminating boundary by minimising the misclassification.

6.3 Representing Multi-View Faces Using KDA

To address the non-linearity brought by the face rotation in depth, we adopt the following approach in this work:

1. using the *shape-and-pose-free* textures to diminish the variance cased by large pose change;

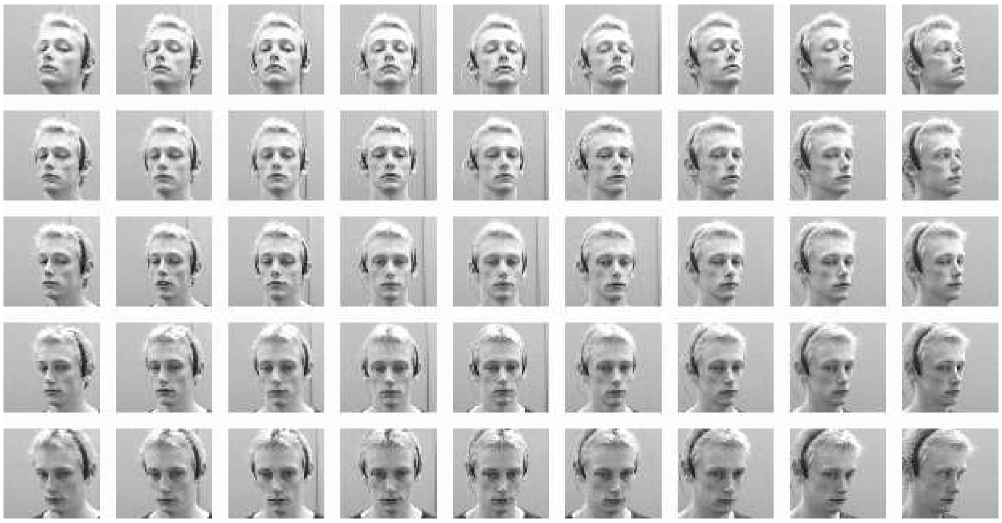
2. using KDA to extract the non-linear discriminating features of the texture patterns;

3. using *identity surfaces* to represent multi-view face patterns, where the discriminating features are indexed by pose information.

The first issue has been discussed in Section 5.2.3. We will discuss the second issue in the following part of this chapter. The last issue will be addressed in the next chapter with the problem of video-based dynamic face recognition.

6.3.1 Variation from Subjects and Variation from Pose

The patterns used for face recognition are represented by *shape-and-pose-free* texture patterns, which are extracted by fitting the multi-view dynamic face model described in Chapter 5 onto multi-view face images and warping them to the model mean shape at frontal view.



(a) The original images of a face class.

(b) The warped facial texture patterns.

Figure 6.4. The original face images of a face class and the warped facial texture patterns. The pose changes in $[-20^\circ, +20^\circ]$ *in tilt and* $[-40^\circ, +40^\circ]$ *in yaw in these images. When one side of a face becomes partially invisible, the texture pattern is constructed from the other, visible side.*

Although the *shape-and-pose-free* facial texture patterns from different views may be more similar than their original forms, the underlying discriminating features for different face classes have not been represented explicitly. Therefore such a representation in itself would not be efficient for recognition. To illustrate this problem, we plot the *shape-and-pose-free* face texture patterns in the PCA space in Figure 6.5. For the sake of conciseness, only the patterns of four face classes are shown here. Figure 6.5(a) illustrates the variation of the first PCA dimension with respect to the pose change. The horizontal axis gives the index number of images in the order of $-20^\circ \sim +20^\circ$ in tilt and $-40^\circ \sim +40^\circ$ in yaw. The orders are identical for all face classes. The patterns belonging to a same face class are linked together. Figure 6.5(b) shows the distribution of the texture patterns in the first two PCA dimension. The original face images and warped texture patterns from one of these subjects are shown in Figure 6.4.

It is noted that the variation from different face classes is not efficiently separated from that for pose change, or more precisely, the former is even overshadowed by the latter.

6.3.2 Extracting the KDA Features of Faces

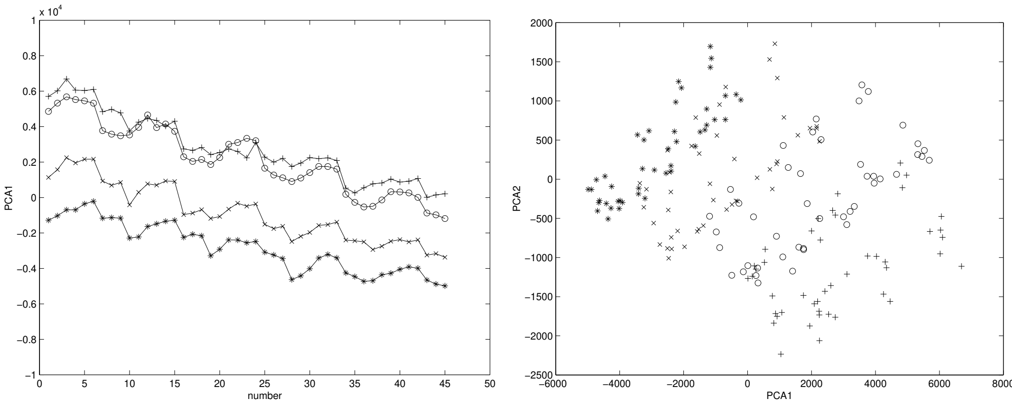
We apply KDA to the same set of data as in Figure 6.5. The Gaussian kernel is adopted,

$$k(\boldsymbol{x}, \boldsymbol{y}) = exp(-\frac{\|\boldsymbol{x} - \boldsymbol{y}\|^2}{2\sigma^2}) \tag{6.29}$$

where $2\sigma^2 = 1$.

The variation and distribution of the patterns are shown in Figure 6.6(a) and 6.6(b) respectively. Compared to the results of the PCA patterns in Figure 6.5, the improvement on class separability is significant. It is worth pointing out that such separability is achieved by using only two KDA dimensions.

We have experimented with different types of kernel functions such as polynomial, sigmoid and Gaussian kernel functions. Similar results have been obtained for different choice of kernel. Meanwhile, when the parameter of the Gaussian



(a) Variation of the 1st PCA dimension with respect to pose change.

(b) Pattern distribution in the first two PCA dimensions

Figure 6.5. Face class separability under multiple views: variation from different face classes vs. variation from pose change. The horizontal axis in (a) gives the index number of pose changing between $[-20^\circ, +20^\circ]$ *in tilt and* $[-40^\circ, +40^\circ]$ *in yaw.*

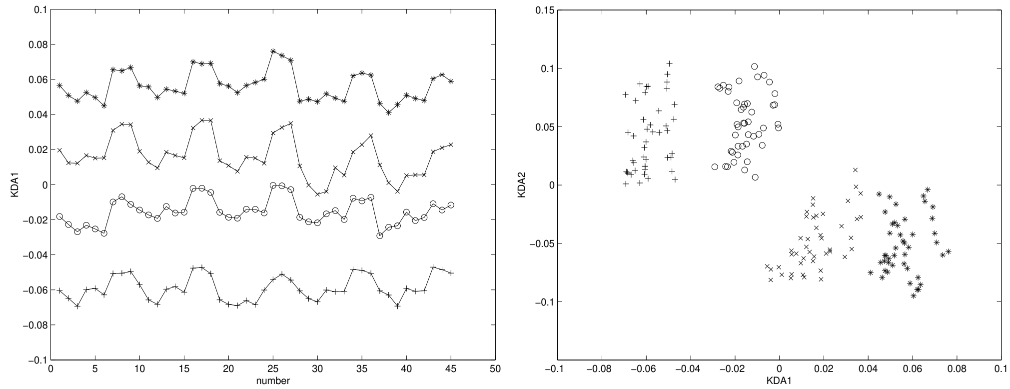


Figure 6.6. Distribution of the KDA patterns obtained from the same face images as in Figure 6.5.

kernel is chosen as $2\sigma^2 = 1$, a satisfactory result in terms of recognition accuracy and reliability (defined in Section 6.3.3) has been achieved [2].

[2]See Section 3.4.2 for a similar discussion.

6.3.3 Multi-View Face Recognition and Performance Analysis

The classical statistical method performs the task of face recognition as follows: First, estimate the probability density function of each face class. Then, for a new face pattern, evaluate it on each of the density function and assign a class label to it based on maximum likelihood. When dealing with the multi-view face recognition problem, this method is ill-suited since large number of learning examples are needed and the pattern distribution is irregular due to the severe non-linearity.

As an alternative, large margin classification, such as SVMs, seeks to model the boundaries between different face classes. However, this method also requires large number of training examples. Moreover, multi-class classification using this method is usually computationally expensive.

we do not try to solve the specific multi-view face recognition problem using a general method. Instead, a pose based approach is proposed in this work. This approach is similar to the parametric eigenspace method presented by Murase and Nayar (Murase and Nayar, 1994; Murase and Nayar, 1995). The basic idea of this approach is described as follows:

The KDA vectors of *shape-and-pose-free* facial texture patterns are adopted to represent faces. The identities of faces are learnt from a sparse sample of the face patterns of this face class. All the learning face patterns are represented as KDA feature vectors and labelled with pose which is known or can be estimated from the face images. For a novel face image, we extract its KDA feature vector and estimate its pose, and interpolate a “virtue” pattern from the known patterns of each face class using the same pose information as the face to be recognised. Face recognition can then be performed by computing the distances between the novel pattern and the “virtue” patterns. The simple Euclidean distance is adopted in this work since all patterns are indexed by the same pose information.

It is noted that this approach has been extended to the concept of *identity surfaces* which enables face recognition to be performed dynamically from video input. More details of this issue will be described in Chapter 7.

We tested the performance of PCA, KPCA, LDA and KDA on a set of multi-view face images. The data set include 540 face images from 12 subjects, 45 of each subject. The pose ranges from $-20°$ to $+20°$ in tilt and from $-40°$ to $+40°$ in yaw with an interval of $10°$. Example face patterns from one of these subject are shown in Figure 6.4. We used 180 face images, 15 of each subject, to train PCA, KPCA, LDA and KDA and to learn the identities of these subjects. The pose covered by these training face images is $\{-20°, 0°, +20°\}$ in tilt and $\{-40°, -20°, 0°, +20°, +40°\}$ in yaw. The other 360 face images were used as test patterns. In this experiment, the dimension of the PCA, KPCA, LDA and KDA feature vectors varies from 1 to 10.

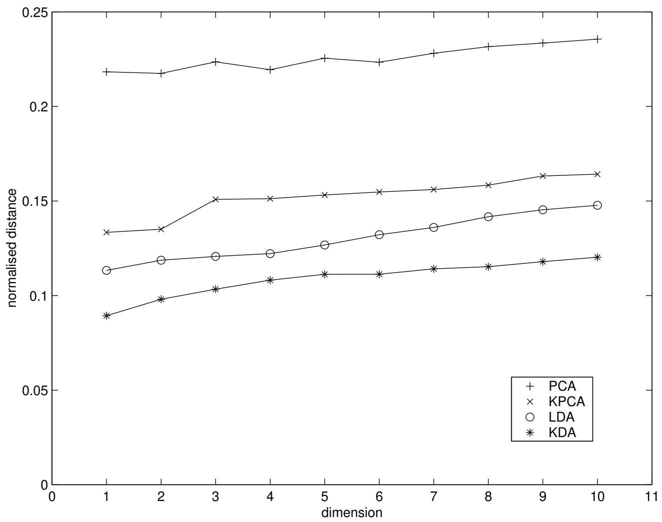


Figure 6.7. Recognition reliability. It is indicated that the reliability of recognition, from the best to the worst, is achieved with KDA, LDA, KPCA and PCA.

We define the following criterion to evaluate the reliability of different representations:

$$d' = \frac{1}{N} \sum_{i=1}^{N} \frac{C \cdot d_{i0}}{\sum_{j=1}^{C} d_{ij}} \tag{6.30}$$

where C is the number of face classes, N is the total number of test face patterns, d_{ij} is the pattern distance between the ith test pattern and the jth face class, and

dimension	PCA	KPCA	LDA	KDA
1	0.21827	0.13345	0.11329	0.08938
2	0.21750	0.13511	0.11868	0.09805
3	0.22354	0.15090	0.12072	0.10341
4	0.21942	0.15125	0.12220	0.10817
5	0.22551	0.15316	0.12672	0.11127
6	0.22339	0.15478	0.13221	0.11133
7	0.22812	0.15609	0.13601	0.11412
8	0.23163	0.15836	0.14172	0.11526
9	0.23348	0.16321	0.14536	0.11795
10	0.23559	0.16418	0.14772	0.12031

Table 6.1. Reliability of recognition using the four types of representation: PCA, KPCA, LDA and KDA. The values are computed using Equation (6.30) with respect to the dimension of the features.

d_{i0} is the pattern distance between the ith test pattern and the ground-truth face class.

Criterion d' can be interpreted as a summation of normalised pattern distances to their ground-truth face class. The smaller the d', the more reliable the classification performance. Figure 6.7 shows the values of d' for different representations, PCA, KPCA, LDA and KDA, with respect to the dimension of the feature spaces. The results indicate that KDA gives the most reliable classification performance. The values of the normalised distance d' are presented in Table 6.1.

The recognition accuracy with respect to the dimension of feature spaces is shown in Figure 6.8 and Table 6.2. It is interesting to note that the KDA features are very effective. A 93.9% recognition accuracy was achieved when the dimension of the KDA vector was only 2. It is also observed that, for the small scale problem with 12 subjects, all the method except for KPCA achieved a 100% recognition accuracy when the dimension of features is not less than 6. We will investigate how these techniques perform on large scale problems in future work.

6.4 Summary

Recognising faces with large pose variation involves a severe non-linearity caused by rotation in depth, self-occlusion, self-shading, and illumination change. Under these circumstances, extracting non-linear discriminating features which maximise the variance from different people and minimise the variance from pose change is crucial.

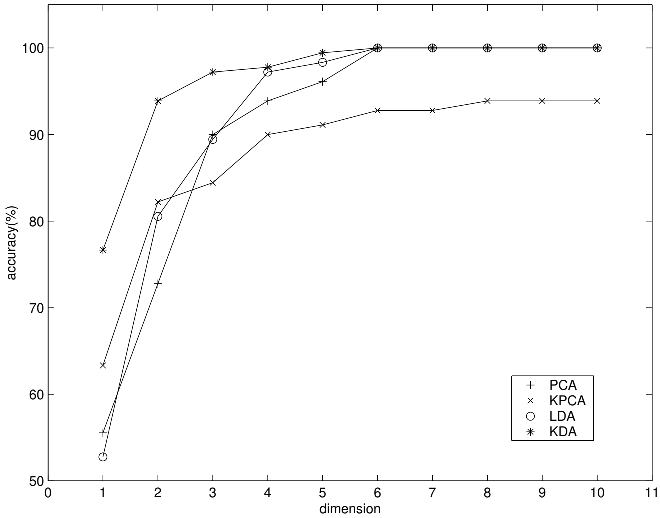


Figure 6.8. Recognition accuracy. It is indicated that the KDA features are very effective: it achieves a high recognition rate with a dimensionality as low as 2.

PCA, LDA and KPCA have been widely used in face recognition. But PCA and LDA are limited to linear applications while KPCA is designed to capture the *overall* rather than the *discriminating* variance of patterns even though it is non-linear. To efficiently extract the discriminating features of multi-class patterns with severe non-linearity, KDA is developed in this work. We applied this method to multi-view face recognition, and significant improvement has been achieved both in reliability and accuracy.

One of the main drawbacks of this approach is the intensive computation involved. To obtain the KDA projection of an unknown pattern, one has to compute the kernel functions of this pattern with all training examples. Actually

dimension	PCA(%)	KPCA(%)	LDA(%)	KDA(%)
1	55.6	63.3	52.8	76.7
2	72.8	82.2	80.6	93.9
3	90.0	84.4	89.4	97.2
4	93.9	90.0	97.2	97.8
5	96.1	91.1	98.3	99.4
6	100.0	92.8	100.0	100.0
7	100.0	92.8	100.0	100.0
8	100.0	93.9	100.0	100.0
9	100.0	93.9	100.0	100.0
10	100.0	93.9	100.0	100.0

Table 6.2. Accuracy of recognition using the four types of representation: PCA, KPCA, LDA and KDA.

this is a common drawback of all kernel techniques such as SVMs and KPCA. Although some methods such as the reduced set technique (Burges, 1996; Burges and Scholkopf, 1997) can be adopted for computation reduction, an additional non-linear optimisation problem is usually introduced which is not guaranteed to provide a global optimal solution.

Chapter 7

Video Based Face Recognition Using Identity Surfaces

So far, we have discussed the problems of face detection, pose estimation, face modelling, and feature extraction. Also, we have briefly described the issue of static face recognition by computing a distance or a similarity between face patterns (Section 6.3.3). However, face recognition is more than static image matching. The dynamic characteristics of a moving face can provide far enriched information for the task of face recognition. In this chapter, we will focus on the issue of dynamic face recognition from video input.

7.1 Background

In previous research, most the existing methods are for frontal-view or near frontal-view face recognition. However, recognising faces with large pose variation is more challenging. A widely adopted approach to address this problem is to develop view-based multiple face models, for example, the view-based and modular eigenspaces approach (Moghaddam and Pentland, 1994; Moghaddam and Pentland, 1997), the view-based piece-wise SVM model (Ng and Gong, 1999b; Ng and Gong, 1999a), and the view-based Active Appearance Models (Cootes et al., 2000). But the division of the face space in these methods is rather arbitrary, ad hoc and often coarse.

Another limitation of the previous work is that the basic methodology adopted for recognition is largely based on matching static face images. Psychology and physiology research indicates that the human vision system's ability to recognise animated faces is better than that on randomly ordered still face images (i.e. the same set of images, but displayed in random order without the temporal context of moving faces). Knight and Johnston (Knight and Johnston, 1997) showed that recognition of famous faces in photographic negatives can be significantly enhanced when the faces were shown moving rather than static. Bruce *et al.* (Bruce et al., 1998b) extended this result to other conditions where recognition is made difficult, e.g. by thresholding the images or showing them in blurred or pixellated formats. Bassili (Bassili, 1979) set up an experiment for expression recognition using the movement of a sparse spatial arrangement of white dots (representing facial features) on a black face surface. The results indicated that a superior accuracy is achieved over that on static images even though only a small number of features are employed. Davis and Bobick (Davis and Bobick, 1997) demonstrated that people can trivially recognise actions even in extremely blurred image sequences where almost no structure is presented in each individual frame.

For computer vision systems, although some preliminary results from approaches such as the temporal signature method (Gong et al., 1994) , the online class-specific correction method (Edwards et al., 1998c) and the subspace method (Yamaguchi et al., 1998), have been reported, the issue of interpreting the dynamics of faces under a spatio-temporal context remains largely unresolved.

Aiming to address these two challenging problems in face recognition, i.e. recognising faces with large pose variation and recognising faces dynamically from video input, we present in this chapter an approach to dynamic face recognition using *identity surfaces*.

An *identity surface* is constructed from the discriminating features of a face class based on pose information. Therefore it is appropriate to deal with the variation from pose change. Moreover, it enables face recognition to be performed dynamically over time. By tracking a moving face from a video input and extracting the discriminating features for this face, one obtains an object trajectory in a discriminating feature space. Meanwhile, a set of model trajectories can be constructed on the *identity surfaces*, one of each face class, using the same pose information and temporal order. Face recognition can then be performed dynamically by matching these two kinds of trajectories.

In the rest part of the chapter, the basic idea of using *identity surfaces* for multi-view face recognition, their construction algorithm, and the approach to construct them from video sequences are introduced in Section 7.2. The approach to video based face recognition using *identity surfaces*, including object and model trajectory construction, pattern distance computing and trajectory matching, is presented in Section 7.3. Section 7.4 is a summary of this chapter.

7.2 Identity Surfaces

One of the most commonly used techniques for recognition is to compute the probabilities of a set of known patterns or the similarities among templates of different classes before selecting the optimal value using a simple metric. For example, the Euclidean distance or the Mahalanobis distance can be adopted if the pattern distribution of each class is compact enough and separable from others. However, usually this simplistic method cannot provide satisfactory solutions to the problem of multi-view face recognition. The reasons are twofold: First, the representation adopted, e.g. the KDA, may not generate a *perfectly* compact distribution of each face class while separating one from another. Second, the distributions of each class cannot be guaranteed to be homogeneous.

When the distribution is irregular, the traditional statistical method for dealing with this problem is to estimate a multi-modal density function for each class. But a very large number of training examples are needed either for parametric or non-parametric modelling. In this work, we do not constrain ourselves to such a strict condition. Instead, we present a novel approach to construct an *identity surface* for each face class from a sparse sample of multi-view face patterns.

As stated previously, one of the key problems of multi-view face recognition is how to separate two kinds of variations: variation from different subjects and variation from pose. Observing the results presented in Figure 6.5(a) and Figure 6.6(a), we find that the features from different face classes share a similar varying tendency with respect to pose change. It suggests that a significant improvement to face identity modelling can be expected if the pose information is exploited explicitly. Based on this idea, we developed a method of multi-view face recognition using *identity surfaces*. The basic idea of the *identity surfaces* is similar to the parametric eigenspace method presented by Murase and Nayar (Murase and Nayar, 1994; Murase and Nayar, 1995).

Assuming that only the appearance variation caused by *rotation in depth* is concerned, i.e. the variation from expression, illumination and facial make-up is excluded, each face class can be represented by a unique hyper surface based on pose information. In other words, the two basis coordinates stand for the head pose: tilt and yaw, and the other coordinates are used to represent the discriminating features of faces, e.g. the KDA vectors. For each pair of tilt and yaw values, there is one unique "point" for a face class. The distribution of all the "points" of the same face class with regard to pose change form a hyper surface in the space spanned by the discriminating features and pose. We call this surface an *identity surface*. Face recognition can then be performed by computing and comparing the distances between a given pattern and a set of *identity surfaces*.

7.2.1 Construction Algorithm

If sufficient patterns of a face class in different views are available, the *identity surface* of this face class can be constructed precisely. However, we do not require such a strict condition. In this work, we develop a method to synthesise the *identity surface* of a face class from a small sample of face patterns which sparsely cover the view sphere.

The basic idea is to approximate the *identity surface* using a set of N_p planes separated by a number of N_v predefined views. The problem can be formally defined as follows:

Suppose x, y are tilt and yaw respectively, $\boldsymbol{z}$ is the discriminating feature vector of a face pattern, e.g. a KDA vector. A list $(x_{01}, y_{01}), (x_{02}, y_{02}), ..., (x_{0N_v}, y_{0N_v})$ gives predefined views which discretise the view sphere into N_p grids. On each grid, the *identity surface* of a face class is approximated by a plane

$$\boldsymbol{z} = \boldsymbol{a}x + \boldsymbol{b}y + \boldsymbol{c} \tag{7.1}$$

Suppose the M_i sample patterns covered by the ith plane are $(x_{i1}, y_{i1}, \boldsymbol{z}_{i1}), (x_{i2}, y_{i2}, \boldsymbol{z}_{i2}), ..., (x_{iM_i}, y_{iM_i}, \boldsymbol{z}_{iM_i})$, then one minimises

$$\begin{aligned} \mathcal{Q} &= \sum_{i}^{N_p} \sum_{m}^{M_i} \|\boldsymbol{a}_i x_{im} + \boldsymbol{b}_i y_{im} + \boldsymbol{c}_i - \boldsymbol{z}_{im}\|^2 && (7.2) \\ \text{subject to} &: \boldsymbol{a}_i x_{0k} + \boldsymbol{b}_i y_{0k} + \boldsymbol{c}_i = \boldsymbol{a}_j x_{0k} + \boldsymbol{b}_j y_{0k} + \boldsymbol{c}_j \\ & \quad k = 0, 1, ..., N_v, \\ & \quad \text{plane } i, j \text{ intersect at } (x_{0k}, y_{0k}). && (7.3) \end{aligned}$$

This is a Quadratic Programming problem which can be solved using the interior point method (Vanderbei, 1994).

Figure 7.1 shows a real identity surface of a face class using 45 example views ($-20^\circ \sim +20^\circ$ in tilt and $-40^\circ \sim +40^\circ$ in yaw with an interval of 10°) and the synthesised identity surface using only 15 example views, i.e. the same pose ranges but with an interval of 20° .

7.2.2 Learning Identity Surfaces from Example Sequences

Before recognition is carried out, a face class should be registered to a system, i.e. let the system learn the identity of the face class. The face class registration can be conducted through one or more example sequences containing the faces patterns of this class. For example, we can record a small video clip of a subject while he/she rotates the head in front of a camera. After applying the multi-view dynamic face model described in Chapter 5 on the video sequence, we obtain a

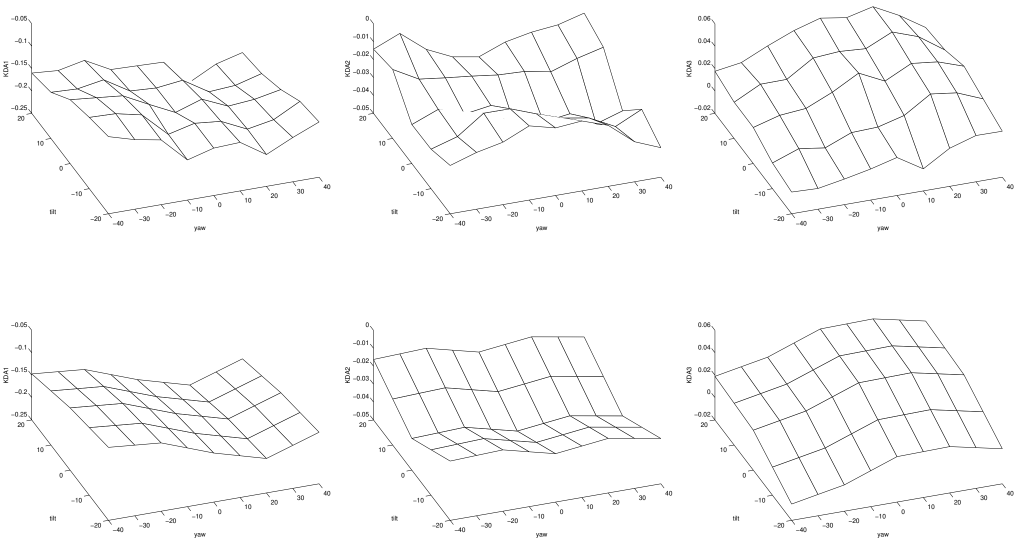


Figure 7.1. The identity surface constructed from all 45 views (first row) and that synthesised from 15 prototype patterns (second row). Only the first three KDA components are shown here.

set of face patterns of this subject. These patterns are then stored to construct the *identity surface* of this face class, and, if necessary, to train (or re-train) the KDA.

To simplify the computation, normally we do not use all the patterns of each subject to train the KDA since the sizes of the kernel matrix $\boldsymbol{K}$ and $\boldsymbol{K}_c$ are directly related to the number of training examples. A pragmatic approach to selecting the KDA training patterns is to factor-sample the patterns from the training sequences so that the result patterns uniformly cover the view sphere.

After KDA training, all face patterns are projected onto the feature space spanned by the significant KDA base vectors. Then the method described in Section 7.2.1 is employed to construct the *identity surfaces*.

7.3 Recognising Faces Dynamically from Video

Even for the human vision system, the performance of face recognition is not very reliable on *static images*. However, the situation can be considerably improved when video input is available where faces move continuously. Recall the discussions in Section 7.1, psychological and physiological research suggests that modelling and recognising moving faces dynamically have the potential of achieving a superior performance over that on static images.

7.3.1 Video-Based Online Face Recognition

We argue that the performance of face recognition for a computer based vision system can be significantly enhanced if the facial dynamics is modelled as follows:

1. Instead of exhaustively scanning an image, which is notoriously slow, *selective attention* can be performed effectively using motion, colour, and background information.

2. Information from individual frames of a video input may be ambiguous, or incomplete. However, the accumulated evidence from all frames can provide a more reliable performance.

3. It is interesting to note that the human vision system works in an *interactive* or *close-looped* rather than an *open-looped* manner. For example, when observing a moving face, we predict the next likely position, pose, and appearance of the face as well as collecting the information at the time being. The coincidence or difference between our prediction and observation allow us to adjust the perception we have of a face. Thus a reinforced effect is achieved in this *interactive* manner. As for computer based vision systems, an improved performance can be achieved if the model parameters, or even the model itself, is adapted to the observations and measurements dynamically.

7.3.2 Recognising Faces Dynamically Using Identity Surfaces

The simplest method of using *identity surfaces* for face recognition is to compute the pattern distances to the *identity surfaces.* This method gives the frame-by-frame recognition results from a video input containing faces. However, more reliable and accurate recognition can be achieved by matching the object trajectory tracked from a video input with a set of model trajectories constructed from the known *identity surfaces* using the same pose information and temporal order.

For computer based vision systems, the issue of formulating and modelling the facial dynamics is non-trivial and still largely under-developed. However, significant improvement in terms of recognition accuracy and reliability may still be achieved when the spatio-temporal information is modelled in a rather straightforward way, e.g. simply accumulating the discriminating evidences with the spatio-temporal order encoded in an input sequence. As a practical implementation, we formulate the following approach to video-based online face recognition using *identity surfaces.*

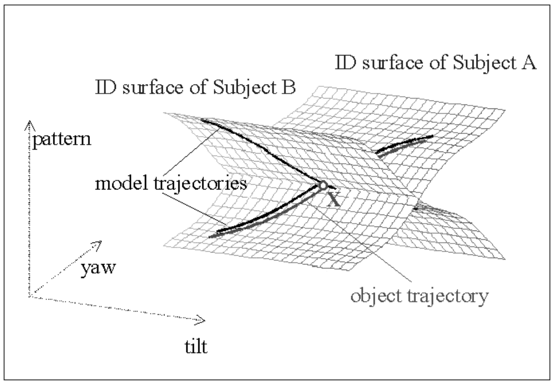


Figure 7.2. Identity surfaces for face recognition

As shown in Figure 7.2, when a face is detected and tracked in an input video sequence, one obtains the *object trajectory* of the face in the feature space. Also, its projection onto each of the *identity surface* with the same poses and temporal order forms a *model trajectory* of the specific face class. It can be regarded as the ideal trajectory of this face class encoded by the same spatio-temporal information (pose information and temporal order from the video sequence) as the tracked face. Then face recognition can be carried out by matching the object trajectory with a set of model trajectories. Compared to face recognition on static images, this approach can be more reliable and accurate. For example, it is difficult to decide whether the pattern X in Figure 7.2 belongs to subject A or B for a single pattern, however, if we know that X is tracked along the object trajectory, it is more likely to be subject A than B.

The complete process of our video-based face recognition includes:

Registration Construct the *identity surface* for each face class from one or more training sequences;

Tracking Fit the multi-view dynamic model (Chapter 5) on an input video sequence containing faces to be recognised, and extract the discriminating features;

Recognition Compute the object and model trajectories and match these trajectories.

The method described in Section 7.2 is used for registration. Also, we have discussed a Kalman filter based face tracking method using the multi-view dynamic face model in Section 5.4. In this chapter, we mainly discuss the issue of multi-view dynamic face recognition.

7.3.3 Pattern Distances to the *Identity Surfaces*

For an unknown face image, one first fits the multi-view dynamical face model (Chapter 5) onto the image and projects the extracted face pattern into the KDA feature space to yield a pose labelled feature vector $(x, y, \boldsymbol{z}_0)$ where $\boldsymbol{z}_0$ is the KDA vector and x, y are the pose in tilt and yaw. Then the pattern distance to one of the *identity surfaces* can be computed as the Euclidean distance between $\boldsymbol{z}_0$ and the corresponding point $\boldsymbol{z}$ on the *identity surface*

$$d = \|\boldsymbol{z}_0 - \boldsymbol{z}\| \tag{7.4}$$

where $\boldsymbol{z}$ is given by (7.1).

7.3.4 Trajectory Matching

When a face is tracked in an input video sequence, an object trajectory can be obtained by projecting the face patterns into the KDA feature space. Furthermore, a model trajectory can be built on the *identity surface* of each subject using the same pose information and temporal order of the object trajectory. Those two kinds of trajectories, given any sequence of specific poses in a temporal order, encode the spatio-temporal information of the tracked face. And finally, recognition is performed dynamically by matching the object trajectory to a set of identity model trajectories.

A preliminary realisation of this approach is implemented by computing a trajectory distance

$$d_m = \sum_{i=1}^{t} w_i d_{mi} \tag{7.5}$$

where d_{mi} is the pattern distance to the *identity surface* of the mth face class in the ith frame computed using (7.4), and w_i is the weight of this distance. Recognition is performed by selecting the subject with minimum trajectory distance.

7.3.5 Experiments

We applied this approach to a small scale multi-view face recognition problem. Twelve sequences, each from a set of 12 subjects, were used as training sequences to construct the *identity surfaces*. The number of frames contained in each sequence varies from 40 to 140. We randomly selected 180 images (15 images of each subject) to train the KDA. The first ten KDA basis vectors were used to construct the *identity surfaces*. Then recognition was performed on new test sequences of these subjects.

Figure 7.3 shows the results on one of the test sequences. It is noted that a more reliable performance is achieved when recognition is carried out using the trajectory distances which include the accumulated evidence over time, although the pattern distances in each individual frame already provides good recognition accuracy on a frame by frame basis.

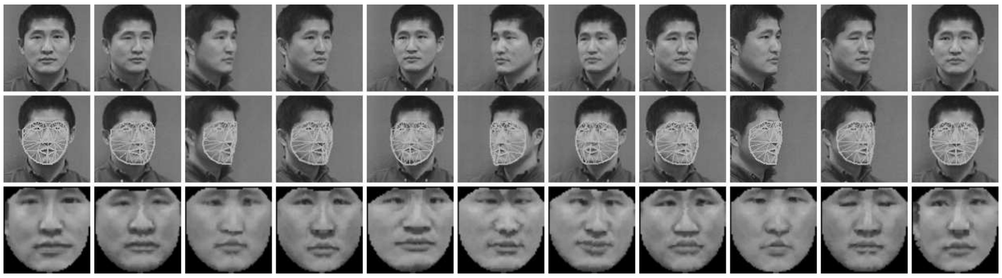
Figure 7.4 shows the results on another sequence where the face is undergoing significant expression change. Since all the training face images are taken in neutral expression, the results of model fitting is not as good as those in Figure 7.3. Also, the pattern distance from an individual frame gives the wrong recognition result in a few frames. However, it is important to point out that the trajectory distance still provides a reliable and accurate recognition.

7.4 Summary

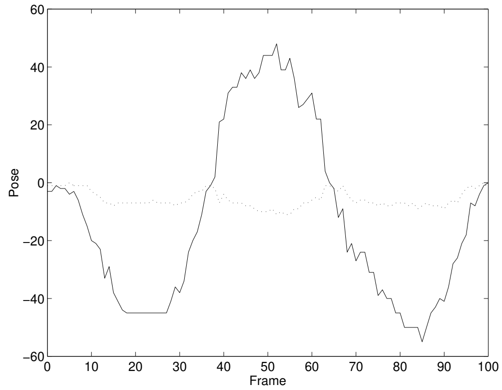
Recognising faces across views is more challenging than that from a fixed view because of the severe non-linearity caused by rotation in depth, self-occlusion, self-shading, and illumination change. To model the variance from rotation in depth, we proposed a method of *identity surface* which can be constructed from a sparse sample of multi-view face images. Then recognition can be performed by computing the pattern distances or trajectory distances to a set of *identity surfaces*.

Psychological and physiological research suggests that modelling and recognising moving faces dynamically have the potential of achieving a superior performance over that on static images. Inspired by this idea, we present an approach to dynamic face recognition by computing and matching the object and model trajectories. A more reliable recognition is achieved since these trajectories encode the spatio-temporal information of a moving face and provide the accumulated evidence of identity.

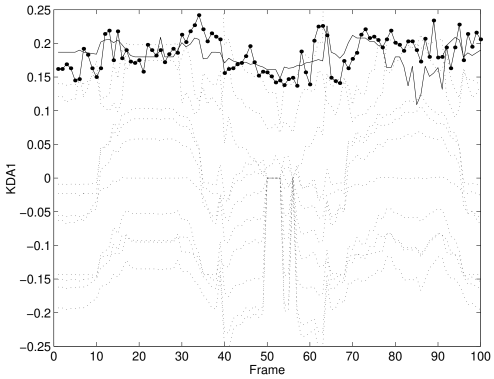
For visual interaction and human-computer interface, the problem of face recognition involves more than matching static images. At a low-level, the face



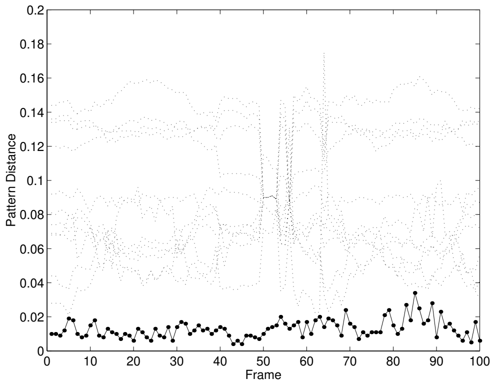
(a) Sample frames with an interval of 10 frames, fitted 3D shape patterns, and the *shape-and-pose-free* texture patterns.



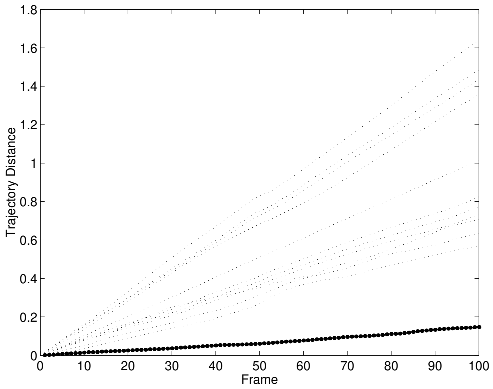
(b) Pose in tilt (dotted) and yaw (solid).



(c) Object and model trajectories.

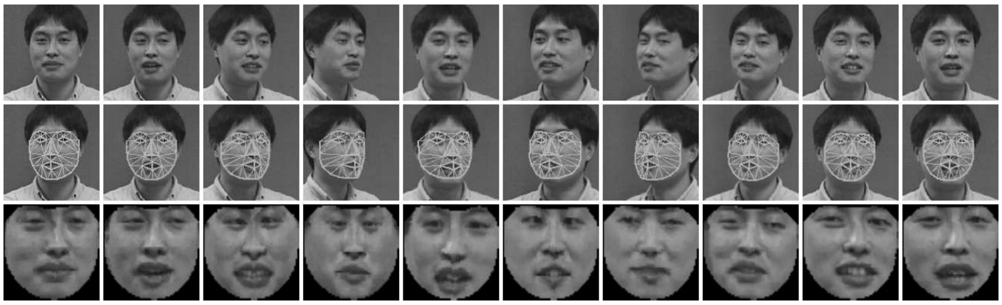


(d) Pattern distances.

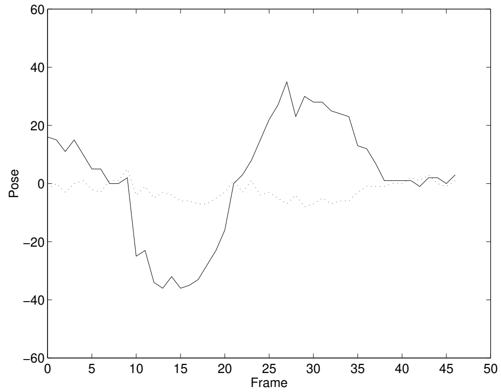


(e) Trajectory distances.

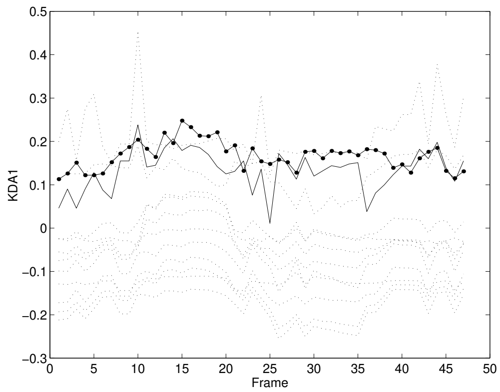
Figure 7.3. Video-based multi-view face recognition. (c) shows the object trajectory (solid line with dots) and model trajectories in the first KDA dimension where the model trajectory from the ground-truth subject is highlighted with solid line. It is noted from (d) and (e) that the pattern distances can give an accurate recognition result; however, the trajectory distances provide a more reliable performance, especially its accumulated effects (i.e. discriminating ability) over time.



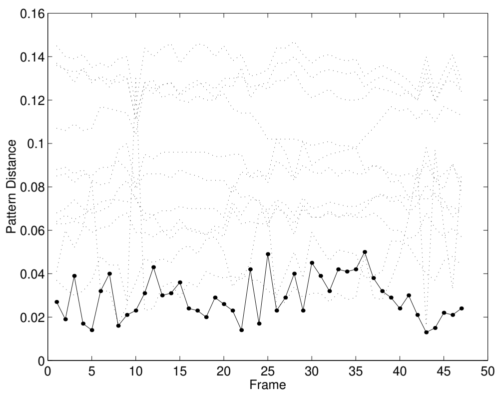
(a) Sample frames with an interval of 5 frames, fitted 3D shape patterns, and warped texture patterns.



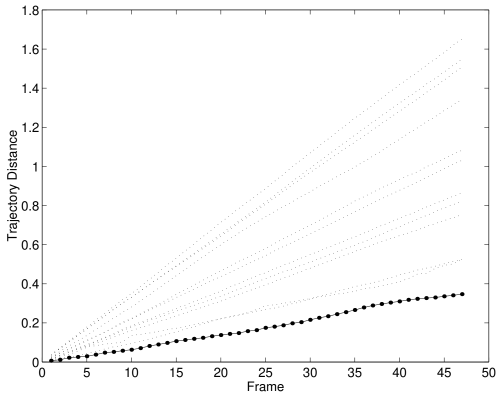
(b) Pose in tilt (dotted) and yaw (solid).



(c) Object and model trajectories.



(d) Pattern distances.



(e) Trajectory distances.

Figure 7.4. Face recognition on a face sequence with significant expression change. The pattern distance is less reliable for a few frames, however, the trajectory distance still provides a reliable and accurate recognition.

dynamics should be accommodated in a consistent spatio-temporal context where the underlying variations with respect to changes in identity, view, scale, position, illumination, and occlusion are integrated together. At a higher level, more sophisticated behaviour models, including individual-dependent and individual-independent models, may supervise and co-operate with all the low level modules.

In this chapter, we highlighted the nature of the problem and showed the potential of modelling face dynamics as an effective means to face recognition. However, some of the implementation such as trajectory matching is still simplistic in its present form. The temporal information has been intensively used in tracking faces and constructing the object and model trajectories. However, when matching these trajectories, we do not explicitly use the temporal information - the trajectory distance is simply computed as a weighted summation of the pattern distances in individual frames. The underlying mechanism of this spatio-temporal dynamics remains to be an interesting issue for both psychological and computational vision research. Extensive further work is still needed.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

The human face provides a large variety of information, such as identity, gender, emotion, interest, and intention, in our social life. Computer based face recognition has been emerging as an active research area over the past decade with a number of potential applications such as authentication, visual surveillance, video conferencing, and human-computer interaction.

In this thesis, we have presented a comprehensive approach to face modelling, detection, tracking and recognition. In particular, three challenging problems have been emphasised:

1. **Modelling faces with large pose variation**

   This problem is more challenging than that from a fixed view, e.g. frontal view or near frontal view, due to the severe non-linearity caused by rotation in depth, self-shading and self-occlusion.

2. **Extracting the non-linear discriminating features for face recognition**

   A good representation for recognition should provide the significantly discriminating features which maximise the variation from identities and minimise that from other sources. For multi-view face patterns, the typical linear techniques such as PCA and LDA are ineffective.

3. **Recognising the dynamics of faces**
   Recognising a face is more than static image matching. When a moving face is tracked in an image sequence continuously, the spatial and, in particular, temporal information of the face provides far enriched clue about the emotion, intention, interest and identity of the subject.

8.1.1 Multi-View Dynamic Face Model

A multi-view dynamic face model has been presented in this thesis. It consists of a 3D Point Distribution Model where the 3D shape vectors of faces are constructed from 2D images with labelled pose and sparse landmarks, a *shape-and-pose-free* texture model which is built from patterns warped to the mean shape in the frontal view, and an affine geometrical model which provides the position, scale, and pose of a face.

One of the significant characteristics of the model is that it is capable of dealing with faces with large pose variation. Experimental results showed a robust tracking is achieved between $[-70°, +70°]$ in yaw. The reasons for this capability include the 3D shape model and the view-specific fitting criteria defined in the local and temporal terms of the fitting function.

By fitting the model to a face image or an image sequence, the identity parameters and the geometrical parameters are obtained. The former are crucial to face recognition, and the latter are important to face tracking. A Kalman filter based scheme is designed to provide a temporal estimation of model parameters.

8.1.2 Multi-View Face Detection

For a computer based system, face detection is usually regarded as a necessary process before face recognition. Although this process can be simplified when continuous video input is available, e.g. using the information in the previous time step, it is still important in the initialisation stage when no prior knowledge about a face to be tracked is available, or when recovery is needed from tracking failure.

To address the irregular variation caused by large view change, the view sphere is segmented into several pieces. On each of these pieces, a face detector is constructed. A pose estimation based scheme is developed for searching a face from an image, i.e. the pose of the image patch from a search window is estimated first, then one of the face detectors is selected using the pose information. Computational efficiency is achieved since only one face detector needs to be computed.

Support Vector Regression has been successfully applied in this work for pose estimation. A hybrid method of eigenface and SVM has been proposed for multi-view face detection. Compared with the eigenface method and SVM method, this method achieves the best balance between accuracy and speed: it is almost as accurate as the SVM method, and as fast as the eigenface method.

8.1.3 Kernel Discriminant Analysis

There are many sources of variation involved in the problem of face recognition, for example, variation from identities, expression, illumination, and pose. An ideal representation of the problem should isolate these different kinds of variation from each other. In practice, we usually seek to maximise the variation from identities and minimise that from others.

Linear techniques such as LDA have been widely used in many pattern recognition problems including face recognition. However, when significant non-linearity exists, for example, in the multi-view face recognition problem, these techniques are ill-suited. To extract the non-linear discriminating features, we developed a non-linear method, Kernel Discriminant Analysis.

The kernel technique is adopted in this method. Assuming the patterns with non-linear distribution can be linearly separable in a high-dimensional feature space, one performs a LDA in the feature space to equivalently express the non-linear discriminating characteristics in the original input space. Then by using a kernel function, all computation can be performed conveniently in the input space.

This method has been applied to multi-view face recognition. Experimental results showed that a superior performance has been achieved over other methods such as PCA, LDA and KPCA.

8.1.4 Video Based Face Recognition Using Identity Surfaces

Psychological and physiological research has shown that the dynamic information of an animated face can significantly improve the performance of face recognition although the underlying mechanism is still unclear.

The concept of *identity surfaces* has been proposed in this thesis. An *identity surface* is constructed in a discriminating feature space as a unique description of a face class. It provides a test-bed for dynamic face recognition from video input. In other words, an object trajectory is obtained when a face is tracked continuously over time. At the same time, a set of model trajectories, each for a face class in the database, can be synthesised on the *identity surfaces* using the same pose information and temporal order as the object trajectory. Then face recognition can be performed dynamically by matching the object trajectory with the model trajectories.

Preliminary experiments demonstrated the promise of this approach. More robust and accurate results have been achieved by this method than those of static image matching.

8.2 Future Work

So far the issue of constructing a multi-view face model and applying this model to dynamic face recognition has been intensively discussed. Despite its promise which has been shown in this thesis, this work still has some limitations. In this section, we discuss these limitations and possible directions for future research.

8.2.1 Fitting Algorithm

The performance bottle-neck of the approach is the model fitting algorithm (Section 5.3) since it is based on a stochastic searching. We have tried to use some fast methods, for example, the linear fitting algorithm in AAMs (Cootes et al., 1998). However, these efforts failed for multi-view face images, especially when large pose variation is involved. An improvement in fitting efficiency may be achieved if the following strategies are adopted:

1. Use the image property, e.g. edge, valley, colour, and histogram of intensity, to constrain the shape parameters of the model.

2. Develop a coarse-to-fine fitting scheme, not only in image resolution, which has been adopted in many previous studies, but also to the complexity of the model itself. For example, start fitting with a simple model with small degree of freedom, then go for more complex ones.

8.2.2 Sparse Representation of KDA

The effect of using KDA to extract the non-linear discriminating features for multi-view face recognition has been demonstrated in this thesis. However, one drawback of KDA is that its dimension is equal to the number of training examples. In other words, to obtain the KDA projection of a new pattern, one has to compute the kernel function of this pattern with all training examples. KDA is not endowed with the sparsity property, which is usually an advantage of kernel based method, for example, in SVM, normally only a small proportion of training examples are taken as Support Vectors.

A mechanism which is able to provide a sparse representation needs to be added to KDA for computational efficiency. Although some methods such as the reduced set techniques (Burges, 1996; Burges and Scholkopf, 1997) have been proposed, an additional non-linear optimisation problem is normally introduced which is not guaranteed to converge to the global solution. A more elaborate method is yet to be developed for this purpose.

8.2.3 Modelling Face Dynamics

We believe that characterising the dynamics of face action, which includes the global rigid movement of the overall face, the local non-rigid movement of the facial features, and the dynamical configuration of both, is the most promising issue in face recognition.

However, the methods developed in this work, such as the *identity surfaces* and trajectory matching, are still simplistic in their present form. More elaborated methods for modelling the characteristic patterns of expressive facial gestures are necessary in further research.

Appendix A

Principal Component Analysis

Principal Component Analysis (PCA) seeks to compute the optimal linear transform with reduced dimensionality in the sense of least mean squared reconstruction error. This problem is usually solved by eigen-decomposing the covariance matrix constructed by a set of training patterns. A low-dimensional space is spanned by the first few significant orthogonal eigen-vectors, i.e. principal components. Then any pattern in the original space can be approximated by a linear combination of these eigen-vectors.

This technique has been widely applied in multivariate analysis and pattern recognition for dimension reduction and computation simplification.

A.1 Algorithm

Given a set of training patterns

$$\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\} \tag{A.1}$$

where $\mathbf{x}_i \in R^d$ and $i = 1, 2, \cdots, N$, the mean vector of these patterns is given by

$$\mu = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i \tag{A.2}$$

and the covariance matrix

$$\mathbf{\Sigma} = \frac{1}{N}\sum_{i=1} N(\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T \tag{A.3}$$

Find the eigenvectors $\mathbf{v}_i$ of $\mathbf{\Sigma}$ and their eigenvalues λ_i, which satisfy

$$\mathbf{\Sigma}\mathbf{v}_i = \lambda_i \mathbf{v}_i \tag{A.4}$$

Suppose $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_p$ are the first p eigenvectors with the largest eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_p$. Those p eigenvectors are mutually orthogonal and they span a p-dimensional subspace, which is called the principal subspace.

Construct matrix $\mathbf{U}$ using the first p eigenvectors

$$\mathbf{U} = [\mathbf{v}_1 \mathbf{v}_2 \cdots \mathbf{v}_p], \tag{A.5}$$

then the projection of a new data $\mathbf{x} \in R^d$ in the principal subspace is given by

$$\mathbf{z} = \mathbf{U}^{\mathrm{T}}(\mathbf{x} - \mu) \tag{A.6}$$

On the other side, the original vector x can be reconstructed from z by

$$\mathbf{x}' = \sum_{i=1}^{p} z_i \mathbf{v}_i + \mu \tag{A.7}$$

Note that the dimension of vector $\mathbf{z}$ is p, which is usually much smaller than d. The main characteristics of PCA is that it provides the optimal linear transform with least mean squared reconstruction error, i.e. E receives its minimal value over the training set where

$$E = \frac{1}{2} \sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{x}'_i\|^2 \tag{A.8}$$

A.2 Dimension Selection

The percentage of variance of the first p principal components over the training patterns can be estimated by

$$\eta = \frac{\sum_{i=1}^{p} \lambda_i}{\sum_{i=1}^{M} \lambda_i} \tag{A.9}$$

where M is the order of $\mathbf{\Sigma}$. This equation is very useful in practice for selecting the appropriate dimension of PCA patterns.

A.3 Nonlinear PCA

PCA is a very efficient technique when the patterns of a problem can be approximated with a unimodal distribution. However, it is a linear technique in nature. When dealing with problems with irregular pattern distribution, it may be difficult for the linear PCA to provide a satisfactory solution. It is interesting to note that the research on nonlinear PCA has been receiving more and more attention. Various extensions of the linear PCA have been proposed, for example, Probabilistic PCA (PPCA) (Tipping and Bishop, 1999), Sensible PCA (SPCA) (Roweis, 1998) and Kernel PCA (KPCA) (Scholkopf et al., 1998b; Scholkopf et al., 1998c; Scholkopf et al., 1996; Scholkopf et al., 1997).

Appendix B

Kernel Principal Component Analysis

When the patterns of a given problem cannot be approximated with a unimodal distribution, or the distribution is non-linear, the linear PCA may be inefficient to represent the problem. Based on the kernel method, Scholkopf *et al.* (Scholkopf et al., 1998b; Scholkopf et al., 1998c; Scholkopf et al., 1996; Scholkopf et al., 1997) developed the Kernel Principal Component Analysis (KPCA) to compute the most significant non-linear variance from a set of training patterns with severe non-linearity. The basic idea is that, by mapping the data using a non-linear map from the original input space to a high-dimensional feature space, one obtains a unimodal distribution in the feature space so that the linear PCA can be performed. However, the computation in the feature space may be problematic due to the high dimensionality, or even impossible. By employing a kernel function which is corresponding to the non-linear map, one can carry out all the computation conveniently in the input space to achieve an equivalent solution of the problem.

Suppose the training patterns in the input space $\boldsymbol{R}^N$ are $\{\boldsymbol{x}_k, k = 1, ..., l\}$. $\boldsymbol{\phi}$ is the non-linear map defined from the input space to a high-dimensional feature space

$$\boldsymbol{\phi} : \boldsymbol{R}^N \rightarrow \boldsymbol{F} \tag{B.1}$$

B.1 Centred Data

Assume for the moment that the data mapped into the feature space are centred, i.e.

$$\sum_{k=1}^{l} \boldsymbol{\phi}(\boldsymbol{x}_k) = 0 \tag{B.2}$$

To compute the non-linear principal components of the training data, one eigen-decomposes the covariance matrix in the feature space

$$\boldsymbol{C} = \frac{1}{l} \sum_{i=1}^{l} \boldsymbol{\phi}(\boldsymbol{x}_i)\boldsymbol{\phi}(\boldsymbol{x}_i)^{\mathtt{T}} \tag{B.3}$$

i.e. one finds the eigenvalues $\lambda \geq 0$ and eigenvectors $\boldsymbol{v} \in \boldsymbol{F} \setminus \{0\}$ satisfying

$$\lambda \boldsymbol{v} = \boldsymbol{C}\boldsymbol{v} \tag{B.4}$$

Taking inner-product with vector $\boldsymbol{\phi}(\boldsymbol{x}_k)$ on both sides of (B.4) yields

$$\lambda(\boldsymbol{\phi}(\boldsymbol{x}_k) \cdot \boldsymbol{v}) = (\boldsymbol{\phi}(\boldsymbol{x}_k) \cdot \boldsymbol{C}\boldsymbol{v}) \tag{B.5}$$

Note that all solutions $\boldsymbol{v}$ lie in the span of $\boldsymbol{\phi}(\boldsymbol{x}_1), \boldsymbol{\phi}(\boldsymbol{x}_2), ..., \boldsymbol{\phi}(\boldsymbol{x}_l)$, i.e. there exist coefficient vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_l)^{\mathtt{T}}$ such that

$$\boldsymbol{v} = \sum_{i=1}^{l} \alpha_i \boldsymbol{\phi}(\boldsymbol{x}_i) \tag{B.6}$$

Substituting (B.3) and (B.6) into (B.5), and defining an $l \times l$ matrix $\boldsymbol{K}$ by

$$\boldsymbol{K}_{ij} := (\boldsymbol{\phi}(\boldsymbol{x}_i) \cdot \boldsymbol{\phi}(\boldsymbol{x}_i)), \tag{B.7}$$

one arrives at

$$l\lambda \boldsymbol{K}\boldsymbol{\alpha} = \boldsymbol{K}^2\boldsymbol{\alpha} \tag{B.8}$$

To find solutions of (B.8), one solves the problem

$$l\lambda\boldsymbol{\alpha} = \boldsymbol{K}\boldsymbol{\alpha} \tag{B.9}$$

It is important to note that by introducing kernel function

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{\phi}(\boldsymbol{x}_i) \cdot \boldsymbol{\phi}(\boldsymbol{x}_i)) \tag{B.10}$$

one does not need to compute the non-linear map $\boldsymbol{\phi}$ explicitly. Instead all the computation can be carried out conveniently in the input space through the kernel function.

For a new pattern $\boldsymbol{x}$, the projection of its image in the feature space onto the eigenvector $\boldsymbol{v}$ can be computed by

$$(\boldsymbol{v} \cdot \boldsymbol{\phi}(\boldsymbol{x})) = \sum_{i=1}^{l} \alpha_i (\boldsymbol{\phi}(\boldsymbol{x}_i) \cdot \boldsymbol{\phi}(\boldsymbol{x})) \tag{B.11}$$

Constructing the eigen matrix

$$\boldsymbol{U} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_M] \tag{B.12}$$

from the first M significant eigenvectors, and computing

$$\boldsymbol{k}_x = (k(\boldsymbol{x}, \boldsymbol{x}_1), k(\boldsymbol{x}, \boldsymbol{x}_2), ..., k(\boldsymbol{x}, \boldsymbol{x}_l))^{\mathrm{T}} \tag{B.13}$$

the projection of $\boldsymbol{x}$ in the M-dimensional KPCA space is given by

$$\boldsymbol{y} = \boldsymbol{U}^{\mathrm{T}} \boldsymbol{k}_x \tag{B.14}$$

B.2 Non-centred Data

Now we drop the assumption that the training data in the feature space $\boldsymbol{F}$ is centred. By defining

$$\tilde{\boldsymbol{\phi}}_i := \boldsymbol{\phi}_i - \frac{1}{l} \sum_{n=1}^{l} \boldsymbol{\phi}_n, \tag{B.15}$$

one can go through the above procedure since now all the data are centred. The problem can be finally solved by eigen-decomposing

$$\tilde{\boldsymbol{K}} = \boldsymbol{K} - \frac{1}{l} \mathbf{1}_l \boldsymbol{K} - \boldsymbol{K} \frac{1}{l} \mathbf{1}_l + \frac{1}{l^2} \mathbf{1}_l \boldsymbol{K} \mathbf{1}_l \tag{B.16}$$

where $l \times l$ matrix $\mathbf{1}_l$ is defined by

$$(\mathbf{1}_l)_{ij} := 1 \tag{B.17}$$

The projection of a new pattern $\boldsymbol{x}$ onto the KPCA space is given by

$$\boldsymbol{y} = \boldsymbol{U}^{\mathtt{T}} \tilde{\boldsymbol{k}}_x \tag{B.18}$$

where

$$\tilde{\boldsymbol{k}}_x = \boldsymbol{k}_x - \frac{1}{l}\boldsymbol{K}\mathbf{1}' - \frac{1}{l}\boldsymbol{k}_x\mathbf{1}_l + \frac{1}{l^2}\mathbf{1}'\boldsymbol{K}\mathbf{1}_l \tag{B.19}$$

Appendix C

Linear Discriminant Analysis

PCA tries to capture the *global* representative features for a given data set in the sense of least mean squared reconstruction error. However, these features may not be appropriate for the purpose of discriminating one class of patterns from others. To this aim, Linear Discriminant Analysis (LDA) (Fukunaga, 1972) has been developed. In principle, LDA seeks to find a optimal linear transform over a set of training patterns to maximise the between-class variation and minimise the within-class variation.

For a set of data $\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_C\}$ where $\mathcal{X}_c$ is a class of patterns and C is the number of such classes, the within-class scatter matrix S_w and between-class scatter matrix S_b are defined as:

$$S_w = \sum_{c=1}^{C} P_c \sum_{\mathbf{x}\in\mathcal{X}_c} (\mathbf{x} - \boldsymbol{\mu}_c)((\mathbf{x} - \boldsymbol{\mu}_c)^{\mathtt{T}} \tag{C.1}$$

$$S_b = \sum_{c=1}^{C} P_c(\boldsymbol{\mu}_c - \boldsymbol{\mu})(\boldsymbol{\mu}_c - \boldsymbol{\mu})^{\mathtt{T}} \tag{C.2}$$

where $\boldsymbol{\mu}_c$ is the mean vector of class $\mathcal{X}_c$, $\boldsymbol{\mu}$ is the mean vector of the entire data set $\mathcal{X}$, and P_c is the prior probability of class $\mathcal{X}_c$.

In order to obtain the optimal class separability, one maximises the between-class variance and minimises the within-class variance. There are several ways to formulate the criteria, for example,

$$J = tr(S_w^{-1} S_b) \tag{C.3}$$

By solving the eigen-decomposition problem, one obtains the eigenvalues of matrix $S_w^{-1}S_b$

$$\lambda_1 > \lambda_2 > \cdots > \lambda_n \tag{C.4}$$

and their corresponding eigenvectors

$$\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_n \tag{C.5}$$

Then the objective function can be expressed in terms of eigenvalues,

$$J(p) = tr(S_w^{-1}S_b) = \sum_{i=1}^{p} \lambda_i \tag{C.6}$$

Therefore, it can be maximised by choosing the first p eigenvectors.

When the dimensionality of $\mathbf{x}$ is very high, eigen-decomposing matrix $S_w^{-1}S_b$ may be problematic. In practice, one can perform PCA first on the given data set to capture the global modes, project the data onto the PCA space with reduced dimensionality, and then carry out LDA on the projected data (Etemad and Chellappa, 1997; Gong et al., 1998b).

Appendix D

Support Vector Machines

The Support Vector Machine (SVM) (Burges, 1998; Vapnik, 1995; Hearst et al., 1998; Drucker et al., 1997; Smola et al., 1998) was recently developed by V. Vapnik and his colleagues. It has been of great interest in the research areas of machine learning and pattern recognition, and has found many applications, such as face detection (Osuna et al., 1997c), text information categorisation (Hearst et al., 1998) and Optical Character Recognition (Scholkopf, 1997).

D.1 Definition of SVM Problems

The SVM is based on Structural Risk Minimisation theory. For a set of labelled patterns $\{\mathbf{x}, y\}$ where $\mathbf{x}$ is the observation and y is its interpretation y, one finds the optimal approximation

$$f(\mathbf{x}, \boldsymbol{\alpha}) = \mathbf{w} \cdot \boldsymbol{\Phi}(\mathbf{x}) + b \tag{D.1}$$

where $\boldsymbol{\alpha}$ represents the parameters of the decision function, $\boldsymbol{\Phi}$ is a map from the original data space of $\mathbf{x}$ to a high-dimensional feature space and b is the bias (Vapnik, 1995). If the interpretation y only takes values -1 and $+1$, the learning problem is referred to as *Support Vector Classification* (SVC) (Burges, 1998; Vapnik, 1995). Otherwise, if y has continuous real values, it is referred to as *Support Vector Regression* (SVR) (Vapnik, 1995; Drucker et al., 1997; Smola et al., 1998).

By introducing a kernel function

$$K(\mathbf{x}, \mathbf{y}) = \mathbf{\Phi}(\mathbf{x}) \cdot \mathbf{\Phi}(\mathbf{y}), \tag{D.2}$$

the SVC problem can be solved by maximising

$$W(\boldsymbol{\alpha}) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{D.3}$$

$$\text{subject to} \quad \sum_{i=1}^{l} \alpha_i y_i = 0 \tag{D.4}$$

$$0 \leq \alpha_i \leq C, \quad i = 1, 2, \ldots, l \tag{D.5}$$

which gives a separating function

$$f(\mathbf{x}) = \sum_{i=1}^{l} y_i \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{D.6}$$

On the other hand, the SVR problem can be solved by maximising

$$\begin{aligned} W(\alpha^*, \alpha) = & -\frac{1}{2} \sum_{i,j=1}^{l} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) K(\mathbf{x}_i, \mathbf{x}_j) \\ & -\varepsilon \sum_{i=1}^{l} (\alpha_i^* + \alpha_i) + \sum_{i=1}^{l} y_i (\alpha_i^* - \alpha_i) \end{aligned} \tag{D.7}$$

$$\text{subject to} \quad \sum_{i=1}^{l} (\alpha_i^* - \alpha_i) = 0 \tag{D.8}$$

$$0 \leq \alpha_i^*, \alpha_i \leq C \tag{D.9}$$

which provides the solution

$$f(\mathbf{x}) = \sum_{i=1}^{l} (\alpha_i^* - \alpha_i) K(\mathbf{x}, \mathbf{x}_i) + b \tag{D.10}$$

It is interesting to notice that only a few parameters α_i take non-zero values, i.e. only those "important" examples, known as *Support Vectors* (SVs), are selected to construct the optimal approximation functions (D.6) and (D.10). Functions (D.6) and (D.10) are linear combinations of the SVs in high-dimensional feature space. However, instead of computing the map $\mathbf{\Phi}$ and its inner product explicitly which are computational intensive or even impossible due to the high dimensionality, one only needs to compute the kernel function (D.2) with great ease.

D.2 Algorithms for Solving SVMs

Both the SVC and SVR problems are in the format of Quadratic Programming (QP). Some traditional algorithms, such as Quasi Newton Method, Conjugate

Gradient Descent Method, usually cannot provide a feasible solution to those problems since the Hessen matrix may be too huge to fit in the memory. Some algorithms specifically designed for SVMs have been proposed including the Chunking Algorithm, the Decomposition Algorithm and the Sequential Minimal Optimisation Algorithm.

D.2.1 Chunking Algorithm

Vapnik (Vapnik, 1995) proposed the Chunking Algorithm for solving the SVM QP problem. The algorithm initiates with a random selected set of data to build a smaller QP problem. At each step, it solves the smaller QP problem, and constructs the problem of the next step with all the non-zero multipliers, plus the M (defined *a priori*, according to the computation and memory requirement) worst examples which violate the Kuhn-Tucker (KT) conditions (Walsh, 1975; Bazaraa et al., 1993). If the number of the violating examples is less than M, then all the violating examples are added in. When no more example violates the KT conditions, the algorithm terminates, and all the Support Vectors for the problem, together with their non-zero multipliers are identified.

The Chunking Algorithm reduces the size of Hessen matrix to a great degree, but the matrix with the reduced size may still take a lot of memory for some large scale problems.

D.2.2 Decomposition Algorithm

Osuna (Osuna et al., 1997a; Osuna et al., 1997b; Osuna et al., 1997c) proposed the idea of the Decomposition Algorithm. It uses a fixed size working set B, whose size is big enough to contain all Support Vectors, and small enough to efficiently stored in memory. The algorithm works in the following way:

1. Arbitrarily choose $|B|$ points from the data set;

2. Solve the sub-problem defined by the variables in B;

3. While there exists some $j \in N$, such that x_j violates the KT conditions, replace $\lambda_i = 0, i \in B$ with $\lambda_j = 0$ and solve the new sub-problem.

Since the objective function is convex, and the constraints are also convex, the algorithm must converge to the global optimal solution in a finite number of iterations.

Although Osuna's algorithm suggests that only one training example is added into and another one is deleted from the working set, other researchers add and delete multiple examples using various techniques to achieve an improved training speed.

D.2.3 Sequential Minimal Optimisation

Sequential Minimal Optimisation (SMO) (Platt, 1998) is a simple algorithm that solves the SVM QP problem in a analytical way and without any extra matrix storage. To some extent, it is similar to Osuna's Decomposition Algorithm for it also decomposed the original problem into small sub-problems. However, the sub-problems are in the simplest format, i.e. only dealing with two Lagrange multipliers. At each step of SMO algorithm, the two multipliers are evaluated to give the optimal value of the objective function. Then another two training examples which violate the KT condition most seriously are selected to construct the problem in the next iteration. When no training example violates the KT conditions, the decision function is given by linearly combining the Support Vectors with their relevant multipliers.

Bibliography

Akimoto, T., Suenaga, Y., and Wallace, R. (1993). Automatic creation of 3d facial models. *IEEE Computer Graphics and Applications*, 13(5):16–22.

Atick, J., Griffin, P., and Redlich, A. (1996). Statistical approach to shape from shading: reconstruction of 3d face surfaces from single 2d images. *Neural Computation*, 8(6):1321–1341.

Baker, S. and Kanade, T. (2000). Hallucinating faces. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 83–88, Grenoble, France.

Baron, R. (1981). Mechanisms of human facial recognition. *International Journal of Man Machine Studies*, 15:137–178.

Bassili, J. (1979). Emotion recognition: the role of facial movement and the relative importance of upper and lower areas of the face. *Journal of Personality and Social Psychology*, 37:2049–2059.

Baudat, G. and Anouar, F. (2000). Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12:2385–2404.

Baumberg, A. and Hogg, D. (1994). An efficient method for contour tracking using active shape models. In *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 194–199.

Bazaraa, M., Sherali, H., and Shetty, C. (1993). *Nonlinear programming.* John Wiley & Sons.

Bennett, A. and Craw, I. (1991). Finding image features using deformable templates and detailed prior statistical knowledge. In *British Machine Vision Conference*, pages 233–239, Glasgow, UK.

Beymer, D., Shashua, A., and Poggio, T. (1993). Example based image analysis and synthesis. Technical report, Massachusetts Institute of Technology. A. I. Memo 1431.

Bishop, C., Svensen, M., and Williams, C. (1998). GTM: the generative topographic mapping. *Neural Computation*, 10(1):215–234.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford.

Blake, A., Curwen, R., and Zisserman, A. (1993). A framework for spatiotemporal control in the tracking of visual contours. *International Journal of Computer Vision*, 11(2):127–145.

Blake, A. and Isard, M. (1998). *Active Contours.* Springer-Verlag London Limited.

Blake, A., Isard, M., and Reynard, D. (1995). Learning to track the visual motion of contours. *Artificial Intelligence*, 73(1-2):179–212.

Bookstein, F. (1989). Principal warps: Thin-plate splines and teh decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585.

Bowden, R., Michell, T., and Sarhadi, M. (1998). Reconstructing 3d pose and motion from a single camera view. In *British Machine Vision Conference*, volume 2, pages 904–1013, Southampton, UK.

Brammer, K. and Siffling, G. (1989). *Kalman-Bucy Filters.* Artech House, Norwood, USA.

Bruce, V., Burton, A., and Hancock, P. (1998a). Comparisons between human and computer recognition of faces. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 408–413, Nara, Japan.

Bruce, V., Hancock, P., and A.Burton (1998b). Human face perception and identification. In Wechsler, Philips, Bruce, Fogelman-Soulie, and Huang, editors, *Face Recognition: From Theory to Applications*, pages 51–72. Springer-Verlag.

Bruce, V. and Young, A. (1998). *In the eye of the beholder: the science of face perception*. Oxford University Press, Oxford, UK.

Brunelli, R. and Poggio, T. (1993). Face recognition: features vs. templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1042–1062.

Burges, C. J. C. (1996). Simplified support vector decision rules. In Saitta, L., editor, *Proceedings, 13th Intl. Conf. on Machine Learning*, pages 71–77, San Mateo, CA. Morgan Kaufmann.

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):1–47.

Burges, C. J. C. and Scholkopf, B. (1997). Improving the accuracy and speed of support vector learning machines. In Mozer, M., Jordan, M., and Petsche, T., editors, *Advances in Neural Information Processing Systems 9*, pages 375–381. MIT Press, Cambridge, MA.

Burl, M. and Perona, P. (1996). Recognition of planar object classes. In *IEEE Conference on Computer Vision and Patter Recognition*, pages 223–230.

Burr, D. (1981). Elastic matching of line drawings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(6):708–713.

Chen, C. and Huang, C. (1992). Human face recognition from a single front view. *International Journal of Pattern Recognition and Artificial Intelligence*, 6(4):571–593.

Choi, C., Okazaki, T., Harashima, H., and Takebe, T. (1991). A system of analyzing and synthesizing facial images. In *1991 IEEE International Sympoisum on Circuits and Systems*, volume 5, pages 2665–2668, Singapore.

Choudhury, T., Clarkson, B., Jebara, T., and Pentland, A. (1999). Multimodal person recognition using unconstrained audio and video. In *International Conference on Audio- and Video-Based Person Authentication*, pages 176–181.

Comaniciu, D. and Meer, P. (1999). Mean shift analysis and applications. In *IEEE International Conference on Computer Vision*, volume 2, pages 1197–1203, Kerkyra, Greece.

Cootes, T., C.Taylor, and A.Lanitis (1994). Active shape models: evaluation of a multi-resolution method for improving image search. In *British Machine Vision Conference*, volume 1, pages 327–336, York, England.

Cootes, T., Edwards, G., and Taylor, C. (1998). Active appearance models. In *European Conference on Computer Vision*, volume 2, pages 484–498, Freiburg, Germany.

Cootes, T., Hill, A., and Taylor, C. (1995a). Medical image interpretation using active shape models: Recent advances. In *14th Conference on Information Processing in Medical Imaging*, pages 371–372, France.

Cootes, T. and Taylor, C. (1997). A mixture model for representing shape variation. In *British Machine Vision Conference*, volume 1, pages 110–119.

Cootes, T. and Taylor, C. (1999). A mixture model for representing shape variation. *Image and Vision Computing*, 17:567–573.

Cootes, T., Taylor, C., Cooper, D., and Graham, J. (1995b). Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59.

Cootes, T., Walker, K., and Taylor, C. (2000). View-based active appearance models. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 227–232, Grenoble, France.

Craw, I. and Cameron, P. (1992). Face recognition by computer. In Hogg, D. and Boyle, R., editors, *British Machine Vision Conference*, pages 498–507.

Craw, I., Costen, N., Kato, T., Robertson, G., and Akamatsu, S. (1995). Automatic face recognition: combining configuration and texture. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 53–58, Zurich, Switzerland.

Craw, I., Tock, D., and Bennett, A. (1992). Finding face features. In *European Conference on Computer Vision*, pages 92–96, Santa Margherita Ligure, Italy.

Davis, J. and Bobick, A. (1997). The representation and recognition of action using temporal templates. In *IEEE Conference on Computer Vision and Patter Recognition*, pages 928–934.

DeCarlo, D. and Metaxas, D. (1996a). Deformable model-based face shape and motion estimation. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 146–150, Vermont, US.

DeCarlo, D. and Metaxas, D. (1996b). Integration of optical flow and deformable models with applications to human face shape and motion estimation. In *IEEE Conference on Computer Vision and Patter Recognition*, pages 231–238, San Francisco, CA, USA.

DeCarlo, D. and Metaxas, D. (2000). Optical flow constraints on deformable models with applications to face tracking. *International Journal of Computer Vision*, 38(2):99–127.

Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A., and Vapnik, V. (1997). Support vector regression machines. In Mozer, M., Jordan, M., and Petsche, T., editors, *Advances in Neural Information Processing Systems 9*. MIT Press, Cambridge, MA.

Edwards, G., Lanitis, A., Taylor, C., and Cootes, T. (1996). Statistical models of face images - improving specificity. In *British Machine Vision Conference*, volume 2, pages 765–774, Edinburgh, Scotland.

Edwards, G., Lanitis, A., Taylor, C., and Cootes, T. (1998a). Statistical models of face images - improving specificity. *Image and Vision Computing*, 16(3):203–211.

Edwards, G., Taylor, C., and Cootes, T. (1998b). Interpreting face images using active appearance models. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 300–305, Nara, Japan.

Edwards, G., Taylor, C., and Cootes, T. (1998c). Learning to identify and track faces in sequences. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 260–267, Nara, Japan.

Edwards, G., Taylor, C., and Cootes, T. (1999). Improving identification performance by integrating evidence from sequences. In *IEEE Conference on Computer Vision and Patter Recognition*, volume 1, pages 486–491, Fort Collins, CO, USA.

Elagin, E., Steffens, J., and Neven, H. (1998). Automatic pose estimation system for human faces based on bunch graph matching technology. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 136–141, Nara, Japan.

Etemad, K. and Chellappa, R. (1997). Discriminant analysis for recognition of human face images. *Journal of the Optical Society of America A-Optics & Image Science*, 14(8):1724–1733.

Ezzat, T. and Poggio, T. (1996). Facial analysis and synthesis using image-based methods. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 116–121, Vermont, US.

Fischler, M. and Elschlager, R. (1973). The representation and matching of pictorial structures. *IEEE Transactions on Computers*, C-22(1):67–92.

Fisher, R. A. (1938). The statistical utilization of multiple measurements. *Annals of Eugenics*, 8:376–386.

Fukunaga, K. (1972). *Introduction to statistical pattern recognitiion.* Academic Press.

Galton, F. (1883). *Inquiries into human faculty and its development.* MacMillan. Second Edition: Dent 1907, London. Also available at http://www.mugu.com/galton/human-faculty/pdf/.

Gee, A. and Cipolla, R. (1994). Determining the gaze of faces in images. *Image and Vision Computing*, 12(10):639–647.

Gong, S., McKenna, S., and Collins, J. (1996). An investigation into face pose distributions. In *Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, pages 265–270, Vermont, US.

Gong, S., McKenna, S., and Psarrou, A. (April 2000). *Dynamic Vision: From Images to Face Recognition.* World Scientific Publishing and Imperial College Press.

Gong, S., Ong, E., and Loft, P. (1998a). Appearance-based face recognition under large head rotations in depth. In *Asian Conference on Computer Vision*, volume 2, pages 679–686, Hong Kong.

Gong, S., Ong, E.-J., and McKenna, S. (1998b). Learning to associate faces across views in vector space of similarities to prototypes. In *British Machine Vision Conference*, pages 54–64, Southampton, England.

Gong, S., Psarrou, A., Katsouli, I., and Palavouzis, P. (1994). Tracking and recognition of face sequences. In *European Workshop on Combined Real and Synthetic Image Processing for Broadcast and Video Production*, pages 96–112, Hamburg, Germany.

Graf, H., Chen, T., Petajan, E., and Cosatto, E. (1995). Locating faces and facial parts. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 41–46, Zurich, Switzerland.

Hastie, T. and Stuetzle, W. (1989). Principal curves. *Journal of the American Statistical Association*, 84(406):502–516.

Heap, A. and Hogg, D. (1997). Improving specificity in pdms using a hierarchical approach. In *British Machine Vision Conference*, volume 1, pages 80–89.

Heap, A. and Hogg, D. (1998). Wormholes in shape space: Tracking through discontinuous changes in shape. In *IEEE International Conference on Computer Vision*, pages 344–350.

Hearst, M., Scholkopf, B., Dumais, S., Osuna, E., and Platt, J. (1998). Trends and controversies – support vector machines. *IEEE Intelligent Systems*, 13(4).

Hildreth, E. (1984). *The measurement of visual motion*. MIT Press, Cambridge, USA.

Horn, B. and Brooks, M. (1989). *Shape from shading*. MIT Press, Cambridge, USA.

Horprasert, T., Yacoob, Y., and Davis, L. (1996). Computing 3d head orientation from a monocular image squence. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 242–247, Vermont, USA.

Howell, A. and Buxton, H. (1996). Towards unconstrained face recognition from image sequences. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 224–229, Vermont, USA.

Hunke, M. and Waibel, A. (1994). Face locating and tracking for human-computer interaction. In *28th Asilomar Conference on Signals, Systems and Computers*, California.

Hutchinson, T. E., White Jr., K. P., Martin, W. N., Reichert, K. C., and Frey, L. A. (1989). Human-computer interaction using eye-gaze input. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(6):1527–1534.

Jebara, T. and Pentland, A. (1997). Parametrized structure from motion for 3D adaptive feedback tracking of faces. In *IEEE Conference on Computer Vision and Patter Recognition*.

Kanade, T. (1973). *Picture Processing by Computer Complex and Recognition of Human Faces*. PhD thesis, Dept. of Information Science, Kyoto University.

Kass, M., Witkin, A., and Terzopoulos, D. (1987a). Snakes: Active contour models. In *IEEE International Conference on Computer Vision*, pages 259–268, London.

Kass, M., Witkin, A., and Terzopoulos, D. (1987b). Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331.

Kaucic, R. and Blake, A. (1998). Accurate, real-time, unadorned lip tracking. In *IEEE International Conference on Computer Vision*, pages 370–375, New Delhi, India.

Kaya, Y. and Kobayashi, K. (1972). A basic study on human face recognition. In Watanabe, S., editor, *Frontiers of Pattern Recognition*, pages 265–289. Academic Press, New York, NY, USA.

Kirby, M. and Sirovich, L. (1990). Applications of the karhunen-loeve procedure for the characterisation of human faces. *Perception*, 15:595–602.

Kjeldsen, R. and Kender, J. (1996). Finding skin in color images. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 312–317, Vermont, USA.

Knight, B. and Johnston, A. (1997). The role of movement in face recognition. *Visual Cognition*, 4:265–274.

Kramer, M. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2):233–243.

Kruger, N., Potzsch, M., and Maurer, T. (1996). Estimation face position and pose with labelled graphs. In *British Machine Vision Conference*.

Kruger, N., Potzsch, M., and von der Malsburg, C. (1997). Determination of face position and pose with a learned representation based on labelled graphs. *Image and Vision Computing*, 15(8):665–673.

Lades, M., Vorbruggen, J., Buhmann, J., Lange, J., Malsburg, C., Wurtz, R., and Konen, W. (1993). Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42(3):300–311.

Lanitis, A., Taylor, C., and Cootes, T. (1994). An automatic face identification system using flexible appearance models. In *British Machine Vision Conference*, volume 1, pages 65–74.

Lanitis, A., Taylor, C., and Cootes, T. (1995a). Automatic face identification system using flexible appearance models. *Image and Vision Computing*, 13(5):392–401.

Lanitis, A., Taylor, C., and Cootes, T. (1995b). A unified approach to coding and interpreting face images. In *IEEE International Conference on Computer Vision*, pages 368–373.

Lanitis, A., Taylor, C., and Cootes, T. (1997). Automatic interpretation and coding of face images using flexible models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):743–756.

Lanitis, A., Taylor, C., Cootes, T., and Ahmed, T. (1995c). Automatic interpretation of human faces and hand gestures using flexible models. In *IEEE*

International Conference on Automatic Face & Gesture Recognition, pages 98–102, Zurich, Switzerland.

Lapedes, A. and Faber, R. (1987). How neural nets work. In *IEEE conference on neural networks*, pages 442–456.

Li, H., Roivainen, P., and forchheimer, R. (1993). 3-d motion estimation in model-based facial image coding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):545–555.

Matsumoto, Y. and Zelinsky, A. (2000). An algorithm for real-time stereo vision implementation of head pose and gaze direction measurement. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 499–504, Grenoble, France.

Maurer, T. and Malsburg, C. (1995). Single-view based recognition of faces rotated in depth. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 248–253, Zurich, Switzerland.

Maurer, T. and von der Malsburg, C. (1996). Tracking and learning graphs and pose on image sequences of faces. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 176–181, Vermont, USA.

McKenna, S. and Gong, S. (1996). Tracking faces. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 271–276, Vermont, US.

McKenna, S. and Gong, S. (1998a). Face recognition from sequences using models of identity. In *Asian Conference on Computer Vision*, Hong Kong.

McKenna, S. and Gong, S. (1998b). Real time face pose estimation. *International Journal on Real Time Imaging, Special Issue on Real-time Visual Monitoring and Inspection*, 4:333–347.

McKenna, S. and Gong, S. (1998c). Recognising moving faces. In Wechsler, Philips, Bruce, Fogelman-Soulie, and Huang, editors, *Face Recognition: From Theory to Applications*, pages 578–588. Springer-Verlag.

McKenna, S., Gong, S., and Collins, J. (1996). Face tracking and pose representation. In *British Machine Vision Conference*, pages 755–764, Edinburgh, Scotland.

McKenna, S., Gong, S., and Raja, Y. (1997). Face recognition in dynamic scenes. In *British Machine Vision Conference*, pages 140–151, Colchester, UK.

McKenna, S., Gong, S., and Raja, Y. (1998). Modelling facial colour and identity with gaussian mixtures. *Pattern Recognition*, 31(12):1883–1892.

Mika, S., Ratsch, G., Weston, J., Scholkopf, B., and Muller, K. (1999). Fisher discriminant analysis with kernels. In *IEEE Neural Networks for Signal Processing Workshop*, pages 41–48.

Moghaddam, B. and Pentland, A. (1994). Face recognition using view-based and modular eigenspaces. In *Automatic Systems for the Identification and Inspection of Humans, SPIE*, volume 2277.

Moghaddam, B. and Pentland, A. (1995). Probabilistic visual learning for object detection. In *IEEE International Conference on Computer Vision*, pages 786–793, Cambridge, MA, USA.

Moghaddam, B. and Pentland, A. (1997). Probalilistic visual learning for object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):137–143.

Moghaddam, B., Wahid, W., and Pentland, A. (1998). Beyond eigenfaces: probalilistic matching for face recognition. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 30–35, Nara, Japan.

Murase, H. and Nayar, S. K. (1994). Illumination planning for object recognition using parametric eigenspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(12):1219–1227.

Murase, H. and Nayar, S. K. (1995). Visual learning and recognition of 3-D objects from appearance. *International Journal of Computer Vision*, 14:5–24.

Ng, J. and Gong, S. (1999a). Learning support vector machines for a multi-view face model. In *British Machine Vision Conference*, volume 2, pages 503–512, Nottingham, UK.

Ng, J. and Gong, S. (1999b). Multi-view face detection and pose estimation using a composite support vector machine across the view sphere. In *IEEE International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pages 14–21, Corfu, Greece.

Okubo, M. and Watanabe, T. (1998). Lip motion capture and its application to 3-D molding. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 187–192, Nara, Japan.

Ong, E. and Gong, S. (1999a). A dynamic human model using hybrid 2d-3d representations in hierarchical pca space. In *British Machine Vision Conference*, pages 33–42, Nottingham, UK.

Ong, E. and Gong, S. (1999b). Tracking hybrid 2d-3d human models through multiple views. In *IEEE International Workshop on Modelling People*, pages 11–18, Corfu, Greece.

Osuna, E., Freund, R., and Girosi, F. (1997a). An improved training algorithm for support vector machines. In Principe, J., Gile, L., Morgan, N., and Wilson, E., editors, *Neural Networks for Signal Processing VII – Proceedings of the 1997 IEEE Workshop*, pages 276–285, New York. IEEE.

Osuna, E., Freund, R., and Girosi, F. (1997b). Support vector machines: Training and applications. Technical report, Massachusetts Institute of Technology. AI Memo 1602.

Osuna, E., Freund, R., and Girosi, F. (1997c). Training support vector machines: An application to face detection. In *Proc. Computer Vision and Pattern Recognition'97*, pages 130–136.

Parke, F. (1974). *A Parametric Model for Human Faces*. PhD thesis, University of Utah, Salt Lake City, Utah. UTEC-CSc75-047.

Parke, F. (1975). A model for human faces that allows speech synchronized animation. *Computers & Graphics*, 1:3–4.

Pentland, A. and Horowitz, B. (1991). Recovery of nonrigid motion and structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):730–742.

Pentland, A., Moghaddam, B., and Starner, T. (1994). View-based and modular eigenspaces for face recognition. In *IEEE Conference on Computer Vision and Patter Recognition*, pages 84–91, Seatle.

Platt, J. C. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines. Technical report, Microsoft Research. Technical Report MSR-TR-98-14.

Poggio, T. and Vetter, T. (1992). Recognition and structure from one 2d model view: Observations on prototypes, object classes, and symmetries. Technical report, Artificial Intelligence Laboratory, Massachusetts Institute of Technology. A.I. Memo No. 1347.

Raja, Y., McKenna, S., and Gong, S. (1998a). Colour model selection and adaptation in dynamic scenes. In *European Conference on Computer Vision*, Freiburg, Germany.

Raja, Y., McKenna, S., and Gong, S. (1998b). Segmentation and tracking using colour mixture models. In *Asian Conference on Computer Vision*, Hong Kong.

Romdhani, S., Gong, S., and Psarrou, A. (1999a). Learning a single active face shape model across views. In *IEEE International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pages 31–38, Corfu, Greece.

Romdhani, S., Gong, S., and Psarrou, A. (1999b). A multi-view nonlinear active shape model using kernel pca. In *British Machine Vision Conference*, pages 483–492, Nottingham, UK.

Romdhani, S., Gong, S., and Psarrou, A. (June 2000b). On utilising template and feature-based correspondence in multi-view appearance models. In *European Conference on Computer Vision*, volume 1, pages 799–813, Dublin, Ireland.

Romdhani, S., Gong, S., and Psarrou, A. (September 2000a). A generic face appearance model of shape and texture under very large pose variations from profile to profile views. In *International Conference on Pattern Recognition*, volume 1, pages 1060–1063, Barcelona, Spain.

Roweis, S. (1998). EM algorithms for PCA and SPCA. In Jordan, M. I., Kearns, M. J., and Solla, S. A., editors, *Advances in Neural Information Processing Systems*, volume 10. The MIT Press.

Rowley, H., Baluja, S., and Kanade, T. (1996). Neural network-based face detection. In *IEEE Conference on Computer Vision and Patter Recognition*, pages 203–207, San Francisco, CA, USA.

Rowley, H., Baluja, S., and Kanade, T. (1997). Rotation invariant neural network-based face detection. Technical report, School of Computer Science, Carnegie Mellon University. CMU-CS-97-201.

Rowley, H., Baluja, S., and Kanade, T. (1998a). Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Inteligence*, 20(1).

Rowley, H., Baluja, S., and Kanade, T. (1998b). Rotation invariant neural network-based face detection. In *IEEE Conference on Computer Vision and Patter Recognition*.

Scholkopf, B. (1997). *Support Vector Learning*. R. Oldenbourg Verlag, Munich.

Scholkopf, B., Bartlett, P., Smola, A., and Williamson, R. (1998a). Support vector regression with automatic accuracy control. In Niklasson, L., Bod'en, M., and Ziemke, T., editors, *Proceedings of the 8th International Conference on Artificial Neural Networks, Perspectives in Neural Computing*, pages 111–116, Berlin. Springer Verlag.

Scholkopf, B., Mika, S., Smola, A., Ratsch, G., and Muller, K.-R. (1998b). Kernel pca pattern reconstruction via approximate pre-images. In Niklasson, L., Bod'en, M., and Ziemke, T., editors, *Proceedings of the 8th International Conference on Artificial Neural Networks, Perspectives in Neural Computing*, Berlin. Springer Verlag.

Scholkopf, B., Smola, A., and Muller, K.-R. (1996). Nonlinear component analysis as a kernel eigenvalue problem. Technical report, Max-Planck-Institut fur biologische Kybernetik. Technical Report 44.

Scholkopf, B., Smola, A., and Muller, K.-R. (1997). Kernel principal component analysis. In Gerstner, W., Germond, A., Hasler, M., and Nicoud, J.-D., editors, *Artificial Neural Networks – ICANN'97*, pages 583–588, Berlin. Springer Lecture Notes in Computer Science.

Scholkopf, B., Smola, A., and Muller, K.-R. (1998c). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319.

Scholkopf, B., Smola, A., Williamson, R., and Bartlett, P. L. (2000). New support vector algorithms. *Neural Computation*, 12(5):1207–1245.

Shakunaga, T., Ogawa, K., and Oki, S. (1998). Integration of eigentemplate and structure matching for automatic facial feature detection. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 94–99, Nara, Japan.

Sherrah, J., Gong, S., and Ong, E. (2001). Face distribution in similarity space under varying head pose. *Image and Vision Computing*, 19(11).

Shimizu, I., Zhang, Z., Akamatsu, S., and Deguchi, K. (1998). Head pose determination from one image using a generic model. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 100–105, Nara, Japan.

Sinha, P. (1994). Object recognition via image invariances. *Investigative Ophthalmology and Visual Science*, 35(4):1626.

Sirovich, L. and Kirby, M. (1987). Low-dimensional procedure for the characterization of human faces. *Journal of Optical Society of America*, 4:519–524.

Smola, A., Scholkopf, B., and Muller, K.-R. (1998). General cost functions for support vector regression. In Downs, T., Frean, M., and Gallagher, M., editors, *Proc. of the Ninth Australian Conf. on Neural Networks*, pages 79–83, Brisbane, Australia.

Smola, A., Scholkopf, B., and Rotsch, G. (1999). Linear programs for automatic accuracy control in regression. In *The Ninth International Conference on Artificial Neural Networks*, pages 575–580, London.

Sonka, M., Hlavac, V., and Boyle, R. (1996). *Image processing, anylysis and machine vision*. International Thomson Computer Press.

Soulie, F., Viennet, F., and Lamy, B. (1993). Multi-modular neural network architectures: applications in optical character and human face recognition. *International Journal of Pattern Recognition and Artifical Intelligence*, 7(4):721–755.

Stauffer, C. and Grimson, W. (1999). Adaptive background mixture models for real-time tracking. In *IEEE Conference on Computer Vision and Patter Recognition*, volume 2, pages 246–252, Fort Collins, CO, USA.

Steffens, J., Elagin, E., and Neven, H. (1998). Personspotter - fast and robust system for human detection, tracking and recognition. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 516–521, Nara, Japan.

Sung, K. and Poggio, T. (1994). Example-based learning for view-based human face detection. Technical report, Massachusetts Institute of Technology. A.I. MEMO 1521.

Swets, D. and Weng, J. (1996). Using discriminant eigenfeatures for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):831–836.

Swets, D. and Weng, J. (1999). Hierarchical discriminant analysis for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):386–401.

Swets, J. and Pickett, R. (1982). *Evaluation of Diagnostic Systems: Methods from Signal Detection Theory*. Academic Press, New York, USA.

Tibshirani, R. (1992). Principal curves revisited. *Statistics & Computing*, 2(4):183–190.

Tipping, M. E. and Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 21(3):611–622.

Toyama, K., Krumm, J., Brumitt, B., and Meyers, B. (1999). Wallflower: principles and practice of background maintenance. In *IEEE International Conference on Computer Vision*, volume 1, pages 255–261, Kerkyra, Greece.

Turk, M. and Pentland, A. (1989). Face processing: models for recognition. In *Intelligent Robots and Computer Vision VIII: Algorithms and Techniques*, pages 22–32, Philadelphia, PA, USA.

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

Ullman, S. (1979). *The Interpretation of Visual Motion.* MIT Press, Cambridge, USA.

Vanderbei, R. (1994). Loqo: An interior point code for quadratic programming. Technical report, Princeton University. Technical Report SOR 94-15.

Vanderbei, R. J. (1997). Loqo user's manual-version 3.10. Technical report, Princeton University, Statistics and Operations Research. Technical Report SOR-97-08.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory.* Springer Verlag, New York.

Vapnik, V. (1998). *Statistical Learning Theory.* John Wiley & Sons.

Vetter, T. (1996). Learning novel views to a single face image. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 22–27, Vermont, USA.

Vetter, T. (1998). Synthesis of novel views from a single face image. *International Journal of Computer Vision*, 28(2):103–116.

Vetter, T. and Blanz, V. (1998). Generalization to novel views from a single face image. In Wechsler, Philips, Bruce, Fogelman-Soulie, and Huang, editors, *Face Recognition: From Theory to Applications*, pages 310–326. Springer-Verlag.

Vetter, T. and Poggio, T. (1997). Linear object classes and image synthesis from a single example image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):733–742.

Waite, J. B. and Welsh, W. J. (1990a). An application of active contour models to head boundary location. In *British Machine Vision Conference*, pages 407–412.

Waite, J. B. and Welsh, W. J. (1990b). Head boundary location using snakes. *British Telecom Technology Journal*, 8(3):127–136.

Walsh, G. (1975). *Methods of optimization.* John Wiley & Sons.

Webb, A. (1996). An approach to nonlinear principal components analysis using radially symmetrical kernel functions. *Statistics and Computing*, 6(2):159–168.

Wiskott, L., Fellous, J., Kruger, N., and Malsburg, C. (1997). Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):775–779.

Wu, H., Yokoyama, T., Pramadihanto, D., and Yachida, M. (1996). Face and facial feature extraction from color image. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 345–350, Vermont, USA.

Wu, Y. and Toyama, K. (2000). Wide-range, person- and illumination- insensitive head orientation estimation. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 183–188, Grenoble, France.

Xu, M. and Adatsuka, T. (1998). Detecting head pose from stereo image sequence for active face recognition. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 82–87, Nara, Japan.

Yamaguchi, O., Fukui, K., and Maeda, K. (1998). Face recognition using temporal image sequence. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 318–323, Nara, Japan.

Yang, G. and Huang, T. (1994). Human face detection in a complex background. *Pattern Recognition*, 27:53–63.

Yokoyama, T., Yagi, Y., and Yachida, M. (1998). Facial contour extraction model. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 254–259, Nara, Japan.

Yow, K. and Cipolla, R. (1996a). Detection of human faces under scale, orientation and viewpoint variations. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 295–300, Vermont, USA.

Yow, K. and Cipolla, R. (1996b). A probalilistic framework for perceptual grouping of features for human face detection. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 16–21, Vermont, USA.

Yuille, A. and Hallinan, P. (1992). Deformable templates. In Blake, A. and Yuille, A., editors, *Active Vision*, pages 20–38. MIT.

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

Zhao, W. and Chellappa, R. (2000). SFS based view synthesis for robust face recognition. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 285–292, Grenoble, France.

Zhao, W., Chellappa, R., and Krishnaswamy, A. (1998a). Discriminant analysis of principal components for face recognition. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 336–341, Nara, Japan.

Zhao, W., Krishnaswamy, A., Chellappa, R., Swets, D., and Weng, J. (1998b). Discriminant analysis of principal components for face recognition. In Wechsler, Philips, Bruce, Fogelman-Soulie, and Huang, editors, *Face Recognition: From Theory to Applications*, pages 73–85. Springer-Verlag.