FACE VERIFICATION WITH STATISTICAL MODELS OF SHAPE AND APPEARANCE

BY

ODIETE OBARO OGHENEYOMA

A THESIS

SUBMITTED TO THE AFRICAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

ABUJA – NIGERIA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

AWARD OF MASTER OF SCIENCE IN COMPUTER SCIENCE

SUPERVISOR: DR KOLA BABALOLA

APRIL, 2013

FACE VERIFICATION WITH STATISTICAL MODELS OF SHAPE AND APPEARANCE

A THESIS APPROVED

BY

SUPERVISOR
DR KOLA BABALOLA

MEMBER
PROF MAMADOU TRAORE

MEMBER
NAME

DEDICATION

To my parents

ACKNOWLEDGEMENTS

I would like to express my gratitude to those who helped me complete this thesis. I appreciate my supervisor from African University of Science and Technology, Abuja, Dr. Kola Babalola for the advice and challenge to undertake the goal of this thesis.

I want to thank Prof. Mamadou Troare, the head of department of the computer science stream for his advice, scholarly assistance and everything he did in enhancing my graduate school experience.

I also like to say a big thank you to Prof. Adel Bouhoula, the initial challenge of working with him on a different subject helped prepare me in many ways for the challenge of doing research.

My Master's program was fully sponsored by The Nelson Mandela Institute and I am sincerely grateful for this. Without them the past 18 months and this thesis may never have been.

Finally, I appreciate the Almighty God for graciously granting me life, wisdom and strength in completing this Masters program. He alone deserves all the glory and adoration.

ABSTRACT

Research in computer vision and machine learning is a significant part of research in computer science departments of many leading institutions resulting in ideas and products that have direct applications in different industries such as medical image segmentation in the medical industry, and face recognition and tracking in the entertainment and security industry.

Face recognition is a significant part of research in computer vision and machine learning and has a wide range of applications in security, human computer interaction and artificial intelligence in general. The main goal of this thesis was to build a code repository to facilitate research in computer vision and machine learning at The African University of Science and Technology, Abuja. Our work concentrated on implementing some statistical shape and appearance algorithms used in face recognition research.

We trained an appearance model and active shape models for an experiment in face verification. We evaluated the use of parameters from the appearance model for face verification using four very common metrics: Mahalanobis distance, Euclidean distance, normalized correlation and Manhattan distance.

Our results showed that normalized correlation performed least while there was very little difference in the performance of the others.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Research in computer vision and machine learning is a significant part of research in leading computer science departments worldwide. It has led to many breakthroughs in both academic research and commercial applications.

A currently thriving area of research in computer vision and machine learning is face recognition. It was described in [30] as one of the most successful applications of image analysis and understanding, stating two reasons for the strong research efforts in this area as the wide range of applications it provides and the availability of the technology to support the research.

A study of some of the leading institutions in this field and in other fields in computer science reveals each has a thriving code repository which has been built over the years by researchers and is available to new researchers to build upon thus speeding up research work.

Examples include VisionX of Cornell University Vision and Image Analysis Group[1]. FSL, of the Analysis Group, FMRIB, Oxford, UK[2] and STAIR Vision Library (SVL)[3] developed by a Stanford PhD student for research initially to support the Stanford AI robot project.

One goal of the computer vision and machine learning (CVML) group at The Africa University of Science and Technology, Abuja is to build its own code repository from ground up to facilitate research within the group. Many of the algorithms implemented for the repository are not freely available elsewhere or at least not in the organized form implemented in the repository.

Our main goal for this thesis was to build a code repository for the computer vision and machine learning group focusing on research in face recognition and computer animation, and to utilize the code base to perform some experiments in face verification.

The experiments performed evaluated the use of Mahalanobis distance, Euclidean distance, Manhattan distance and normalized correlation as metrics for face verification using parameters

1. http://www.via.cornell.edu/visionx.html
2. http://www.fmrib.ox.ac.uk/fsl/
3. http://ai.stanford.edu/~sgould/svl/

obtained from an appearance model. These measures were chosen because they are the most commonly used.

This chapter briefly describes the initial design of the code repository, the scope covered for this thesis, our contributions and the layout of the report.

1.1 Scope

The figure below shows the proposed framework for the code repository for development during this thesis work.

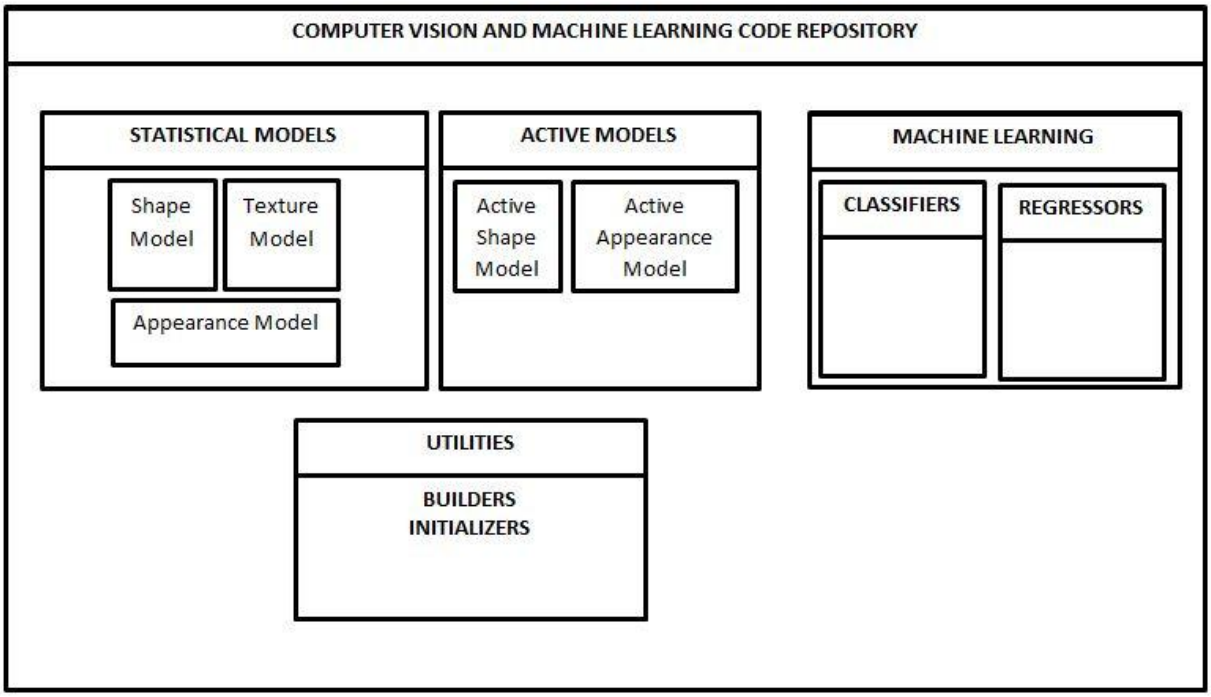


Figure 1.1: CVML Code repository initial framework

Our tasks in this thesis can be summarized in these steps.

1. Develop the following models
    a. Texture Model
    b. Appearance Model (APM)
    c. Active Shape Model (ASM)
2. Experiment with the ASM and the APM for face verification.

1.2 Contributions

The following are the contributions made towards the goal of this thesis

1. Wrote code for building Texture model
2. Wrote code for building Appearance model
3. Contributed code for building Active Shape Model
4. Used the repository code for doing verification experiments with ASMs and APMs

1.3 Layout

The remaining chapters of this thesis are organized in the following way:

Chapter 2 discusses the literature on some background concepts needed for proper understanding of the work done in the thesis, as well as some of the mathematical techniques used and different approaches to face detection and face recognition.

Chapter 3 describes the process involved in building the active shape model.

Chapter 4 describes building appearance models and how to use an ASM to interpret an image.

Chapter 5 explains our face verification experiment framework.

Chapter 6 describes the experiments performed and discusses the results obtained.

Chapter 7 summarizes the achievements of the thesis, the challenges faced and future work.

CHATER 2

BACKGROUND AND LITERATURE REVIEW

This chapter gives a literature study and background on some of the concepts and the mathematical techniques used in this thesis. We also identified some of the different approaches to face recognition.

2.1 Statistical Models

Statistical models [6] are models built from analyzing the appearance of a set of labeled examples of a class of object, such that the model captures the plausible variation in the image structure. A new image can be interpreted by finding the best plausible match of the model to the image data.

The idea is given sufficient training examples; we can capture the variations in a training set made up of instances of the class to be modeled. A statistical model is the model that represents these variations. An example is the shape model in which the set of examples are shapes represented by a number of points called landmarks consistent across all examples.

Statistical Models have the form,

$$x = \bar{x} + Pb \quad (2.1)$$

where x is a vector representing an instance of the object, $\bar{x}$ is the mean of the set of training examples, P is an orthogonal matrix whose columns are unit vectors along the principal modes of variation and b is a vector of the model parameters. New examples can be generated which are similar to those in the training set by varying the model parameters (within some constraints).

Examples of statistical models include statistical models of shape which model the variation due to shape, statistical models of texture which model variation due to grey-level intensity of the set of training examples and appearance models which combine the model of shape and texture.

2.2 Active Shape Models

Active Shape Models, ASM were proposed by Cootes and his colleagues in a series of papers [6-10]. In these papers, they proposed methods for building the models and searching with them. The ASM manipulates the statistical shape model to locate and fit an object of interest in an image.

The ASM has been extensively applied in tracking, and to modeling faces for identification or verification. Baumberg and Hogg [13] used a modified ASM to track people walking and Edwards et. al, [14-16] have applied ASMs for modeling and tracking faces. Other examples of applications of the ASM can be found in [2].

Since the original ASM, there have been many proposed variations. We mention a few here. Rogers and Graham [17] used robust least-square techniques to minimize the residuals between the model shape and the suggested shape instead of the standard least-squares used in basic ASM to address the susceptibility of the standard technique to outliers.

Al-Zubi [18,19] proposed the Active Shape Structural model. It combined ASMs with a model of the structural aspects of the object under consideration.

Van Ginneken et. al. [20] used a k-nearest neighbor classifier instead of the standard ASM profile model search based on Mahalanobis distance.

2.3 Active Appearance Models

Active Appearance Models (AAMs) were also developed by Cootes, and his colleagues [5]. AAMs use the appearance model which combines the shape model and texture model into a single model of appearance. It then treats the problem of interpreting a new image (that is, of locating parameters of the model which can generate a close enough synthetic image of the object of interest) as an optimization problem in which the difference between the query image and the image synthesized by the model is minimized.

The optimization is made faster by pre-learning the relationship between this difference (the residue) and the change in the model parameters. This relationship is learnt during the training

process by systematically displacing each parameter from the optimal value by known amounts and computing the residue.

More details on AAM, its application and variations can be found in [2, 21].

2.4 Differences between AAM and ASM

Cootes [22] describes the difference between the AAM and ASM in details. We present a brief outline here.

ASMs use image texture in small regions around the landmarks points while AAMs use the texture across the entire object.

ASMs search around the current position of the landmarks, usually along a profile normal to the boundary while AAM searches in regions directly under the current shape hence it has a larger capture range.

AAMs require fewer landmark points for training than ASMs.

The paper [22] concludes that the ASM is faster and achieves more accurate feature point location than the AAM though since the AAM explicitly minimizes texture errors it gives a better match to the image texture.

2.5 Principal Component Analysis

Principal Component Analysis, PCA [49] is a linear transformation that transforms data to a new coordinate system such that the new set of variables called the principal components are linear uncorrelated functions of the original variables and the variance of the principal components decreases with the number of the principal component.

In essence it maps data from a high dimensional space to a lower dimensional sub-space such that the number of principal components is less than or equal to the dimension of the original data. See Appendix A for steps in computing the PCA of a multivariate dataset.

The texture vectors encountered in later sections are very high dimensional data and as such computing the covariance using step 2 of appendix A is very computationally expensive.

Another technique for computing the PCA of a multivariate data known as the Snapshot method [48] is in appendix B. This is the procedure followed in this thesis for calculating the principal components of the face image grey-level data.

PCA has been used on complex data such as faces [23, 33-34]; human motion [24], shape outlines [7]. The main idea of PCA for face images as dealt with in this thesis is to find the eigenvectors that best accounts for the variations of face images in the entire image space.

The statistical models discussed here and represented in general form by Eqn (2.1) are essentially PCA models

2.6 Linear Discriminant Analysis

Linear Discriminant Analysis, LDA or Canonical Analysis [25] is a statistical technique used on multivariate data sets which contain multiple data classes. Like PCA, it is a linear transformation which seeks to perform dimensionality reduction while preserving as much of the class separation as possible.

Unlike PCA which does not pay attention to the underlying data, LDA takes into consideration the within-classes scatter and between-classes scatter and compute the transformation that maximizes the ratio of between-class and within-class scatter. An approach for performing LDA on multivariate data is described in [15].

In terms of which is better, PCA can outperform LDA when the training set is small [27] but when there is large enough data properly representing each class, or when lighting variations are to be handled LDA tends to be better [27, 28]. LDA has been used for face recognition [28, 29, and 31] and gait classification [24].

2.7 Distance and Similarity Measures

For our experiments, once the face images have been interpreted using parameters from our model, there is the task of determining how similar the query face is to the template image in the database and we handle this using distance and similarity measures.

We experimented with the four most popular distance measures in this work.

2.7.1 Mahalanobis Distance

Our aim for the Mahalanobis distance is to measure the distance between the parameters used to interpret a face image and the parameters of images stored in the database used to identify an individual (class).

Since it makes use of the covariance information, it is able to suppress the effect of within class variation due to pose, lighting, expression etc. while enhancing the effect of inter-class variation which explains the identity.

The Mahalanobis distance of an interpreted probe image with appearance parameters, c is given by:

$$d_M(c) = \sqrt{(c-\mu)^T S^{-1}(c-\mu)} \quad (2.2)$$

S is the common within-class covariance matrix for all the training examples used in building the database class models and μ is the mean of the extracted parameters of template image for that class.

The use of the common within-class covariance matrix for all the training images used for building the database is because of sufficient training example for each image class.

Mahalanobis distance was used with face images in [46, 47]. The claimed client is accepted if the computed distance is less than a given threshold otherwise it is rejected.

2.7.2 Euclidean Distance

Euclidean Distance or L2 Norm is the geometric distance between two points in the multidimensional Euclidean space. The Euclidean distance [26] between the sample projection x and the projection of the kth client mean μ_k is given by:

$$d_E(x, \mu_k) = \sqrt{(x - \mu_k)^T (x - \mu_k)} \quad (2.3)$$

Euclidean distance just like Mahalanobis distance can be used as distance metric, hence for its application in face verification, the claimed client identity is accepted if the computed distance is less than the threshold otherwise it is rejected.

Euclidean distance was used as distance metric in [23] for face recognition and in [35] for face verification.

2.7.3 Manhattan Distance

The Manhattan Distance measure also known as the L1-distance, between two multivariate datasets is the sum of the distances along each dimension. Mathematically, the Manhattan distance between two n-dimensional vectors is given by:

$$d_T = \sum_{i=1}^{n} |x_i - y_i| \quad (2.4)$$

Manhattan Distance has been used for face recognition in [47].

2.7.4 Normalized Correlation

Unlike Euclidean and Mahalanobis distance, Normalized Correlation score measures similarity. It can be computed [23] as shown below:

$$d_C(x, \mu_k) = \frac{x^T \mu_k}{|x||\mu_k|} \quad (2.5)$$

For its application to face verification, the claimed identity is accepted if the correlation score exceeds the specified threshold. Normalized correlation was used with faces in [36, 37]. Since

normalized correlation measures similarity, by negating the value gotten it becomes a distance measure.

2.8 Face Recognition

Research in face recognition started in the early 1960s and has been active since then. [3]. There have been several reported approaches to face recognition. A thorough review is beyond the scope of this thesis and can be seen in [3, 51].

Face recognition has been categorized into two groups [11, 12], Template based methods and geometric based methods. Template based approaches compare the input image with a set of templates built using statistical tools like support vector machines [52], principal component analysis [23], and linear discriminant analysis [28]. Geometric based methods also called feature based methods [53] analyses the local facial features and their geometric relationships. The approach followed in this thesis falls under the template based methods.

Another categorization of face recognition algorithm given in [51] is based on the method of representation of the face. Face recognition methods based on appearance based algorithms represent a face as a vector of intensity values and then derive a feature space from the image distribution using statistical techniques while Model based face recognition algorithms try to model a human face by learning from a set of training examples, and fits the model to a new sample to obtain parameters of the model used to represent the face. Morphable models are examples of this approach. The work in this thesis used the model based method.

Face recognition is very broad we have only noted some the different approaches taken in literature for the purpose of making it clear where our approach fits in.

CHATER 3

ACTIVE SHAPE MODEL

This chapter describes the basic Active shape model, ASM proposed by T.F Cootes [1]. The approach taken is to describe practically the ideas and steps required in building the Shape model and the Active shape model. A more detailed description can be found in the original papers published on this subject [1, 6-10].

3.1 Landmarked Points

The statistical model of shape models the variation in the shape of the objects in the training set. In order to represent the shape, the objects need to be labeled by a set of points, called landmarks. The choices of these landmarked points are important and they must be consistently located across all the images. Figure 3.1 below shows examples of two face images with 76 landmark points each.

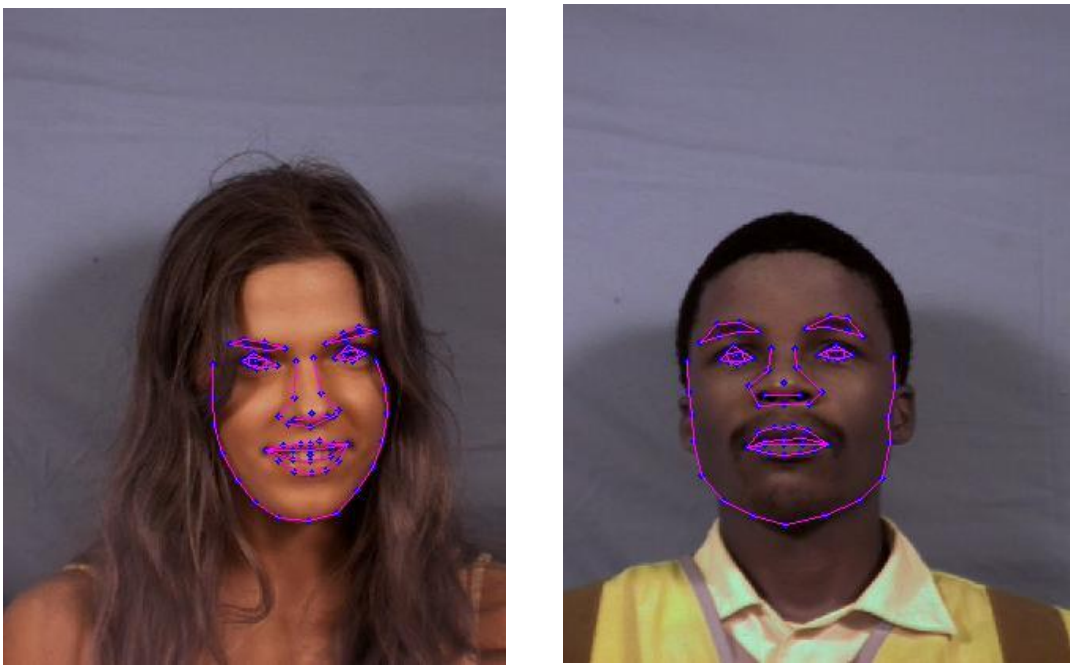


Figure 3.1: Examples of face image annotated with landmarks

3.2 Aligning the Points

Given a set of training images with landmarked points, the first step in modeling the shape is to 'align' the landmarked points. The alignment process, makes each of the landmarked point a 'shape'.

This set of landmarked points needs to be aligned into the same coordinate axis removing the effect of scaling, translation and rotation, for it to become a 'shape'. For our purpose, a shape is defined in [38] as all the geometrical information that remains when location, scale and rotational effects are filtered out from an object. Figure 3.2 below describes the procedure for aligning a set of points. The algorithm described is based on Procrutes Analysis [39]. Figures 3.3(a) and (b) show the effect of alignment on 200 sets of points.

Input: Unaligned landmark points

1. Select the first set of points as the reference shape.
2. Align all points to the reference shape
3. Loop
4. Calculate the mean shape
5. Align all point to the mean shape
6. End loop when mean shape does not change or the change is less than epsilon

Output: Aligned shapes

Figure 3.2: Procedure for Aligning sets of points

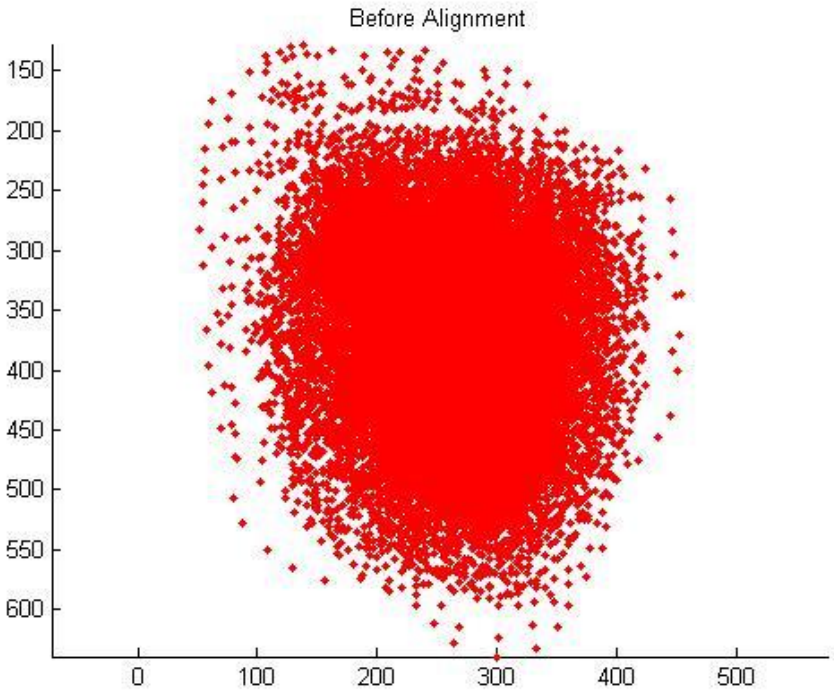


Figure 3.3a: Before Procrutes alignment of 200 sets of points

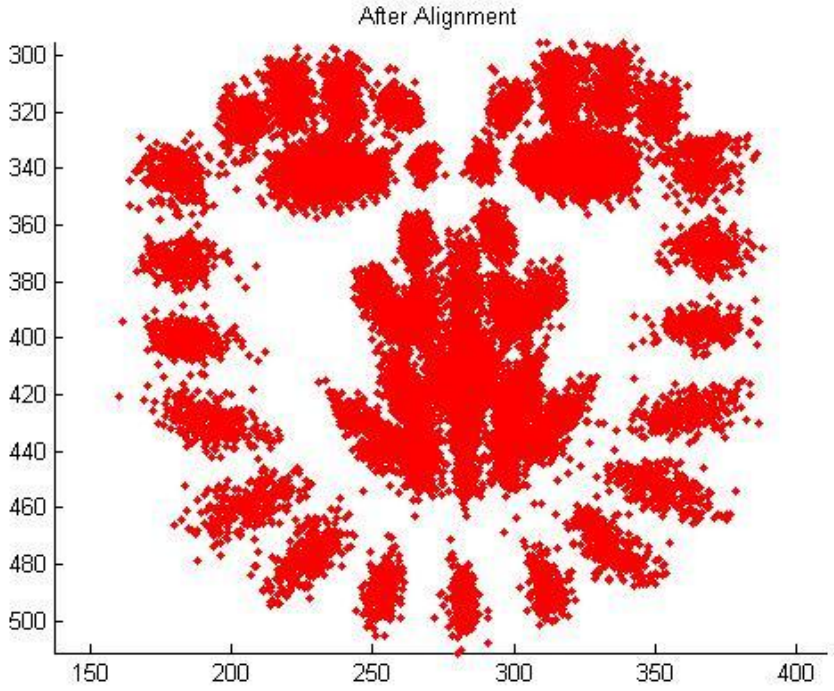


Figure 3.3b: After Procrutes alignment of 200 sets of points

After the alignment, the aligned points can now be represented as a shape vector.

4.3 Modeling Shape Variation

In order to be able to generate an instance of a shape that is representative of what can be found in the training set, the model of how the shape varies with respect to the mean shape is captured.

A linear model can be obtained using the Principal Component Algorithm, PCA:

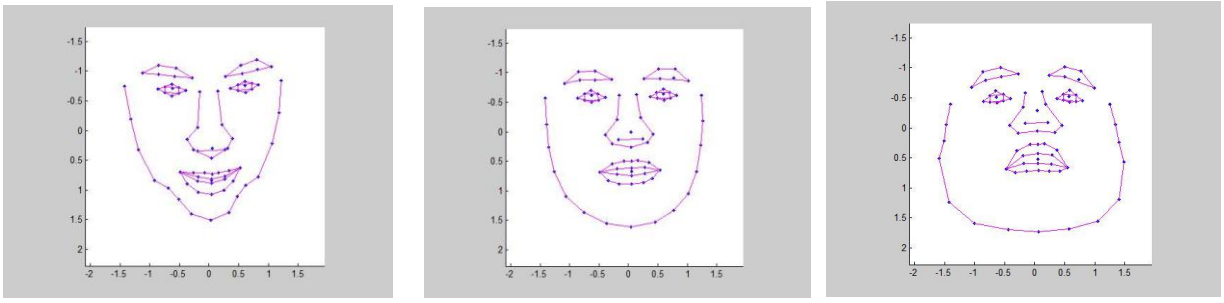
$$x = \bar{x} + P_s b_s \quad (3.1)$$

where P_s is a set of eigenvectors each with a respective eigenvalue showing the magnitude of variation from the mean shape and b_s is a set of shape parameters.

By varying the elements of b_s, we can vary the shape, x, using equation (3.1) and by limiting the shape parameters, b_i to fall between $\pm 3\sqrt{lambda_i}$, we ensure that the shapes generated are similar to those in the training set. Figure 3.4 shows the effect of varying the first three modes of a face model constructed from 300 images from the Milborrow University of Cape Town (MUCT) dataset.

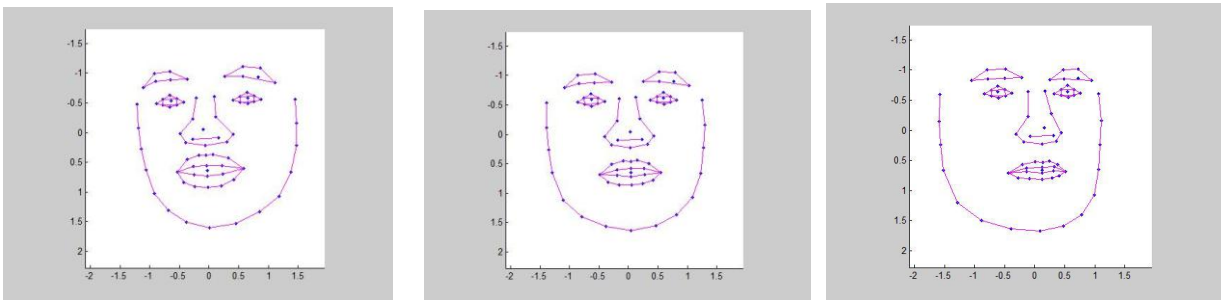
The shape model represents the allowable shape domain.

Mode 1



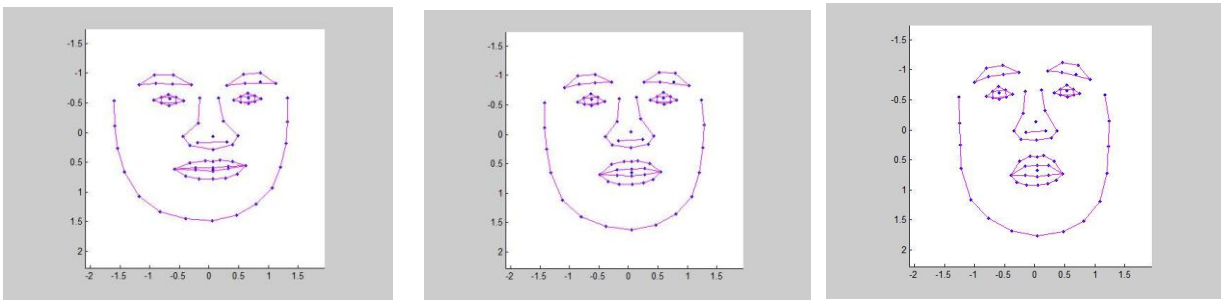
$b_1 = -3\sqrt{lambda_i}$ $b_1 = 0\ (mean)$ $b_1 = +3\sqrt{lambda_i}$

Mode 2



$b_2 = -3\sqrt{lambda_i}$ $b_2 = 0\ (mean)$ $b_2 = +3\sqrt{lambda_i}$

Mode 3



$b_3 = -3\sqrt{lambda_i}$ $b_3 = 0\ (mean)$ $b_3 = +3\sqrt{lambda_i}$

Figure 3.4: Effects of varying the first three face model shape parameters in turn between ± 3 s.d.

3.4 Modeling Local Structure

During ASM search, the region around each landmarked point is examined to locate the best nearby match for the point. The model determines what the best nearby match is by learning from the training set what to look for in the target image.

One method of doing this is to build a statistical model of the image structure around the landmark points. Figure 3.5 describes the procedure to follow in building the statistical model of the grey level profiles about each landmark point.

Given a set of labeled training examples, we wish to sample along the profile k pixels on either side of each landmark for all the examples.

For each image in the training set:
    For each landmark point:
        Computes the normal to the landmark point
        Obtain k sample points along the normal either side of the landmark
        Interpolate and get the exact values on the image
        Normalize the sample by dividing through by the root mean square of …
            … each points to have zero mean and unit standard deviation

Calculate the mean and covariance of the grey level profile about each point. This gives the parameters for a multivariate Gaussian model of gray level profile about the point.

Figure 3.5: Building the statistical model of the grey level profiles about each landmark

3.5 Multi-resolution ASM

The ASM tries to fit an instance of the shape model to an image given an initial estimate. This initial estimate can be provided manually or automatically using a detector (face, eye etc.).

The shape of the object in the object-centered coordinate frame can be obtained from the shape model parameters, b_s, then by defining the position, orientation and scale; an instance of the instance of the model in the image frame can be created. (See equation 3.2)

$$x = T_{X_t,Y_t,s,\theta}(\bar{x} + P_s b_s) \qquad (3.2)$$

where $T_{X_t,Y_t,s,\theta}$ performs a rotation by θ, a scaling by s, and a translation by (X_t, Y_t).

The ASM examines the region of the image around each point (in practice, along profiles normal to the model boundary through each point) either looking for the model boundary corresponding to the strongest edge along the profile or looking for the best match to the neighborhood model structure learnt during the training (as shown in figure 3.5).

Since landmarks are not always placed on the strongest edges (as in the case of the face images used in this thesis) we made use of the model of the local structure learnt during the training.

During search, a profile of the current points are sampled m pixels on either side, where $m > k$ and then the quality of fit of the corresponding grey-level model at each of the $2(m - k) + 1$ possible positions along the profile, is calculated using Equation (3.3) and the match with the lowest value of $\boldsymbol{f}(\boldsymbol{g_s})$ is chosen.

$$\boldsymbol{f}(\boldsymbol{g_s}) = (\boldsymbol{g_s} - \bar{\boldsymbol{g}})^T \boldsymbol{S_g^{-1}} (\boldsymbol{g_s} - \bar{\boldsymbol{g}}) \quad (3.3)$$

The quality of fit is the Mahalanobis distance of the sample from the model mean and it is linearly related to the log of the probability that the sample is drawn from the distribution. [2].

Multi-resolution ASM search is more efficient and robust. It makes use of an image pyramid and the ASM is trained at each level of the pyramid. During search, it starts at a coarse image (at lower resolution) and refines the location in higher resolutions of the image. It makes the algorithm faster as larger jumps can be made at the low resolutions and the model converges to a good solution that only requires minor adjustment at higher resolutions.

Figure 3.6 describes the ASM algorithm implemented in this thesis. The Multi-resolution ASM uses the same algorithm at each level but the result from the previous level is used to provide a better starting approximation for the new level.

Begin Loop
    For each landmark,
        Calculate the normal to the landmark point
        Sample m points along the normal
        Use interpolation to obtain the actual image value at the sampled points
        Check the quality of fit of the corresponding grey-level model at each
            of the $2(\mathrm{m} - \mathrm{k}) + 1$ possible positions along the profile
        Choose the point, y with neighborhood profile having the lowest quality of fit.
    End Loop (for each landmark)
    Initialize $b = 0$
    Loop (Find model instance to best fit the new found points)
        Calculate the model point positions using ($x = \bar{x} + Pb$)
        Calculate the transform parameters from the model frame to the image frame
        Transform the current image point into the model frame
        Update the model parameters
        Apply constraint to the parameter b to ensure plausible shape
    Until (convergence)
    Transform the points from the model frame to the image frame

End loop when converged

In practice, convergence is implied after a specified number of iterations.

Figure 3.6: ASM algorithm

CHATER 4

APPEARANCE MODEL

This chapter briefly describes the Appearance model as implemented in this thesis and explains how this is indirectly manipulated by the Active Shape Model to produce parameters that can synthesize a complete image of the object of interest.

The Shape alone does not provide a complete image of an object of interest. In order to have parameters that are sufficient to represent an object, we need to capture both the variation in its shape and its texture, which is the pattern of intensity across the region of the object.

An Appearance model is the model that combines the model of the shape variation and the model of the texture variation in a shape normalized frame. It therefore is able to capture all the information about the object of interest.

4.1 Modeling the Texture variation

In order to model the variation due to texture, all spurious texture variation due to shape and lighting needs to be removed.

To remove variations due to shape, the images are warped into a common shape, e.g. the mean shape. The procedure to do this makes use of the Delaunay triangulation and a continuous valued function that maps a pixel from the image control points to the mean shape. This gives the shape-free patch. (See figure 4.1).

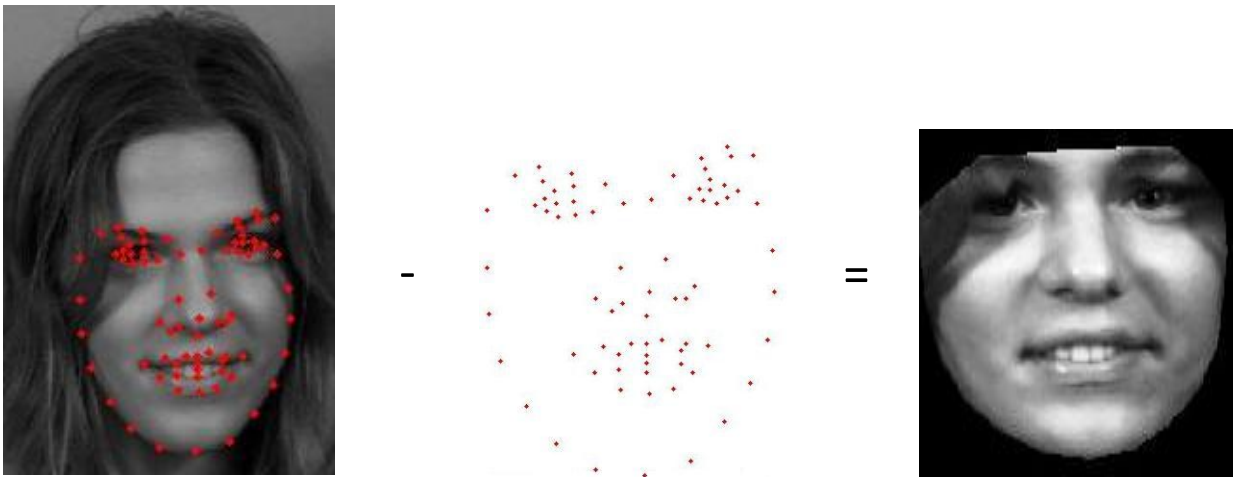


Figure 4.1: Example image, set of points and mean-shape image patch (shape-free patch)

The grey levels of the shape-free patch can now be converted into the texture vector and as it is always advisable when dealing with statistical data, the texture vector is normalized to zero mean and unit standard deviation.

A global normalization using a scaling, α and offset parameter, β is applied using equation 4.1 to reduce the effect of global changes due to lighting.

$$g = \frac{(g_{im} - \beta 1)}{\alpha} \quad (4.1)$$

where g_{im} is the texture vector of the mean shape patch, $\bar{g}$ is the mean normalized grey-level vector, $\alpha \; and \; \beta$ are given by equation 4.2 and n is the number of element in the vectors.

$$\alpha = g_{im} . \bar{g} \; , \beta = \frac{g_{im} . 1}{n} \quad (4.2)$$

As with the sets of shapes, in order to capture the variation due to the grey levels across the training examples, we perform PCA to obtain the eigenvectors and the eigenvalues similar to [23].

Due to the high dimension of the data when compared with the number of examples, we followed the snapshot method for computing the PCA. The result of performing PCA on the texture data is a linear model:

$$g = \bar{g} + P_g b_g \quad (4.3)$$

where P_g is a set of orthogonal modes of the texture variation and b_g is a set of grey-level parameters.

Mode 1

$b_1 = -3\sqrt{lambda_i}$ $b_1 = 0\ (mean)$ $b_1 = +3\sqrt{lambda_i}$

Mode 2

$b_2 = -3\sqrt{lambda_i}$ $b_2 = 0\ (mean)$ $b_2 = +3\sqrt{lambda_i}$

Figure 4.2: First two modes of texture only variation (± 3 s.d).

4.2 Combining the shape and texture model

In order to capture the combined variation in both shape and texture, the parameters of both models are concatenated as in equation 4.4 and a further PCA is applied to the data to remove any correlation between the shape and the texture variations and make the model more compact [38].

$$b = \begin{pmatrix} W_s b_s \\ b_g \end{pmatrix} \quad (4.4)$$

where W_s is a diagonal matrix of weights for each shape parameter which takes care of the difference in units between the shape and the texture model.

A simple method for calculating W_s described in [2] is to set $W_s = rI$ where r^2 is the ratio of the total intensity variation to the total shape variation in the normalized frames.

The result of PCA performed on the data from equation 4.4 is

$$b = P_c c \quad (4.5)$$

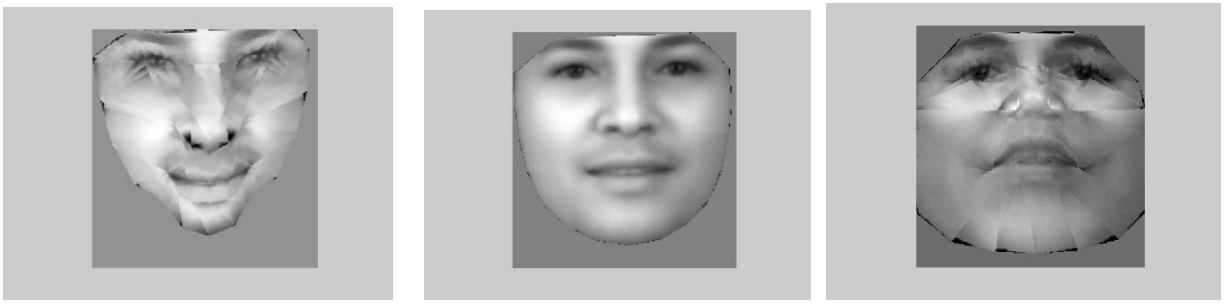
where P_c is a set of eigenvectors and c is a vector of appearance parameters controlling both the shape and texture of the model. Note also that the mean of the data is zero. Figure 4.3 shows the first two modes of appearance (shape and texture) variation ± 3 s.d.

The shape and texture can be expressed directly as functions of c:

$$x = \bar{x} + P_s W_s^{-1} P_{cs} c \ , \ g = \bar{g} + P_g P_{cg} c \qquad (4.6)$$

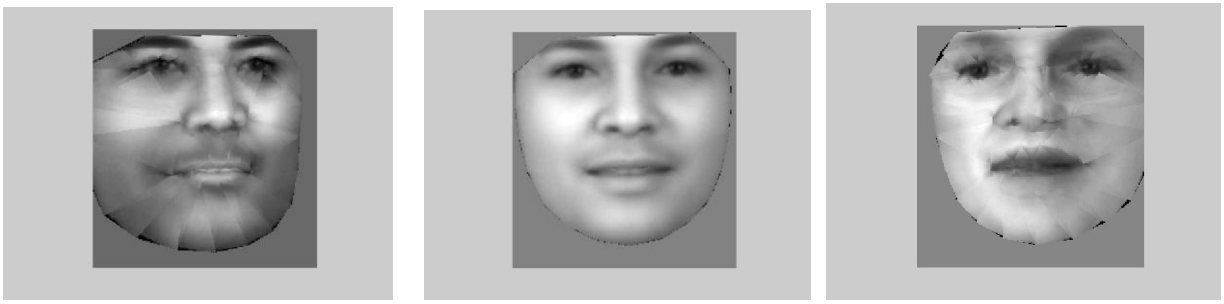
Although the shape and texture models were built separately, as would be seen in the papers by the inventors [5, 6], the shape and grey level variation can be modeled in a single process and only a single PCA performed to get the appearance model. However, building separate models is more popular this way because of its history, as the AAM was like an offspring of the ASM.

Mode 1



$b_1 = -3\sqrt{lambda_i}$ $b_1 = 0\ (mean)$ $b_1 = +3\sqrt{lambda_i}$

Mode 2



$b_2 = -3\sqrt{lambda_i}$ $b_2 = 0\ (mean)$ $b_2 = +3\sqrt{lambda_i}$

Figure 4.3: First two modes of appearance (shape and texture) variation ± 3 s.d.

4.4 Fitting the Appearance Models Using Active Shape Model

Appearance models are capable of synthesizing a new instance of the object of interest as close as possible to the target object. In order to interpret an image using the model, we need to find the set of parameters that best match the model to the image.

Active shape models as explained earlier (see section 3.5) make use of the constraints provided by the shape model to locate a set of points on a new image that give the shape pf the structure of interest. In essence it only makes use of the shape information and information of the image structure near the landmarks (grey-level profile model).

The Active Appearance Model on the other hand makes use of the constraint of the appearance model to find the model parameters capable of generating a synthetic image as close as possible to the target image.

In this thesis, we used the Active Shape model to indirectly find such parameters. The approach is similar to that taken by [50]. To interpret a new image using the ASM, we calculated the shape (represented by the set of points) using equation (3.1) and the model parameters returned by the ASM search. We then warped the image into the space of the mean shape of the appearance model to obtain the shape normalized image.

Using the texture vector from the shape normalized image, we obtained parameters from the texture model required to generate that texture using:

$$b_g = {P_g}^T(g_{im} - \bar{g}) \qquad (4.11)$$

We then combine the shape model parameters,b_s, and the texture model parameters, b_g, as in equation (4.4). The appearance model parameters can be obtained using equation (4.12)

$$c = {P_c}^T b \qquad (4.12)$$

c, can be used to represent the image for the face verification task done in this thesis.

Figure 4.4 gives a graphical illustration of the process described above.

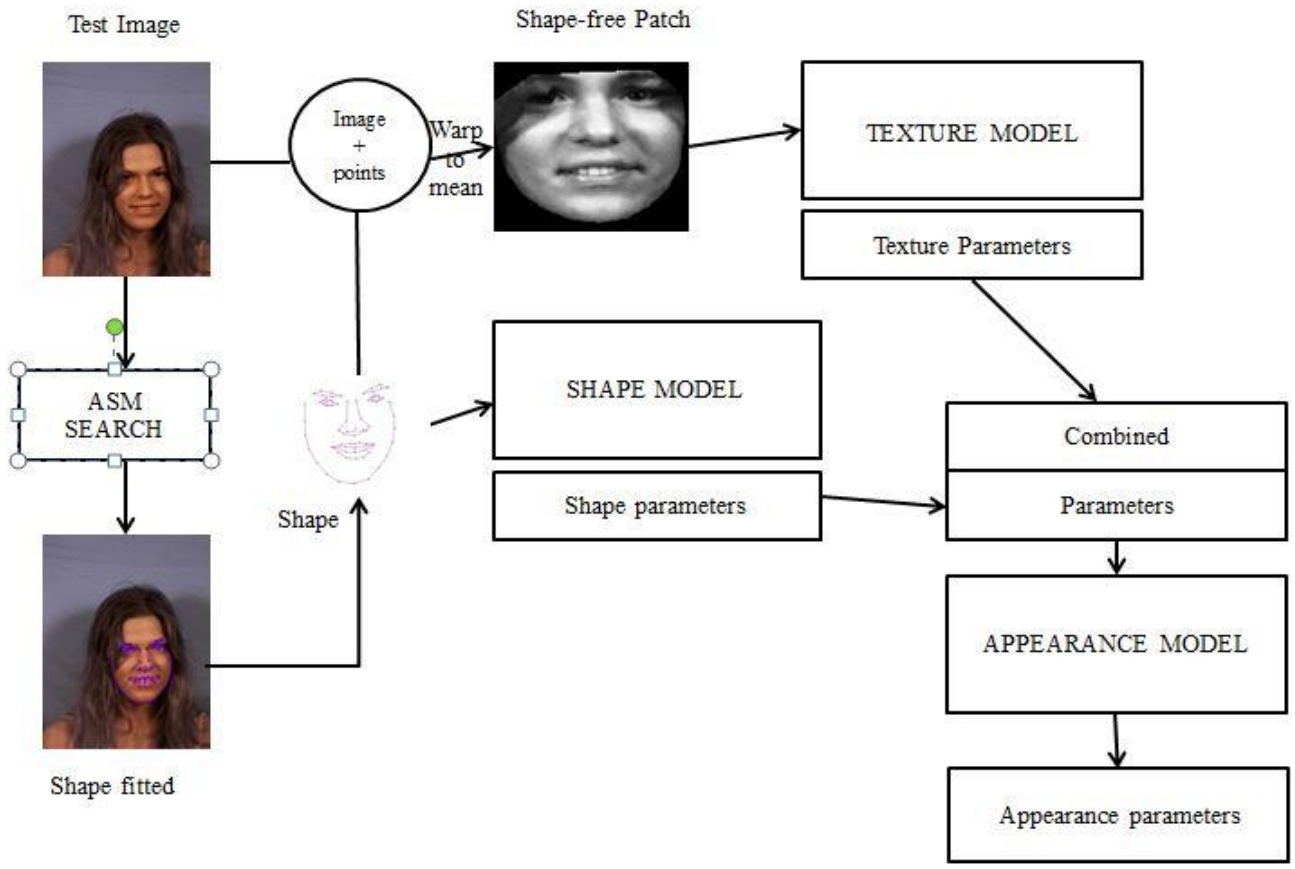


Figure 4.4: Graphical Illustration of face image interpretation using an ASM and an Appearance model

CHAPTER 5

FACE RECOGNITION

Face recognition for humans is a very trivial task, something that is done most times without conscious thought but for computers it is a very difficult task. Some of the challenges for computers include coping with variations in facial appearance due to facial expression, lighting, pose and age of the person.

Face recognition is a general term that includes both face verification and face identification. Face identification is the task of identifying a face from several already known faces. In a very general way it says, "Give me an image of a face, and I would tell whom it belongs to". The ability of the system to recognize the face depends on the face being previously known to it. A face identification system should either identify the individual or label the individual as unknown.

On the other hand, face verification is the task of confirming the identity of a person. Precisely it says, "Give me an image of a face, state the identity of the person and I would tell you if you are correct or wrong." It may also be known as face authentication. A face verification system should either confirm a client or deny an imposter.

Face recognition in general has many important applications for example in security for video surveillance and access control or authentication and also in robotics for human computer interactions.

Our experiment would focus on face verification and this chapter explains our experiment's design, and discusses some of the parameters required for the experiments.

5.1 The Face Verification Framework

The face verification tasks deals with comparing a probe/query face with a template in a database and either classifying the face as a client or an impostor. The diagram below gives a block diagram of the developed face verification system used for the experiment.

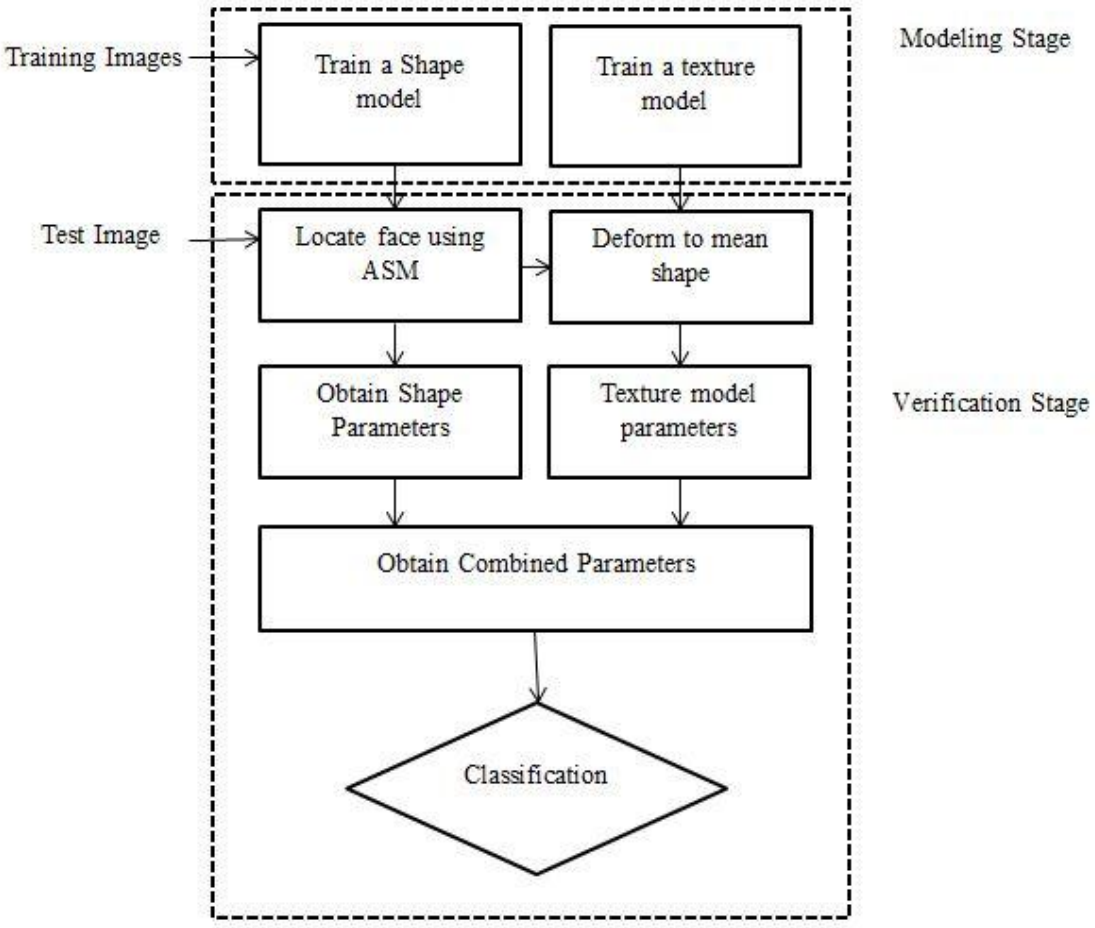


Figure 5.1: Block diagram of the face verification system

5.2 Evaluation of the Face Verification System

We would be adopting the measures for evaluating a face verification system described by Fabien Cardinaux in [40].

Performance Measure

The performance of a face verification system is measured in terms of the False Acceptance Rate, FAR (which occurs when the system accepts an impostor) and False Rejection Rate, FRR (which occurs when the system rejects a client).

$$FAR = \frac{number\ of\ FAs}{number\ of\ imposter\ accesses}$$

$$FRR = \frac{number\ of\ FRs}{number\ of\ client\ accesses}$$

The Half Total Error, HTER, is normally used to aid the interpretation of the performance:

$$HTER = \frac{(FAR + FRR)}{2}$$

The Equal Error Rate, ERR, is when $FAR = FRR$. It gives a threshold independent performance measure; the lower the EER, the better the performance of the verification system.

Trade-off between FAR and FRR

FAR and FRR are related. In some situations, it is more important to have much higher FRR than FAR for example if the Verification system secures access to highly sensitive information (e.g. National Intelligence data). The decision threshold is used to vary the trade-off between FAR and FRR.

The Weighted Error Rate, WER, is used to evaluate the system for a specific situation

$$WER(\tau^*) = \omega FAR(\tau^*) + (1 - \omega)FRR(\tau^*)$$

Where w belongs [0,1] is set for specific situations and τ^* , is the threshold that minimizes the WER for a given w .

Graphical Representation of Performance

There are a number of graphical representations of the system performance used in literature. In this thesis we used the Receiver Operating Characteristics (ROC) curve. The ROC curve plots the relationship of false rejection rate and false acceptance rate as a function of threshold τ.

5.3 Decision Threshold

The face verification step can be seen as a classification task. This task assigns a label of either impostor or client to the query face depending on the value of a particular threshold. There are two sources for which the threshold could be determined.

1. Client Specific threshold, in which each client in the database has a specific threshold
2. Global threshold, in which a single threshold is common for all client.

The problem of using the client specific threshold is it requires a large number of clients for querying the system.

A different approach for obtaining the client specific threshold using an imposter test was adopted in [42]. The client specific threshold can be calculated from the mean, μ_k and the standard deviation, σ_k of the imposter distance for the specific client and the global threshold, τ, using equation 5.1:

$$\tau_k = \mu_k + \tau.\sigma_k \qquad (5.1)$$

Nevertheless many face verification papers in literature use the global threshold. [40]. In this work, we would also make use of global threshold.

Three different global thresholds τ would be computed like in [41]:

$$\tau_{FAR=0} = \arg min_{\tau}(FRR|FAR = 0)$$

$$\tau_{FAR=FRR} = (\tau|FAR = FRR) \tag{5.2}$$

$$\tau_{FRR=0} = \arg min_{\tau}(FAR|FRR = 0)$$

The experiments carried out in this thesis use all three thresholds. The idea of using these three thresholds is simply to get more data for our comparative experiment to be more robust.

CHAPTER 6

EXPERIMENTS

This chapter describes the experimental setup and then explains in details the experiments conducted and the results obtained.

6.1 Database

There are several databases of faces images available [43]; each built under different conditions. For our experiments, we chose the MUCT database [44] because of the number of landmarks and the fact that it is freely available

6.1.1 MUCT Database

The MUCT dataset consist of 3755 face images with 76 manual landmarks taken under different lighting conditions and camera views. A more detailed description of the database is found in [44]. For our experiments, we only made use of camera views a, b, d, and e; (b was only used for testing).

Figure 6.1 shows example images from the MUCT dataset with the camera views used in this work and under different lighting conditions.

Figure 6.1: Example images from the MUCT database

6.1.2 Model Building

To build the face model for our experiment, we made use of 300 images picked randomly from our subset of the MUCT Database.

All the models were built with 100% of the variance retained. The ASM search was run at 5 resolutions with each resolution only making use of 95%, 80%, 40%, 20% and 10% variance with the 10% variance used at the lowest resolution and the variation increasing with resolution.

The shape model had 149 model parameters at 100% variance retained, the texture model had 300 parameters, and each of the shape-free patches was of size 178 by 175 pixels. The appearance model has 282 parameters.

6.1.3 Partitioning Data for Verification Experiments

For our verification experiments, we partitioned our database into three sets (Training set, Evaluation Set and Test set). The training set contained 20 clients each having 2 images. The

evaluation set contained 20 imposters and 3 face images of the 20 clients. The test set contained a different set of 50 imposters, and between 3 and 7 other face images of each of the 20 clients.

The challenge of having very few client images to use for face verification experiment is not new to this work. Other authors, [45, 35], also had to perform similar experiments with small number of client images for the training, testing and evaluation sets.

The client dataset were mirrored to increase the number of images in the test and evaluation set. The evaluation set then had 6 client images and the test set between 6 and 14 client images. The training set was not mirrored. This approach to increase the number of images in the client dataset has been previously used in [29, 40].

The training set was used to build database client models; the evaluation set was used to establish the operating threshold, and the test set was used to confirm the verification rate on an independent dataset.

6.2 Face Verification Experiments

Our main verification task is to determine if a pair of faces (one, the query face and the other, the mean parameter from a set of template faces in the database) belongs to the same individual or not.

We carried out an experiment to evaluate the verification performance of different metrics (Mahalanobis, Euclidean, Manhattan, Normalized Correlation) using the appearance parameters to represent the faces.

6.3 Experiment: Quantitative comparison of the results of the use of four metric (Mahalanobis, Euclidean, Manhattan, Normalized Correlation), in the PCA subspace.

This experiment evaluates the appearance model parameters in the PCA subspace for verification tasks using four different metrics for the classification.

In order to perform this experiment, the operating thresholds would first be calculated on the evaluation set.

Figure 7.2, 7.3, 7.4 and 7.5 shows the ROC curves indicating the global operating thresholds selected for the Mahalanobis, Euclidean and normalized correlation metric. Table 7.1 shows the error rates.

6.3.1 Learning the Global thresholds (Mahalanobis, Euclidean, Manhattan, Normalized Correlation)

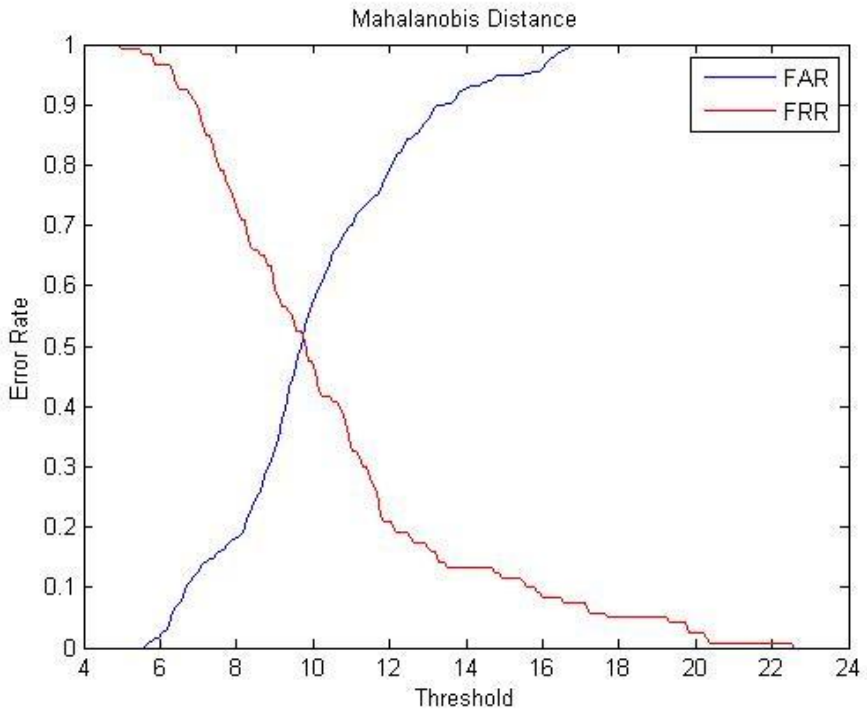


Figure 6.2: ROC curve plot of FAR and FRR on a range of thresholds using Mahalanobis distance

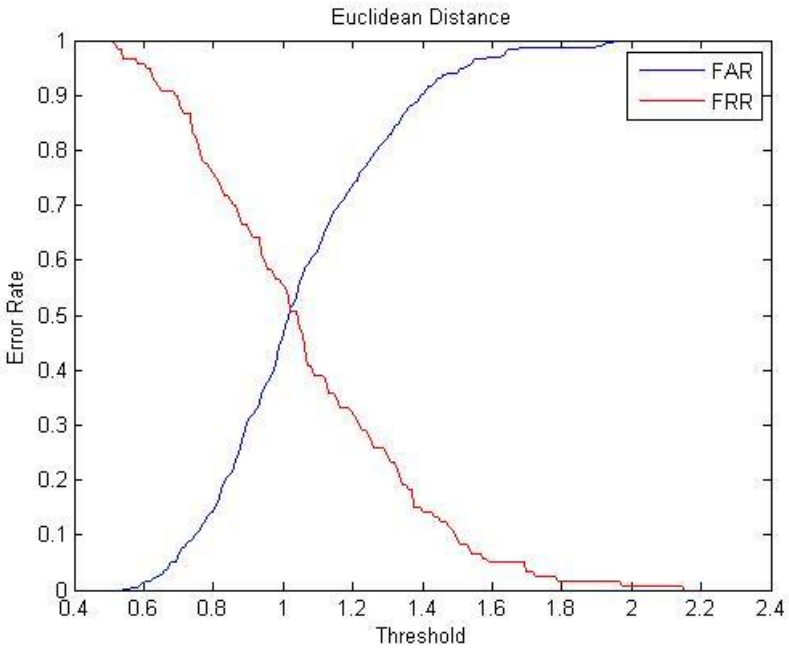


Figure 6.3: ROC curve plot of FAR and FRR on a range of thresholds using Euclidean distance

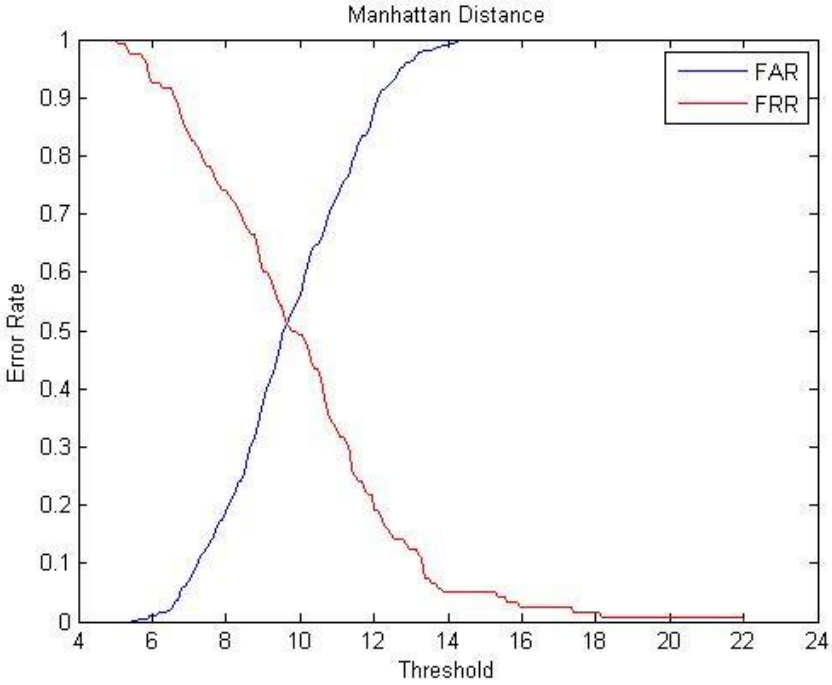


Figure 6.4: ROC curve plot of FAR and FRR on a range of thresholds using Manhattan distance

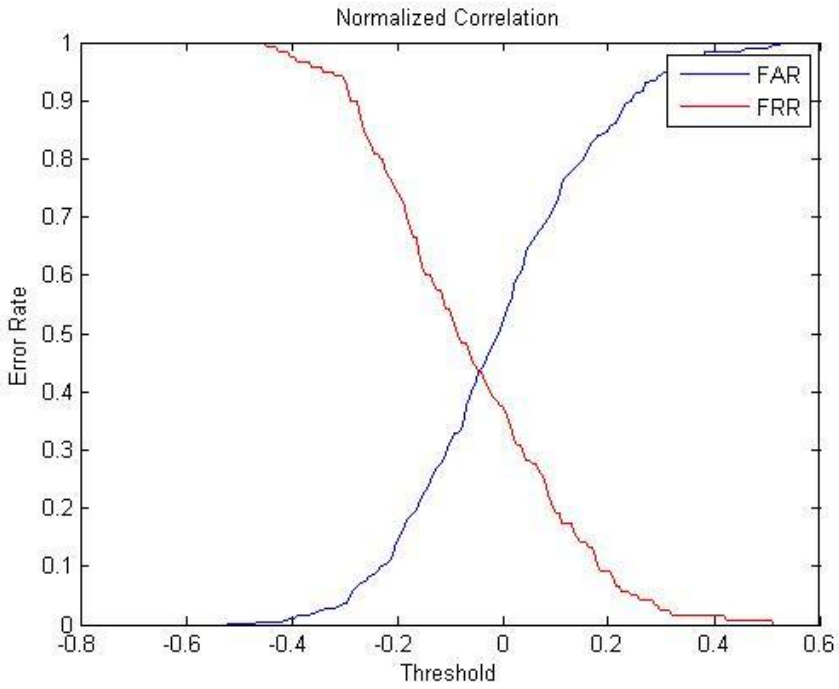


Figure 6.5: ROC curve plot of FAR and FRR on a range of thresholds using normalized correlation

Evaluation Metric	**Performance Measure on Evaluation Set**		
	FAR = FRR	FAR (FRR = 0)	FRR (FAR = 0)
Mahalanobis Distance	0.5165	1.0000	0.9833
Euclidean Distance	0.5108	1.0000	0.9833
Manhattan Distance	0.5130	1.0000	0.9750
Normalized Correlation	0.4339	0.9950	1.0000

Table 6.1: Results of global threshold for the four metric on the evaluation set (Best highlighted)

6.3.2 Error Rate Comparison (Mahalanobis, Euclidean, Manhattan, Normalized Correlation)

The computed global thresholds were then used to test the model using the test set. The results reported in Table 6.2 are the average across all the clients.

Evaluation Metric	**Performance Measure on Test Set**					
	FAR = FRR		FAR (FRR = 0)		FRR (FAR = 0)	
	FAR	FRR	FAR	FRR	FAR	FRR
Mahalanobis Distance	0.5010	0.3571	1.0000	0.0036	0.0030	0.7321
Euclidean Distance	0.4310	0.3786	0.9940	0	0.0020	0.9143
Manhattan Distance	0.4220	0.3929	1.0000	0	0.0010	0.8786
Normalized Correlation	0.4340	0.6000	0.9960	0.2607	0.0040	1.0000

Table 6.2: Performance for the four metric on test set using the global threshold (Best highlighted)

6.4 Discussions

The experiment showed that the Mahalanobis distance, Euclidean distance (L2) and Manhattan distance perform better than Normalized correlation. Nevertheless the best amongst them is not clear as the difference is performance overall is little.

We found out from literatures [47] that in the principal component subspace, Mahalanobis distance is the best followed by Euclidean distance. Also [32] states that the Mahalanobis distance measure performs the best followed in order by Manhattan distance, Euclidean distance.

We note here that the experiment made use of 2 images per class and had 20 classes. We believe that the reason for Mahalanobis distance not performing as well as expected is due to the fact that we used very few images (2) per class and the database had just 20 classes.

CHAPTER 7

CONCLUSION

Producing codes for a repository is like building a bridge, in this thesis we did our best to lay one of the pillars of that bridge. This final chapter summarizes the work carried out and makes recommendations for future works.

7.1 Summary of contributions

In this project we developed codes for the computer vision and machine learning code repository to facilitate further research in face recognition and computer animation. The algorithms implemented are results of previous research in the field and serves as basis for future research and applications. The main papers implemented in the course of this work are as follows:

1. Active shape model papers [1, 9, 10]
2. Appearance model paper [2]

Furthermore, we successfully applied the code base to creating a system for facial verification and evaluated its performance using four different performance measures.

7.2 Challenges

Some of the challenges faced in the course of this work were

1. Limited number of landmarked images available per individual.
2. Implementing the ideas in the papers was not trivial, as many of the details are not exposed in the papers.

7.3 Future Work

There are many parts towards making the repository code base useful for deeper research in the field, still undone. The following are suggestions for work to be done in the future.

1. Implement the Active Appearance model algorithm
2. Create an image database for our local context and landmark these images.

APPENDIX A

PRINCIPAL COMPONENT ANALYSIS

Given a multivariate dataset, $x_i^j \ \forall_{i=1..m,j=1..n}$ where n is the number of dimensions of the dataset and m is the number of examples.

The following steps are used in principal component analysis.

1. Compute the mean of the data,

$$\bar{x}^j = \frac{1}{m}\sum_{i=1}^{m} x_i^j$$

2. Compute the $n \times n$ covariance matrix of the data (Note that x is an $n * m$ matrix),

$$S = \frac{1}{m}\sum_{i=1}^{m}\left(x_i^j - \bar{x}^j\right)\left(x_i^j - \bar{x}^j\right)^T$$

3. Compute the eigenvectors, $\emptyset_i$ and eigenvalues, $lambda_i$ of the covariance matrix.
4. Sort the eigenvalues (and eigenvectors) so that the values are in descending order of magnitude.

APPENDIX B

PRINCIPAL COMPONENT ANALYSIS FOR HIGH DIMENSIONAL DATA

In the case were the dimensions of the data are much greater than the number of examples, the following computationally efficient method (snapshot) can be used.

1. Compute the mean of the data

$$\bar{x}^j = \frac{1}{m}\sum_{i=1}^{m} x_i^j$$

2. Compute the $m \times m$ covariance matrix of the data,

$$S = \frac{1}{m}\sum_{i=1}^{m}\left(x_i^j - \bar{x}^j\right)^T\left(x_i^j - \bar{x}^j\right)$$

3. Compute the eigenvectors, $\emptyset_i$ and eigenvalues, $lambda_i$ of the covariance matrix.

4. Sort the eigenvalues (and eigenvectors) so that the values are in descending order of magnitude.
5. Multiply each of the m eigenvectors by the data matrix,

$$\emptyset_i = X\emptyset_i \ \forall_{i=1\ to\ m}$$

   The remaining eigenvectors have zero eigenvalues.
6. Divide the eigenvectors by their norm

$$v_i = \frac{\widehat{v_i}}{\|\widehat{v_i}\|}$$

REFERENCES

1. Cootes, T. F., Taylor, C. J., Cooper, D. H., & Graham, J. (1995). Active shape models-their training and application. Computer vision and image understanding, 61(1), 38-59.
2. Cootes, T. F., & Taylor, C. J. (2004). Statistical models of appearance for computer vision. Imaging Science and Biomedical Engineering, University of Manchester, Manchester M13 9PT, UK March, 8.
3. Li, S. Z. (2011). Handbook of face recognition. Springer verlag London Limited.
4. Viola, P., & Jones, M. J. (2004). Robust real-time face detection. International journal of computer vision, 57(2), 137-154.
5. Cootes, T. F., Edwards, G. J., & Taylor, C. J. (1998). Active appearance models. In Computer Vision—ECCV'98 (pp. 484-498). Springer Berlin Heidelberg.
6. Cootes, T. F., & Taylor, C. J. (1992). Active shape models—'smart snakes'. In BMVC92 (pp. 266-275). Springer London.
7. Cootes, T. F., Taylor, C. J., Cooper, D. H., & Graham, J. (1992). Training models of shape from sets of examples. In BMVC92 (pp. 9-18). Springer London.
8. Cootes, T. F., & Taylor, C. J. (1993, September). Active shape model search using local grey-level models: A quantitative evaluation. In British Machine Vision Conference (Vol. 31, pp. 743-756).
9. Cootes, T. F., Taylor, C. J., & Lanitis, A. (1994, September). Active shape models: Evaluation of a multi-resolution method for improving image search. In Proc. British Machine Vision Conference (pp. 327-338).
10. Cootes, T. F., Taylor, C. J., Lanitis, A., Cooper, D. H., & Graham, J. (1993, May). Building and using flexible models incorporating grey-level information. In Computer Vision, 1993. Proceedings., Fourth International Conference on (pp. 242-246). IEEE.
11. Torres, L. (2004, April). Is there any hope for face recognition?. In Proc.of the 5th International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS (Vol. 21, p. 23).

12. Gross, R. Shi, J. & Cohn, J. Quo vadis face recognition? - the current state of the art in face recognition. Technical report, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, June 2001.
13. Baumberg, A., & Hogg, D. (1994). Learning flexible models from image sequences (pp. 297-308). Springer Berlin Heidelberg.
14. Edwards, G. J., Taylor, C. J., & Cootes, T. F. (1999). Improving identification performance by integrating evidence from sequences. In Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on. (Vol. 1). IEEE.
15. Edwards, G. J., Lanitis, A., Taylor, C. J., & Cootes, T. F. (1998). Statistical models of face images—Improving specificity. Image and Vision Computing, 16(3), 203-211.
16. Edwards, G. J., Taylor, C. J., & Cootes, T. F. (1998, April). Interpreting face images using active appearance models. In Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on (pp. 300-305). IEEE.
17. Rogers, M., & Graham, J. (2006). Robust active shape model search. In Computer Vision—ECCV 2002 (pp. 517-530). Springer Berlin Heidelberg.
18. Al-Zubi, S. (2004). Active shape structural model (Doctoral dissertation, Otto-von-Guericke-Universität Magdeburg, Universitätsbibliothek).
19. Al-Zubi, S., & Toennies, K. (2003, January). Generalizing the active shape model by integrating structural knowledge to recognize hand drawn sketches. In Computer Analysis of Images and Patterns (pp. 320-328). Springer Berlin Heidelberg.
20. Van Ginneken, B., Frangi, A. F., Staal, J. J., ter Haar Romeny, B. M., & Viergever, M. A. (2002). Active shape model segmentation with optimal features. medical Imaging, IEEE Transactions on, 21(8), 924-933.
21. Cootes, T. F., & Kittipanya-ngam, P. (2002, September). Comparing variations on the active appearance model algorithm. In British Machine Vision Conference (Vol. 2, p. 837).
22. Cootes, T. F., Edwards, G., & Taylor, C. J. (1999, September). Comparing active shape models with active appearance models. In British Machine Vision Conference (Vol. 1, pp. 173-183).
23. Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. Journal of cognitive neuroscience, 3(1), 71-86.

24. Huang, P. S., Harris, C. J., & Nixon, M. S. (1998). Recognising humans by gait via parametric canonical space.
25. Krzanowski, W. J., & Marriott, F. H. C. (1994). Multivariate analysis. London: Edward Arnold.
26. Jonsson, K., Kittler, J., Li, Y. P., & Matas, J. (2002). Support vector machines for face authentication. Image and Vision Computing, 20(5), 369-375.
27. Martinez, A. M., & Kak, A. C. (2001). Pca versus lda. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 23(2), 228-233.
28. Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 19(7), 711-720.
29. Etemad, K., & Chellappa, R. (1997). Discriminant analysis for recognition of human face images. JOSA A, 14(8), 1724-1733.
30. Zhao, W., Chellappa, R., Phillips, P. J., & Rosenfeld, A. (2003). Face recognition: A literature survey. Acm Computing Surveys (CSUR), 35(4), 399-458.
31. Swets, D. L., & Weng, J. J. (1996). Using discriminant eigenfeatures for image retrieval. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 18(8), 831-836.
32. Nguyen, H. (2011). Linear subspace methods in face recognition (Doctoral dissertation, University of Nottingham).
33. Sirovich, L., & Kirby, M. (1987). Low-dimensional procedure for the characterization of human faces. JOSA A, 4(3), 519-524.
34. Kirby, M., & Sirovich, L. (1990). Application of the Karhunen-Loeve procedure for the characterization of human faces. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 12(1), 103-108.
35. Marcialis, G. L., & Roli, F. (2006). Fusion of LDA and PCA for Face Verification. In Biometric Authentication (pp. 30-37). Springer Berlin Heidelberg.
36. Kittler, J., Li, Y. P., & Matas, J. (2000, September). On matching scores for LDA-based face verification. In Proceedings of British Machine Vision Conference (pp. 42-51).
37. Li, A., Shan, S., Chen, X., & Gao, W. (2009, June). Maximizing intra-individual correlations for face recognition across pose differences. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on (pp. 605-611). IEEE.

38. Stegmann, M. B. (2007). Active appearance models (Doctoral dissertation, Master's thesis, IIM, Technical University of Denmark).
39. Goodall, C. (1991). Procrustes methods in the statistical analysis of shape. Journal of the Royal Statistical Society. Series B (Methodological), 285-339.
40. Cardinaux, F. (2005). Face Authentication based on local features and generative models (Doctoral dissertation, PhD thesis, Ecole Polytechnique Fédérale de Lausanne).
41. Matas, J., Hamouz, M., Jonsson, K., Kittler, J., Li, Y., Kotropoulos, C., & Mayoraz, E. (2000).. In Pattern Recognition, 2000. Proceedings. 15th International Conference on (Vol. 4, pp. 858-863). IEEE.
42. Jonsson, K., Matas, J., Kittler, J., & Li, Y. P. (2000). Learning support vectors for face verification and recognition. In Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on (pp. 208-213). IEEE.
43. Gross, R. (2005). Face databases. In Handbook of Face Recognition (pp. 301-327). Springer New York.
44. Milborrow, S., Morkel, J., & Nicolls, F. (2010). The muct landmarked face database. Pattern Recognition Association of South Africa.
45. Kittler, J., Ghaderi, R., Windeatt, T., & Matas, J. (2001). Face verification using error correcting output codes. In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on (Vol. 1, pp. I-755). IEEE.
46. Draper, B. A., Baek, K., Bartlett, M. S., & Beveridge, J. R. (2003). Recognizing faces with PCA and ICA. Computer vision and image understanding, 91(1), 115-137.
47. Beveridge, J. R., She, K., Draper, B. A., & Givens, G. H. (2001). A nonparametric statistical comparison of principal component and linear discriminant subspaces for face recognition. In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on (Vol. 1, pp. I-535). IEEE.
48. Yambor, W. S. (2000). Analysis of PCA-based and Fisher discriminant-based image recognition algorithms (Master's thesis, Colorado State University).
49. Jolliffe, I. T. (2002). Principal component analysis. Springer verlag.

50. Lanitis, A., Taylor, C. J., Cootes, T. F., & Ahmed, T. (1995). Automatic interpretation of human faces and hand gestures using flexible models. In In International Workshop on Automatic Face-and Gesture-Recognition.
51. de Carrera, P. F. (2010). Face Recognition Algorithms.
52. Guo, G., Li, S. Z., & Chan, K. (2000). Face recognition by support vector machines. In Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on (pp. 196-201). IEEE.
53. Brunelli, R., & Poggio, T. (1993). Face recognition: Features versus templates. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 15(10), 1042-1052.