A Computerised Face Recall System Using Eigenfaces

by

Yon Aryeh Rosenthal

Submitted to the Department of Electrical Engineering
in partial fulfilment of the requirements for the degree of

Master of Science in Engineering

at the

UNIVERSITY OF CAPE TOWN

October 1998

Declaration

I, Yon Aryeh Rosenthal, declare that this dissertation is my own work. It is being submitted for the degree of Master of Science in Engineering at the University of Cape Town and it has not been submitted before for any degree or examination, at any other university.

Yon Rosenthal

Acknowledgements

I would like to thank the following people and institutions for their contributions towards this thesis.

- My supervisor, Professor Gerhard de Jager, for his guidance and help.
- Fred Nicolls, for proof reading this thesis.
- Everybody in the UCT Digital Image Processing laboratory for the stimulating, interesting and enjoyable working environment.
- John Greene, who's course initiated this face recall system.
- Colin Tredoux, who helped with the evaluation of the developed face recall system.
- Lisa da Costa, who ran the experiments.
- De Beers, for their financial support.
- The Foundation for Research and Development, for their financial support.

Abstract

This work outlines the development of a face recall system. A face recall system is a system that is used to translate a mental image of a face to a visual image that can be shown to others. Face recall systems are used primarily in law enforcement related tasks.

A literature survey on existing face recall systems shows that there exists much room for improvement to the presently available systems. The face recall system developed in this thesis addresses most of the shortcomings of the existing face recall systems that were noted in the survey.

This novel system builds up its face likeness images as a linear combination of eigenfaces. It is demonstrated that a linear combination of eigenfaces can be used to adequately reconstruct new face images, with the coefficients of the linear combination determining what the face image looks like. This system constructs a face image by searching for the coefficients that will result in a good likeness to the desired face. This search for the correct coefficients is done using an evolutionary optimisation algorithm called PBIL, with the person trying to generate the likeness face image performing the evaluation of the cost function.

The results obtained from experiments performed using this system show that in its present state it is not a viable system, but if these are interpreted as being preliminary feasibility studies the results are promising.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter explains what a face recall system is and where it is used. It also gives a brief motivation for this thesis. The chapter ends with an outline of the rest of the thesis.

1.1 Face recall systems

A face recall system is a system that is used to translate a mental image of a face to a visual image that can be shown to others. The most prevalent use of face recall systems is to aid a witness of a crime to generate a visual face image from the mental images that they have of the perpetrator(s).

The reason for having to have a medium or system to produce a visual facial likeness is that most people are not capable of producing (drawing) a facial image that would accurately reflect their memory of the face [18].

It is shown in Chapter 2 that existing face recall systems do not perform their required task as well as would be desired. The face recognition system developed in this thesis is an attempt at providing a system that does.

1.2 Outline of thesis

This thesis presents the development of a face recall system that uses an optimisation algorithm called PBIL to search through the coefficients of a linear combination of eigenfaces. The search is to arrive at a likeness to a desired face. The underlying theory is presented, then the implementation of the face recall system is given and finally the results of experiments to assess it are presented. What follows is breakdown of each chapter.

The Evolution of Face Recall Systems

Chapter 2 describes and analyses existing face recall systems. From this analysis it is seen that there is scope for improvement on the existing systems.

Eigenfaces for face image formation

Chapter 3 presents the topic of eigenfaces. It describes what they are, how to calculate them and how they can be linearly combined to reconstruct face images and to generate new ones. The eigenfaces are the principal component axes of a set of standardised face images, which can be used to reduce the dimensions of the space in which face images are represented. The reduced space is called *face space*. Issues relating to the use of eigenfaces are explained, this includes different distance measurements in face space.

Optimisation

Chapter 4 explains the theory of optimisation and how it can be used to search through the coefficients of the eigenfaces to help a witness find an image that is a likeness to the perpetrator. Different optimisation algorithms are considered for the task and it is decided that Probability Based Incremental Learning (PBIL) is the optimisation algorithm to be used. PBIL is then described.

Creating and Analysing the Face Space

Chapter 5 explains the need to standardise face images, before applying principal component analysis on them to generate the eigenfaces. It then presents how the face images used in this project are standardised. Investigations are done into the ability of the eigenfaces to generate new face images, which is found to be promising. The face space defined by the eigenfaces is then analysed. This is done by traversing the space and generating faces corresponding to points along the path traversed. Finally the distribution of faces in the space is looked at and the assumption that this approximates a Gaussian distribution is made.

The Developed Face Recall System

Chapter 6 starts with an explanation of why the face recall system was implemented in MATLAB. It is then shown how the PBIL optimisation algorithm can be modified to take advantage of the fact that the distribution of faces in face space is Gaussian. This is followed by the system implementation. The results of initial computer simulations with the face recall system are presented. In these simulations instead of a human interacting with the system to find a face, the computer automates this interaction. These were done to make the initial feasibility assessments of the system and to fine tune the free parameters of the PBIL algorithm used. The initial feasibility assessments prove to be surprisingly good. The chapter ends with an explanation of how to use the face recall system.

Experiments

Chapter 7 presents the results of experiments run with the system. In these experiments the participants try and construct face likeness images to faces that they are shown. Two different experiments are presented in this chapter. In the first the face to be reconstructed is viewable during the whole construction process. In the second the face to be reconstructed is only shown for a brief period before the reconstruction. All the constructed face images are judged, on their similarity to the face that they are supposed to be likenesses of, by sets of judges.

Comparison of the Developed Face Recall System with Existing Systems

Chapter 8 compares the developed face recall system with the existing face recall systems presented in Chapter 2. It shows that the developed face recall system addresses many of the shortcomings of the existing systems.

Conclusion

Chapter 9 summarises and highlights the important findings and results of the thesis.

Chapter 2

Analysis of Existing of Face Recall Systems

This chapter gives a description and analysis of existing face recall systems. This is done by presenting the existing face recall systems and then performing a critical analysis of them.

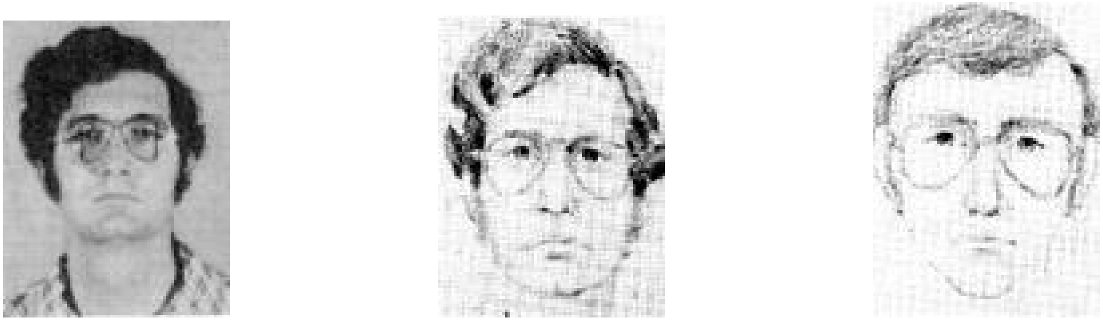
It shows how face recall systems have evolved, with newer systems trying to provide solutions to the shortcomings of previous systems. Also highlighted is that, according to the literature, there is presently no face recall system that works as well as would be desired.

2.1 Existing face recall systems

The existing face recall systems have categorised and presented in the following categories: *sketch artists*, *non computerised facial composite systems*, *computerised facial composite systems* and *computerised facial generation systems.* A facial composite system is a system that makes up a face by joining together *existing* facial features, like eyes, noses, mouths etc. A facial generation system is considered to be any system that does not use existing facial features. These face recall systems are presented in chronological order.

2.1.1 Sketch artists

The first widely used method to generate visual images of crime perpetrators was by having a police sketch artist draw the face, while the witness to the crime describes him/her. The primary limitation with this method is the witness's ability to describe the target face and the artist's ability to accurately sketch the features verbally described by the witness [17]. Figure 2.1 shows examples of sketches produced by sketch artists.



(a) Photograph

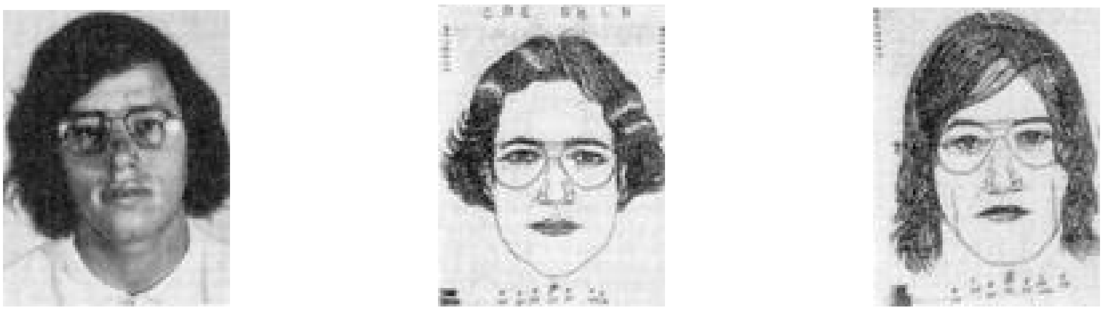
(b) High similarity sketch

(c) Low similarity sketch

Figure 2.1: Examples of a good and a poor sketch produced by sketch artists (*Laughery and Fowler [18]*).

2.1.2 Non computerised facial composite systems

The best known face recall systems, Identi-kit and Photofit, fall into this category. Both were and are widely used in the law-enforcement field [18, 9].



(a) Photograph

(b) High similarity composite

(c) Low similarity composite

Figure 2.2: Examples of a good and a poor Identi-kit composite produced by trained operators (*Laughery and Fowler [18]*).

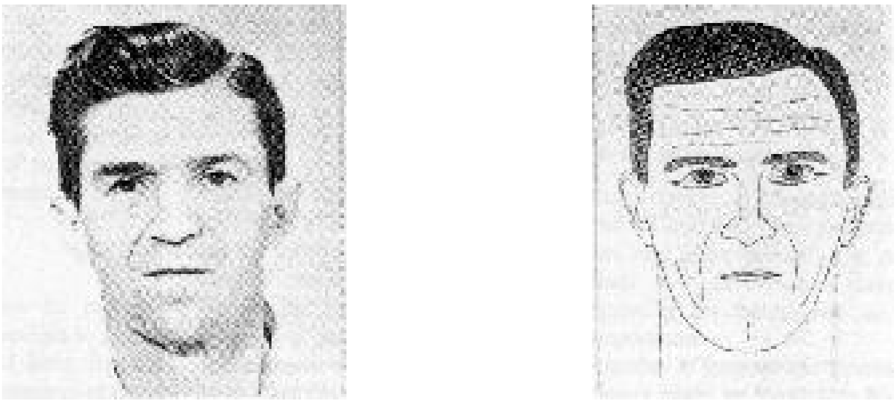
- With an Identi-kit system the composite face image is created by superimposing transparencies of line drawings of different facial features [6, 15, 11]. The features used in the composite vary with different implementations of the Identi-kit. The number of mouths, noses etc. from which to choose from also vary. Kovera et al. [17] describe one made up of 130 hairlines, 102 eyes, 37 noses, 40 mouths and 52 chins, accessories that can be added include hats, glasses, beards, scars and age lines. Figure 2.2 shows Identi-kit reconstructions of a face. A trained technician is used to construct the composite face by interacting with the witness to select appropriate features and then superimpose them. The technician can then use a marking pen to make additional modifications or add detail [18].

- Photofit is similar to Identi-kit, the main difference being that the transparencies contain gray-scale photographic images [6, 15]. Kovera et al. [17] describe a Photofit kit containing 195 hairlines/ears, 99 eyes/eyebrows, 89 noses, 105 mouths and 74 chins/cheeks. It also contains accessories such as hats, glasses and beards.

2.1.3 Computerised facial composite systems

Many computerised facial composite systems have been developed in the past decade. By being implemented on computer they can be made more user friendly, give users greater control over the resultant image, and be more interactive by providing quicker feedback than the non-computerised methods. Computer-based systems generally have larger sets of features to work with and therefore have greater expressive power in generating face images.

Choice based systems



(a) Photograph

(b) Composite

Figure 2.3: A sample result generated with the WHATSISFACE system (*Gillenson and Chandrasekaran [11]*).

Though computerised composite systems are more technically advanced than Photofit and Identi-kit, the fundamental concept underlying them is the same: construction of a facial composite by choosing from sets of features.

The following are implementations of computerised facial composite systems:

- WHATSISFACE, the first computerised facial compositor, was developed in 1975 by Gillenson and Chandrasekaran [11]. The system is capable of generating an extremely good likeness when a photograph is available to the user. An example of such a likeness can be seen in Figure 2.3. But when a face from memory was constructed the "result was relatively poor" [11]. This and the inordinate cost of the system at the time ruled it out for use in criminological applications.
- Facette from Softwehr is used by German law enforcement agencies [20]. It generates photo-style portraits. Chosen facial features can be positioned and scaled. Drawing tools are provided to touch up facial images [20]. Figure 2.4 shows the user interface and an facial image generated by the system.
- Mac-a-Mug Pro also allows the positioning of facial features. The available facial features are 184 hairlines, 117 eyes/eyebrows, 13 ears, 65 noses, 80 mouths, 45 chins, 34 eyeglasses, 36 mustaches, 23 sideburns, 18 beards, 7 head-wears and 20 different types of facial lines. The system has editing tools such as erasers, pencils and brushes [17].
- Compusketch from Visitex Corporation is a software package with which a trained operator can assemble a facial likeness in 45-60 minutes [15].

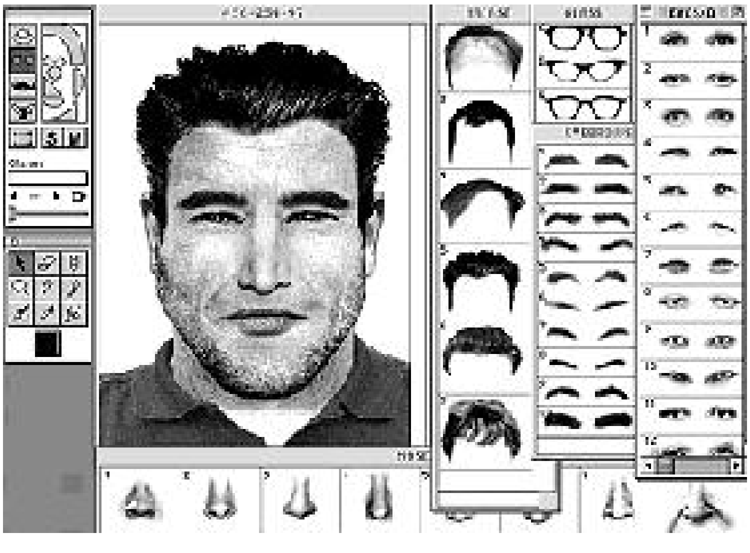


Figure 2.4: User interface of and composite produced by the Facette system.

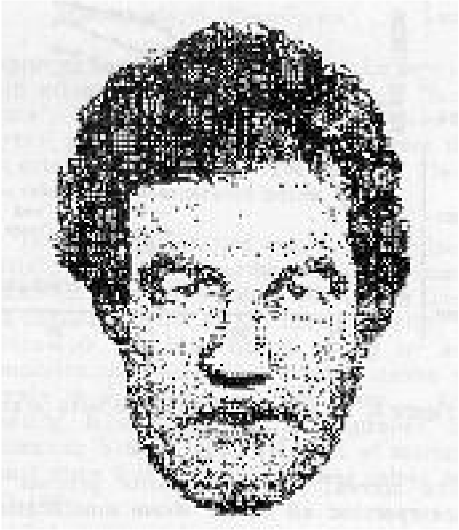


Figure 2.5: Example output face image from Johnston's system (*Caldwell and Johnston [5]*).

Johnston's recognition based system

The one exception to the method of building up a face image by choosing the features and then putting them on the composite face image being constructed is a system developed by Johnston and Caldwell [15, 5].

Johnston's system works as follows [5]:

1. 20 faces (the population) are generated by randomly selecting a forehead, a set of eyes, a nose, a mouth and a chin from the available sets of hand drawn facial features. These features are then randomly positioned, within predefined allowable ranges, to generate each of the 20 faces.

2. The 20 faces are shown sequentially to the witness and for each face the witness must rate the similarity of the displayed face image to the perpetrator's face image in memory.

3. Once all the faces have been rated, the ratings are fed into a genetic algorithm which will accordingly generate another population of 20 faces by choosing features and their associated positions. This population should look more like the highest rated faces in the previous population.

4. Return to step 2 until one of the faces in the population is considered to be a good enough likeness to the perpetrator.

Figure 2.5 shows an image generated by Johnston's system.

2.1.4 A computerised facial generation system

The facial composite systems described above all build up their face images through grouping together facial features chosen from available sets of features. Thus the number of features to choose from is limited by the sizes of the provided sets. Brunelli and Mich propose a face generation system, called SpotIt, where each facial feature in the generated face is created, instead of being chosen [4]. Now each facial feature can be one of a virtually unlimited set of features. SpotIt also searches through a mug-shot database and continuously returns the closest matches in the database to the built up face.

This is achieved by providing a GUI, shown in Figure 2.6, that seems to have a set of 15 scroll bars associated with each of 5 facial features. By adjusting the scroll bars in one of the sets, the associated facial feature is changed real time on the displayed image. This is implemented by creating each feature as a linear combination of eigenfeatures, or basis surfaces, that are associated with each facial feature (see Chapter 3, specifically Section 3.3.1, for an explanation of building facial features from eigenfeatures). Each scroll bar is associated with an eigensurface, and by adjusting the scroll bar the coefficient for that eigensurface is changed. This results in a change in the look of that facial feature. There are also scroll bars associated with the positioning of the mouth and nose.

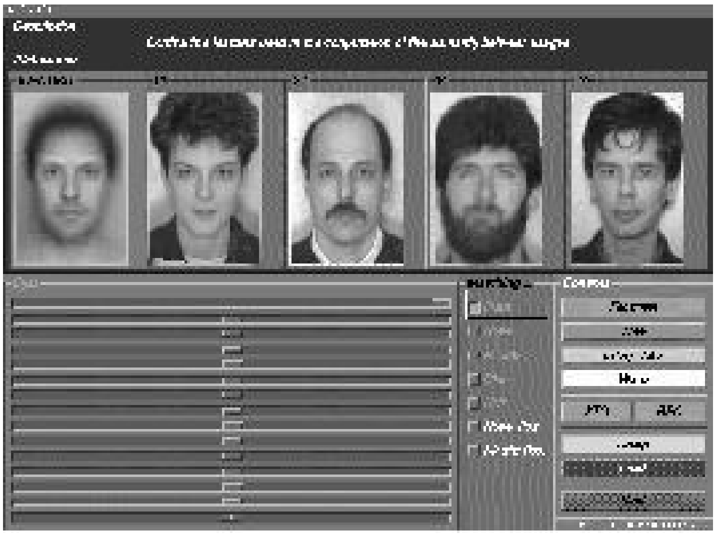


Figure 2.6: User interface of the SpotIt system. The face image on the left is the built-up face and the face images to the right of it are the closest matches in the mug-shot database (*Brunelli and Mitch [4]*).

Brunelli and Mich state that not all the functionality they describe has been implemented [4].

2.2 Critique of existing face recall systems

Three main factors contribute to the poor performance of most face recall systems. They are:

1. The limited expressive capabilities of the system to generate a good enough likeness. This is often due to the limited number of facial components available in the system [18, 4].

2. The need to have a skilled operator acting as an intermediary between the witness and the system. The witness must verbalise the mental image to the operator, who in turn translates the description into an image. This transfer of information is very susceptible to corruption and is where much of the difficulty in generating a likeness lies [17, 18, 33]. Communication difficulties could also be as a result of the witness's poor grasp of the language used by the operator. The introduction of a potential bias by the operator is also a problem, due to the use of the systems in legal and law enforcement areas [15].

3. The selection of facial features out of context of the facial image. Research notes that we perceive features of a face interdependently and not independently [28]. Specifically, it has been found that judging resemblance from features seen in isolation may be a serious source of distortion in composite systems [7]. A more holistic approach to face recall would potentially provide better results [18, 7].

Sections 2.2.1 to 2.2.4 discuss the shortcomings of the face recall systems previously described with respect to the above three main problems and others. These sections show how each new system tries to address the described shortcomings of its predecessors.

2.2.1 Sketch artists

There exist a limited number of people who can perform the job of a sketch artist and they are relatively expensive to employ. There exists a wide range in sketch artist's abilities to draw facial likenesses [18], this is a function of the artist's sketching abilities and his/her ability to communicate with the witness.

There is also the difficulty the witnesses have in recalling and verbally describing the perpetrator's face [17].

Laughery and Fowler note that sketches drawn by sketch artists are "considerably short of perfect" [18].

2.2.2 Non computerised facial composite systems

- Research has shown that Identi-kit composites generally bear little similarity to the faces they are meant to represent [17, 18].
- Experiments performed with the Photofit system also suggest that witnesses are unable to produce recognisable composites with the Photofit system [17, 9].

The failure of the above two systems can be attributed to the selection of features in isolation, the limited expressive capabilities of the system and the need to work through an intermediary.

2.2.3 Computerised facial composite systems

Choice-based systems

Kovera et al.'s [17] experiments raise questions about the effectiveness of composite systems as tools to promote the recognition of suspects in criminal contexts. Even though the experiments were performed using Mac-a-Mug Pro, they feel confident enough to extend their findings to choice-based composite systems in general. Wogalter et al. [33] found that using Mac-a-Mug Pro more useful composites could be generated than with non-computerised systems.

The improvements achieved with computerised facial composite system can be attributed to the fact that two of the three main problems are partially removed. The expressive capability of the systems are improved by providing more features to chose from and the ability to position them. With a well developed GUI the communication through the operator can be reduced or removed altogether.

Johnston's recognition-based system

Johnston's face recall system is different to the above-mentioned systems in that it is a face recognition system as opposed to a face generation system. This takes advantage of mankind's excellent facial recognition ability. The nature of the system removes the need for the selection of out of context facial features. The system is also fully automated and simple to use, therefore there is no need to have an operator.

Potential drawbacks of the system are its limited expressive capabilities due to the limited number of features that it can draw from to generate the facial composites and the fact that the features are line drawings. There is also no natural ordering of the features in each set, which makes the search space difficult for the genetic algorithm to traverse. This is explained in Section 5.3.2.

2.2.4 The computerised facial generation system

Brunelli and Mich's SpotIt system [4] seems to have solved the three main problems in face recall systems. The expressive capability of a correctly implemented and used system should be close to perfect. As there is a good GUI there is no need to have a skilled operator acting as an intermediary. The creation of facial features is in context of the facial image.

Due to the fundamentally different nature of the system, an issue exists that is critical to this system that was not put forward with the three main problems. For a person to position the scroll bars to generate an appropriate likeness seems to be a potentially impossible task. There are probably more than 75 of these scroll bars that have to be positioned to generate the face image. The user also does not know what the effect of changing one of the scroll bars associated with a facial feature will have on that particular facial feature.

Brunelli and Mich [4] do not discuss any trial runs that they have put their system through. Thus it is difficult to assess the effectiveness of their system in aiding people to generate an image of a facial likeness from memory.

Chapter 3

Eigenfaces for Face Image Formation

This chapter explains what eigenfaces are, how to obtain them from a set of face images and how they can be used to generate face images. The eigenfaces are the principal component axes of a set of standardised face images, which can be used to reduce the dimensions of the space in which face images are represented. The reduced space is called *face space*. The chapter includes explanations of the issues involved with using the eigenfaces. These include distance measurements in face space and how well they can construct face images in general. The chapter ends by relating face image construction using eigenfaces and human face recognition.

The chapter discusses eigenfaces without showing many results of using them. This is because the results are very susceptible to the set of face images used in the eigenface generation. By leaving this out the chapter is kept general and not specific to the set used. The next chapter shows the results obtained with the face images used in this project.

3.1 Introduction to eigenfaces

Eigenfaces were first used by Kirby and Sirovich [16, 29] in their work on face representation and compression. They were then popularised by Pentland et al. [22, 32] who used it for face recognition.

Eigenfaces are a set of face-like images that can be linearly combined to obtain face images. If $E_1 \ldots E_n$ are the n eigenfaces used and Av is the average face associated with the eigenfaces, then face images can be generated as follows:

$$\textit{face image} = Av + c_1E_1 + c_2E_2 + \cdots + c_nE_n, \tag{3.1}$$

with the coefficients $c_1 \ldots c_n$ determining what *face image* looks like.

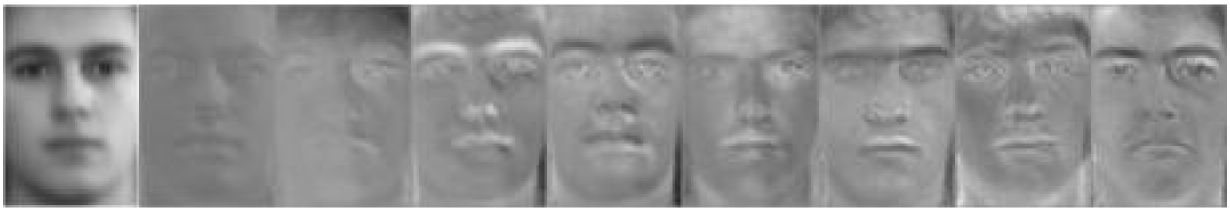


Figure 3.1: The average face Av followed by the first 8 eigenfaces $E_1, E_2, \ldots, E_8$.

Figure 3.1 shows the average face and the first 8 eigenfaces that have been used in this project. Figure 3.2 shows the results of using 46 eigenfaces to reproduce a new face. As can be seen the reconstruction looks very similar to the original. The reconstructed face image Figure 3.2(b) can be recognised as belonging to the person in the original face image Figure 3.2(a). This reproduction demonstrates the feasibility of using eigenfaces to reconstruct face images.

3.2 Principal component analysis

Principal component analysis [14] is the technique used to obtain the eigenfaces. Principal component analysis is closely related to the *Karhunen-Loeve expansion* or the *Hotelling transform*. The name *factor analysis* is also sometimes used to describe principal component analysis, but is ambiguous since it also describes other statistical techniques.

Principal component analysis is used as a technique to reduce the dimensionality of a set of data points, while still being able to account for as much of their variance as possible. It defines a new set of orthogonal axes to describe a set of data points, the points can be projected onto this new set of axes via a linear transform. The first principal component axis is oriented so that the points will have the largest possible variance along it. The nth principal component axis is oriented so that the points will also have as large a variance as possible along it, subject to the constraint that it must also be orthogonal to the principal component axes $1, \ldots, n-1$.

In Figure 3.3 a two dimensional dataset and its associated principal component axes are shown. If it is decided that the first principal component axis describes enough of the positional information of the data points then it is possible using only one number (c_{P_1}, the coefficient along P_1), instead of the original two numbers (c_{X_1}, c_{X_2}) to represent the data point $\mathbf{a}$ with

(a) The original face image

(b) The reconstructed face image

Figure 3.2: Reconstruction of a face image using a linear combination of 46 eigenfaces and the average face. The face in Figure 2(a) was not in the set of faces used to generate the eigenfaces.

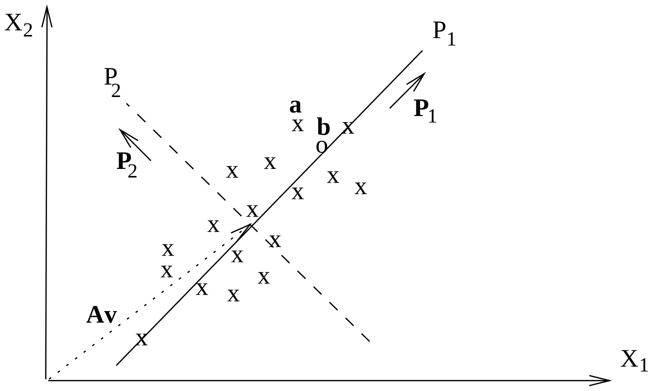


Figure 3.3: Principal component analysis. The x's represent data points initially described in term of the axes X_1 and X_2. P_1 and P_2 are the orthogonal principal component axes. $\mathbf{P}_1$ and $\mathbf{P}_2$ are unit vectors in the direction of their respective principal component axes P_1 and P_2. **Av** is the mean of the data points. **a** is a data point that can be represented by using only the first principal component axis in position **b**.

an associated error in the position $\mathbf{b}$. This is shown in (3.2):

$$\mathbf{b} = \mathbf{A}\mathbf{v} + c_{P_1}\mathbf{P}_1. \tag{3.2}$$

Notice that (3.1) is of the same form as (3.2). The average face and the first eigenface shown in Figure 3.1 correspond to $\mathbf{Av}$ and $\mathbf{P}_1$ respectively. This shows that the eigenfaces are in fact the principal component axes.

3.2.1 Formal definition of the principal components

Let $\Gamma_1, \Gamma_2, \ldots, \Gamma_M$ be a set of M vectors, each of size N. If Ψ is the mean of $\Gamma_1, \Gamma_2, \ldots, \Gamma_M$, let $\Phi_1, \Phi_2, \ldots, \Phi_M$ be $\Gamma_1, \Gamma_2, \ldots, \Gamma_M$ with Ψ subtracted out from each. These vectors represent points in $\mathbf{R}^N$. By applying principal components analysis on the vectors $\Phi_1, \Phi_2, \ldots, \Phi_M$ we will obtain a set of $N' \leq N$ orthonormal vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}'_N$, each of size N. The N' principal component axes have their origin at Ψ and are parallel to the vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}'_N$. Each vector $\mathbf{u}_i$ also defines the projection from the original space onto $\mathbf{u}_i$'s associated principal component axis.

The principal component axes are sequentially chosen as follows.

The vector $\mathbf{u}_1$ defining the first principal component axis is chosen such that the variance γ_1 of the data points projected onto it is a maximum. Written out, this is the case if

$$\gamma_1 = \frac{1}{M}\sum_{n=1}^{M}(\mathbf{u}_1^T\Phi_n)^2 \tag{3.3}$$

is a maximum, subject to the condition that

$$\mathbf{u}_1^T\mathbf{u}_1 = 1. \tag{3.4}$$

The the vector defining the kth principal component axis, $\mathbf{u}_k$, is chosen so that the variance γ_k of the data points projected onto the axis is a maximum subject to it being orthogonal to the already defined vectors $\mathbf{u}_1, \ldots \mathbf{u}_{k-1}$. (If $\mathbf{u}_k$ is orthogonal to $\mathbf{u}_1, \ldots \mathbf{u}_{k-1}$ then the principal component axis associated with it will be orthogonal to the principal component axes associated with $\mathbf{u}_1, \ldots \mathbf{u}_{k-1}$.) Written out, this is the case if

$$\gamma_k = \frac{1}{M}\sum_{n=1}^{M}(\mathbf{u}_k^T\Phi_n)^2 \tag{3.5}$$

is a maximum, subject to the condition that

$$\mathbf{u}_l^T\mathbf{u}_k = \begin{cases} 1 & \text{if } l = k \\ 0 & \text{otherwise} \end{cases} \qquad l = 1 \ldots k. \tag{3.6}$$

In the above equation $\mathbf{u}_l^T\mathbf{u}_k = 1$ forces normality so that $\mathbf{u}_k$ can be used to project the data onto the new principal component axis, while $\mathbf{u}_l^T\mathbf{u}_k = 0$ ensures that the kth principal component axis is orthogonal to the previous $k-1$ principal component axes.

3.2.2 Principal component analysis with digital images

Image space

A digital image can be considered as a point in so called image space $\mathbb{R}^{height \times width}$, where *height* and *width* are the height and width of the image in pixels. Each pixel can be considered to be an orthogonal axis in this space and the pixel value a variable along that axis. Figure 3.4 is a representation of this.

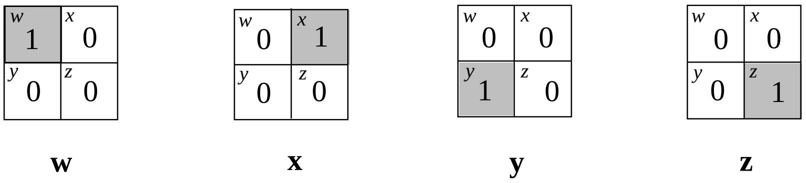


Figure 3.4: Representation of each image pixel as an orthogonal axis: $\mathbf{w}$, $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ are unit vectors along their associated axes w, x, y, z.

By specifying distances along each of the orthogonal axes above any 2×2 image can be generated. This is demonstrated with Figure 3.5 being the result of say, $5\mathbf{w} + 6\mathbf{x} + \mathbf{y} + 2\mathbf{z}$.

The image space described by the unit vectors in the pixel positions $(\mathbf{w}, \mathbf{x}, \mathbf{y}, \mathbf{z})$ can also be described using other sets of orthonormal vectors. An example of another such set of orthonormal vectors is shown in Figure 3.6.

The image in Figure 3.5 can also be reconstructed using the new set of orthogonal axis shown

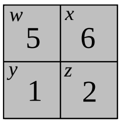


Figure 3.5: The image corresponding to $5\mathbf{w} + 6\mathbf{x} + \mathbf{y} + 2\mathbf{z}$.

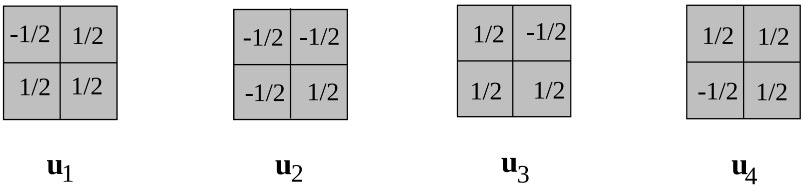


Figure 3.6: A new set of orthogonal axes u_1, u_2, u_3, u_4 described by the unit vectors $\mathbf{u}_1$, $\mathbf{u}_2$, $\mathbf{u}_3$, $\mathbf{u}_4$ along them.

in Figure 3.6 as follows

$$2\mathbf{u}_1 - 5\mathbf{u}_2 + \mathbf{u}_3 + 6\mathbf{u}_4 \tag{3.7}$$

Like Figure 3.6 each eigensurface in Figure 3.1 is a unit vector, which together describe a linear space. A difference exists in that the vectors $\mathbf{u}_1$, $\mathbf{u}_2$, $\mathbf{u}_3$, $\mathbf{u}_4$ describe the whole original space, while the eigensurfaces span only a subspace. This is illustrated by the fact that any 2×2 image can be generated using the axes shown in Figure 3.6, while not all images can be generated with the eigensurfaces. The eigensurfaces do not have enough dimensions to span the whole image space. If axis $u4$ was removed Figure 3.6 the comparison between the previous example and the eigensurfaces would be more appropriate.

Converting images to vectors

Images are two dimensional matrices yet the mathematics used in section 3.2.1 uses vectors representing points in space. This is easily accommodated by reading out the pixel values of the images in the order top to bottom, left to right and putting these sequentially into a vector of size $N = height \times width$. *height* and *width* being the height and width of the image. Reading out the pixel values of the image in Figure 3.5 would result in the following vector.

$$\begin{pmatrix} 5 \\ 1 \\ 6 \\ 2 \end{pmatrix} \tag{3.8}$$

To revert the results of the mathematical manipulation on the vectors back to image format, the vector is read back into a 2D matrix so that the vector positions are returned to the pixel positions from which they came.

3.3 Principal component analysis on face images

Frontal face images are homogeneous with respect to the features they contain, the position and the look of the features. Therefore if they are standardised[1] for scale and position within the image, they will not be uniformly distributed in image space. Instead they are grouped together in a region of the image space in a manner similar to that shown in Figure 3.3, except the space will have a higher dimensionality. Thus a relatively small number principal component axes should be able to describe most of the variation in facial images. The more similar the set of face images being described, the fewer principal component axes are needed to approximate them acceptably. Sirovich and Kirby [29] were the first to use this, when they demonstrated the possibility of using eigenfaces for face image compression.

3.3.1 Principal component analysis on face feature images

Images of facial features like the eyes, nose and mouth are also homogeneous and can therefore also be succinctly approximated by the linear combination of the average feature and a small set of eigenfeatures. The result of applying principal component analysis on images of right eyes is shown in Figure 3.7. The figure shows the average right eye followed by the first seven right eigeneyes.

Brunelli and Mich build up their composite face (see Section 2.1.4) by positioning together facial features. The facial features that they use are the hair, eyes, nose, mouth and chin. Each of these facial features is made up of a linear combination of eigenfeatures and the average feature.

[1] See Section 5.1.2 for a description of how and why the images are standardised.

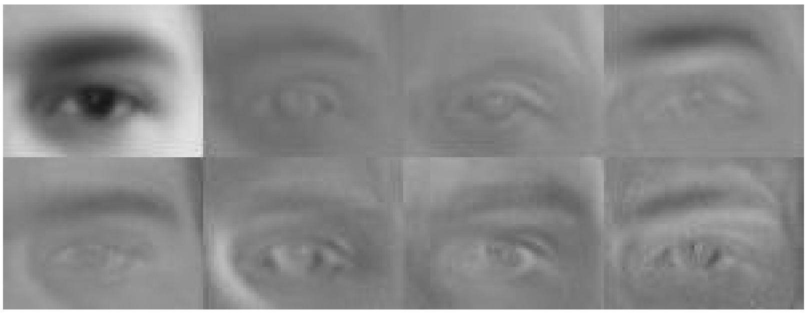


Figure 3.7: The average right eye followed by the first seven right eigeneyes. The sequence goes form left to right, top to bottom.

3.4 Computing the eigenfaces

This section shows how the principal component axes are computed by calculating the eigenvectors of the covariance matrix of the set of face images (subsection 3.4.1). The covariance matrix of images is shown to be generally too large to be solved for its eigenvectors. It is then set forth how by solving for the eigenvectors of a smaller matrix, the desired eigenvectors of the covariance matrix can be arrived at (subsection 3.4.2). The final subsection (subsection 3.4.3) explains why the eigenvectors of the covariance matrix give the principal component axes.

The descriptions in subsections 3.4.1 and 3.4.2 are similar to those presented by Sirovich and Kirby [29] and Turk and Pentland [32], but have been included in this thesis to make it self-contained.

3.4.1 Calculating the eigenvectors of the covariance matrix

Let $I_1(x,y), I_2(x,y), \dots, I_M(x,y)$ be a set of M *height*-by-*width* standardised[2] face images that will be used to generate the eigenfaces. Form a set of vectors $\Gamma_1, \Gamma_2, \dots, \Gamma_M$ by reading out the pixel values of each image $I_i(x,y)$ into its corresponding column vector Γ_i as described in section 3.2.2. This will result in each image now being stored in a vector of size $N = \mathit{height} \times \mathit{width}$, where N is the number of pixels in the image. This vector represents a point in N dimensional image space, $\Gamma_i \in \mathbb{R}^N$.

From $\Gamma_1, \Gamma_2, \dots, \Gamma_M$ calculate the average face image Ψ as follows:

$$\Psi = \frac{1}{M} \sum_{n=1}^{M} \Gamma_n. \tag{3.9}$$

[2] See Section 5.1.2 for a description of how and why the images are standardised.

Because it seems reasonable to assume that an efficient procedure for recognising and storing face images concentrates on departures from the mean or average face image [29], we will work with these deviations. Sirovich and Kirby [29] called these deviations from the mean *caricatures.* The caricatures $\Phi_1, \Phi_2, \ldots, \Phi_M$ are calculated as follows:

$$\Phi_i = \Gamma_i - \Psi. \tag{3.10}$$

From the caricatures $\Phi_1, \Phi_2, \ldots, \Phi_M$ calculate the covariance matrix

$$C = \frac{1}{M}\sum_{n=1}^{M} \Phi_n \Phi_n^T. \tag{3.11}$$

If the matrix A is defined to be the N-by-M dimensional matrix $[\Phi_1 \; \Phi_2 \; \Phi_3 \; \ldots \; \Phi_M]$, then the covariance matrix can also be calculated as follows

$$C = \frac{1}{M} A A^T. \tag{3.12}$$

Let $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N$ be the normalised eigenvectors of the covariance matrix C and $\gamma_1, \gamma_2, \ldots, \gamma_N$ their corresponding eigenvalues. The ordering of the eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N$ is such that their associated eigenvalues $\gamma_1, \gamma_2, \ldots, \gamma_N$ are in order from largest to smallest ($\mathbf{u}_1$ is the eigenvector with the largest eigenvalue γ_1 and $\mathbf{u}_N$ is the eigenvector with the smallest eigenvalue γ_N). The quantities $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N$ and $\gamma_1, \gamma_2, \ldots, \gamma_N$ described here are then the same $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N$ and $\gamma_1, \gamma_2, \ldots, \gamma_N$ used in (3.3) — (3.6). Thus the eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N$ are parallel to the principal component axes and can be used to project onto them. This is why the term eigenfaces is used — the eigenfaces are the eigenvectors of the covariance matrix of the set of face images.

Removing the scaling factor of $\frac{1}{M}$ from (3.12) will not change the eigenvectors of the covariance matrix. It will only result in a scaling of the eigenvalues. Since it is the eigenvectors that are of interest, to reduce calculation and for simplification we will use

$$C' = A A^T \tag{3.13}$$

For $\mathbf{u}_i$ to be an eigenvector of the modified covariance matrix C', the following must be true for each $\mathbf{u}_i$ and its associated eigenvalue γ_i'

$$AA^T\mathbf{u}_i = \gamma_i'\mathbf{u}_i \tag{3.14}$$

The matrix A is of dimension N-by-M. Therefore the dimension of AA^T is N-by-N. It is computationally unfeasible to solve for the eigenvectors and eigenvalues of such a large matrix; the height and width of typical images results in a matrix that is far too large.

The images in this thesis on which principal component analysis was performed were $N = 51 \times 76$. Thus the matrix to be solved for the eigenvalues would have been of the size $3876 \times 3876 = 15023376$. If each pixel is stored as a double precision floating point[3] data type of size 8 bytes this would take up 114MB of memory to store the matrix. Apart from the storage considerations this massive matrix would still have to be solved for its eigenvectors and eigenvalues.

3.4.2 Reducing the dimensionality of the eigenvector problem

Consider the eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N$ of A^TA and their associated eigenvalues $\mu_1, \mu_2, \ldots, \mu_N$. The following is true of each $\mathbf{v}_i$ and μ_i:

$$A^TA\mathbf{v}_i = \mu_i\mathbf{v}_i. \tag{3.15}$$

If we pre-multiply both sides by A, we have

$$AA^TA\mathbf{v}_i = \mu_iA\mathbf{v}_i \tag{3.16}$$

and by comparing this to (3.14) we can see that if we set

$$A\mathbf{v}_i = \mathbf{u}_i \tag{3.17}$$

[3] The pixels are stored as double precision floating points because fractional precision is wanted in the eigenvalues, single precision floating point would be sufficient but the platform used (MATLAB) does not support them.

then

$$\mu_i = \gamma_i'. \tag{3.18}$$

So $A\mathbf{v}_1, A\mathbf{v}_2, \ldots, A\mathbf{v}_N$ are the eigenvectors of $C' = AA^T$, with $\mu_1, \mu_2, \ldots, \mu_N$, being their associated eigenvalues. This means that another way to obtain the eigenvectors of the modified covariance matrix $C' = AA^T$ is to calculate the eigenvectors for A^TA and then multiply them with A as shown in (3.17).

The matrix A is of dimension N-by-M. Therefore A^TA is of dimension M-by-M. In practice the number of images M is much smaller than the number of pixels N in each image $I_i(x, y)$, in other words $M \ll (height \times width)$. This changes the calculation of the eigenvectors into a computationally feasible problem.

Equation (3.17) can also be written as

$$\mathbf{u}_i = \sum_{j=1}^{M} v_{ij} \Phi_j \tag{3.19}$$

where v_{ij} is the jth element of the intermediate eigenvalue $\mathbf{v}_i$. This shows that each eigenvector $\mathbf{u}_i$ is actually a linear combination of the caricatures $\Phi_1, \Phi_2, \ldots, \Phi_M$.

3.4.3 Why the eigenvectors of the covariance matrix give the principal component axes

This subsection shows that the eigenvectors of the covariance matrix can be used to linearly project the images onto a new set of axes, along which they are uncorrelated. It is shown that because the images are uncorrelated along these axes they must be the principal component axes.

Describing the covariance matrix

The covariance matrix C in (3.12) is a N-by-N matrix with its individual elements defined as

$$c_{ij} = \frac{1}{M} \sum_{k=1}^{M} \Phi_{ki} \Phi_{kj} \tag{3.20}$$

where c_{ij} is the (i,j)th component of the matrix C and Φ_{ki} and Φ_{kj} are the ith and jth elements of the caricature vector Φ_k.

The relation in (3.20) would normally be of the form $c_{ij} = \frac{1}{M}\sum_{k=1}^{M}(\Phi_{ki} - \bar{\Phi}_i)(\Phi_{kj} - \bar{\Phi}_j)$ where $\bar{\Phi}_i$ is the mean of the ith elements across the caricatures $\Phi_1, \Phi_2, \ldots \Phi_M$. However $\Phi_1, \Phi_2, \ldots \Phi_M$ have a mean of zero, so the expression is simplified.

The diagonal elements c_{ii} of C are the variances of the ith elements of $\Phi_1, \Phi_2, \ldots \Phi_M$, while c_{ij} is the covariance of the ith element with the jth element of the caricatures. The covariance c_{ij} is directly related to the correlation coefficient r_{ij} which is a measure of the correlation between the two elements [10, page 467]

$$r_{ij} = \frac{c_{ij}}{\sqrt{c_{ii}c_{jj}}}. \tag{3.21}$$

This shows that when the covariance $c_{ij} = 0$, the correlation coefficient $r_{ij} = 0$.

Describing the correlation coefficient

The correlation coefficient is an inverse measure of the deviation of a set of points from the least squares best-fit line through them. The correlation coefficient r_{xy} is defined as

$$r_{xy} = \sqrt{\frac{\sum(\hat{y} - \bar{y})^2}{\sum(y - \bar{y})^2}} \tag{3.22}$$

where y is the y value of a point, $\bar{y}$ is the average y value of the set of points, and $\hat{y}$ is the predicted y value of the point according to the least squares best-fit line through the points (the summation being across all the points in the set). The relationship between y, $\bar{y}$, $\hat{y}$ is shown in Figure 3.8.

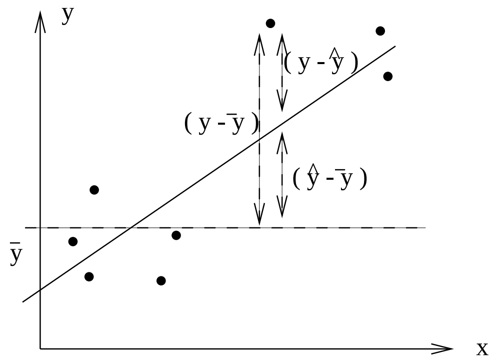


Figure 3.8: Diagram showing the relationship between y, $\bar{y}$, $\hat{y}$.

Figure 3.8 shows that

$$(y - \bar{y}) = (\hat{y} - \bar{y}) + (y - \bar{y}). \tag{3.23}$$

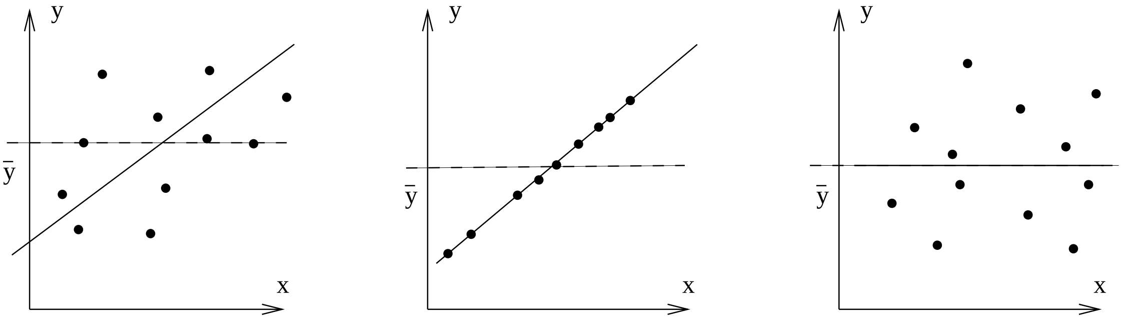
so

$$\sum(y - \bar{y})^2 = \sum(\hat{y} - \bar{y})^2 + \sum(y - \bar{y})^2 \tag{3.24}$$

$\sum(y - \hat{y})^2$ is called the *residual sum of squares* and is a measure of the deviation of the points from the least squares best-fit line. $\sum(\hat{y} - \bar{y})^2$ is called the *regression sum of squares* and it measures that part of the total variation of the y's that can be ascribed to the relationship between the two variables x and y.

Using (3.24), (3.22) can be written out as

$$r_{xy} = \sqrt{\frac{\sum(\hat{y} - \bar{y})^2}{\sum(\hat{y} - \bar{y})^2 + \sum(y - \bar{y})^2}}. \tag{3.25}$$



(a) $0 < r_{xy} < 1$

(b) $r_{xy} = 1$

(c) $r_{xy} = 0$

Figure 3.9: Three different point distributions and their associated correlation coefficient r_{xy} values. The solid line through the data points is the least squares best fit line.

Figure 3.9 shows three different point distributions and their associated correlation coefficient values. Figure 3.9(c) shows a point distribution where $r_{xy} = 0$, which means that x and y are uncorrelated. This occurs when the scatter of the points is such that the least squares best-fit line is parallel to the x axis and goes through $\bar{y}$. This happens when the best estimate of a y value given the associated x value is $\bar{y}$ because there is no correlation between y and x. $\sum(y - \bar{y})^2$ then equals $\sum(y - \hat{y})^2$ so $\sum(\hat{y} - \bar{y})^2 = 0$, which will result in $r_{xy} = 0$ (from (3.25)).

The least squares best-fit line is the line along which the variance is the greatest. It can be seen from Figure 3.9 that by rotating the x and y axes so that the x axis is parallel to the least squares best-fit line the correlation coefficient becomes zero. This is what is effectively done in principal component analysis: the original axes are rotated so that the correlation of the data points along every combination of two of the axes is zero. This will result in a set of axes along which the variances will be the greatest possible.

Rotating the axes to make covariances zero

If we manage to rotate the axes so that the covariances of the caricatures $\Phi_1, \Phi_2, \ldots \Phi_M$ are zero, then the correlations of the caricatures will also become zero because of the relationship between the covariances and the correlations shown in (3.21). We then have the principal component axes.

The eigenvalue problem which has to be solved to obtain the eigenvectors of the modified covariance matrix C' is

$$\begin{aligned} \Lambda &= \grave{U}^T C' \grave{U} \qquad (3.26)\\ &= \grave{U}^T A A^T \grave{U}, \qquad (3.27) \end{aligned}$$

where Λ is the diagonal matrix of the eigenvalues and $\grave{U}$ is the eigenvector matrix which is made up of the eigenvectors as $[\grave{\mathbf{u}}_1, \grave{\mathbf{u}}_2, \ldots, \grave{\mathbf{u}}_M]$.

Consider $(\grave{U}^T A)^T$ which can be written out in the following manner:

$$\begin{aligned} (\grave{U}^T A)^T &= A^T \grave{U}^{TT} \qquad (3.28)\\ &= A^T \grave{U}. \qquad (3.29) \end{aligned}$$

Using this (3.27) can be rewritten as

$$\Lambda = \grave{U}^T A (\grave{U}^T A)^T. \qquad (3.30)$$

If the eigenvectors $\grave{\mathbf{u}}_1, \grave{\mathbf{u}}_2, \ldots, \grave{\mathbf{u}}_M$ are normalised then $\grave{U}^T A$ is the projection of caricatures $\Phi_1, \Phi_2, \ldots \Phi_M$ onto the axes defined by the eigenvectors. By comparing the form of (3.30) to (3.13) it can be seen that Λ is actually the covariance matrix of the projected points. As the eigenvalue matrix Λ is a diagonal matrix it means that there are no correlations between the projected data points. Thus the variances along these axes will be a maximum and this means that they are the principal component axes. All that is left is to order the eigenvectors so that their eigenvalues go from largest to smallest and then we have the ordering of the principal component axes.

3.5 Face space

In psychological literature on face recognition the term *face space* has been used to describe a space with dimensions along which faces vary. A face can be uniquely represented as a point in that space. However, there is no description of what the axes are that define the space. Proposals include hair colour and length, face shape and age [13].

In scientific and engineering literature on eigenface based computerised face recognition the term *face space* is used to describe the subspace, of the original $N = height \times width$ image space, described by the eigenfaces. A face image corresponds to, or can be approximated by, a point in this subspace. The use of the term *face space* in this thesis conforms to this definition.

Face space is a subspace of image space that captures the statistical properties or modes of variation of the face images which were used to generate the eigenfaces. Points in face space will therefore have a disposition to produce face-like images. These face-like images will also have a higher probability of looking like faces belonging to the same sex, race and age groups as the face images used in the generation of the eigenfaces.

3.6 Face image reconstruction using eigenfaces

3.6.1 Finding the eigenface coefficients (projecting onto face space)

Let Ω be a face image in vectorised form. It can be projected onto the new set of orthogonal principal component axes in the following manner:

$$\omega_k = \mathbf{u}_k^T(\Omega - \Psi) \qquad k = 1, \dots, M'. \tag{3.31}$$

M' is the number of principal components that are to be used in the face image reconstruction. This means that M' is the number of dimensions of the reduced face space in which we will work ($M' \leq M$). The scalars $\omega_1, \omega_2, \dots \omega_{M'}$ are the coefficients for the face image Ω projected onto the principal component axes. These are therefore the eigenface coefficients for the face Ω.

The vector Ω can be a face image from the set of face images that principal component analysis was performed on, or it can be a face image from outside that set.

3.6.2 Building up a face image with eigenfaces (projecting back to image space)

A reconstruction of Ω using only the M' eigenfaces is done as follows:

$$\tilde{\Omega} = \Psi + \sum_{n=1}^{M'} \omega_n \mathbf{u}_n. \tag{3.32}$$

The vector $\tilde{\Omega}$ is a reconstruction of Ω. It is just the average face Ψ summed with the eigenfaces multiplied by their associated coefficients. Furthermore, $\tilde{\Omega}$ is the best approximation that can be made to Ω with a linear combination of the eigenfaces and the average face. The use of the words *best approximation* is with respect to minimising the root mean square (rms) error between $\tilde{\Omega}$ and Ω (Section 3.7 discusses this error).

What (3.32) effectively does is to project the point in face space back to the original image space. Once we have the point back in image space we can display it because we know what the pixel values for each pixel to be displayed are.

3.7 Distance measures between face images

Figure 3.10 shows the decomposition of the original image space into two mutually exclusive and complementary subspaces: face space $\mathbf{F}$ defined by the M' eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}'_M$ used and its orthogonal complement $\bar{\mathbf{F}}$ spanning the rest of the original image space.

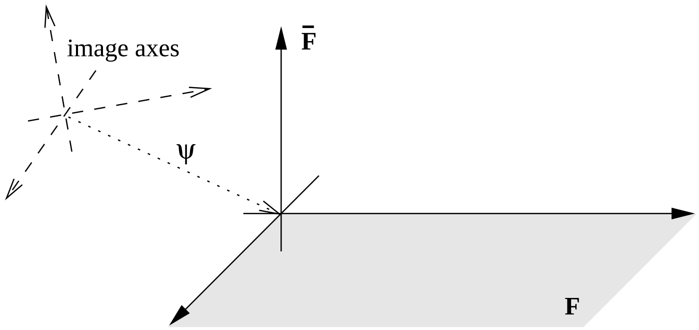


Figure 3.10: Decomposition of the original image space into face space $\mathbf{F}$ and its orthogonal complement $\bar{\mathbf{F}}$.

Figure 3.11 shows two face images Ω_A and Ω_B and their projections onto face space, $\tilde{\Omega}_A$ and $\tilde{\Omega}_B$. It shows how distance measures between the two face images in $\mathbf{F}$ and the original image

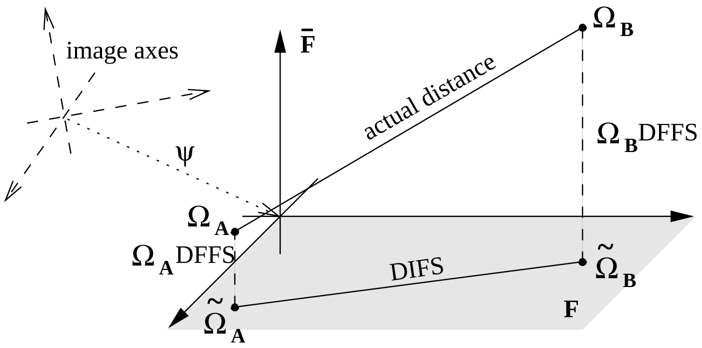


Figure 3.11: The projection of face images onto face space and the resulting distance measures. $\tilde{\Omega}_A$ and $\tilde{\Omega}_B$ are the face images Ω_A and Ω_B projected onto the face space $\mathbf{F}$. DIFS is the *Distance In Face Space* between Ω_A and Ω_B, while DFFS is the *Distance From Face Space* of Ω_A and Ω_B.

space relate to each other. All distances shown in Figure 3.11 are calculated as Euclidean distances.

The *Distance In Face Space* (DIFS) is simply the Euclidean distance between $\tilde{\Omega}_A$ and $\tilde{\Omega}_B$. If the variables along the face space axes, calculated using (3.31), for $\tilde{\Omega}_A$ are $\omega_{\Omega_A 1}, \omega_{\Omega_A 2}, \ldots, \omega_{\Omega_A M'}$ and those for $\tilde{\Omega}_B$ are $\omega_{\Omega_B 1}, \omega_{\Omega_B 2}, \ldots, \omega_{\Omega_B M'}$ then

$$\text{DIFS}(\Omega_A, \Omega_B, \Psi, \mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{M'}) = \sqrt{\sum_{i=1}^{M'} (\omega_{\Omega_A i} - \omega_{\Omega_B i})^2} \tag{3.33}$$

The DIFS is dependent on the average face and the eigenvectors which define the face space since the projections of Ω_A and Ω_B onto face space are calculated using them as shown in (3.31). It can be seen from Figure 3.11 that by changing the orientation and position of the axes the DIFS will change even though Ω_A and Ω_B are in the same position relative to the original image space. DIFS is also particularly sensitive to M' the number of dimensions of face space since it accounts for the number of terms in (3.33).

The *Distance From Face Space* (DFFS) is the distance from a face image Ω to its projection onto face space $\tilde{\Omega}$. If the axes of face space are well chosen it is hoped that the DFFS for all standardised[4] face images of the desired population group will be small. It can be seen from Figure 3.11 that if the DFFS for both the images being compared is zero then the DIFS between them is the actual distance between them.

[4] See Section 5.1.2 for the type of standardising and why it is needed.

DFFS could be calculated as follows:

$$\mathrm{DFFS}(\Omega, \Psi, \mathbf{u}_{M'+1}, \mathbf{u}_{M'+2}, \dots, \mathbf{u}_N) = \sqrt{\sum_{i=M'+1}^{N} \omega_i^2}. \tag{3.34}$$

This is the Euclidean distance between Ω and $\tilde{\Omega}$. However, since only the first M principal component axes are calculated it is not possible to calculate it in this way. Another way of calculating DFFS is

$$\mathrm{DFFS}(\Omega, \Psi, \mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{M'}) = \sqrt{\sum_{i=1}^{N} (\Omega_i - \tilde{\Omega}_i)^2}. \tag{3.35}$$

This calculation is done by projection Ω onto face space using (3.31) to obtain $\omega_1, \omega_2, \dots, \omega_{M'}$ and then projecting from face space back to the original image space to obtain $\tilde{\Omega}$ in terms of the image space. Then the Euclidean distance is calculated along the dimensions of the image space.

From (3.35) it can be seen that the DFFS is a measure of how well a face image can be approximated with a set of eigenfaces.

The *root mean square* (rms) error between a face image and its reconstruction is

$$\begin{aligned} \text{rms error}(\Omega, \tilde{\Omega}) &= \sqrt{\frac{1}{N}\sum_{i=1}^{N} (\Omega_i - \tilde{\Omega}_i)^2} && (3.36) \\ &= \frac{1}{\sqrt{N}}\sqrt{\sum_{i=1}^{N} (\Omega_i - \tilde{\Omega}_i)^2}. && (3.37) \end{aligned}$$

The rms error is also a function of Ψ and $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{M'}$ since $\tilde{\Omega}$ is a function of them, but to conform to the standard use of the rms error the notation 'rms error$(\Omega, \tilde{\Omega})$' is used.

Comparing (3.37) to (3.35) it can be seen that

$$\text{rms error}(\Omega, \tilde{\Omega}) = \frac{1}{\sqrt{N}} \; \mathrm{DFFS}(\Omega, \Psi, \mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{M'}) \tag{3.38}$$

This shows that the rms error is a more stable and interpretable measure since it is not affected by the image size because of the $\frac{1}{\sqrt{N}}$ scaling. If N is kept constant then comparisons of different reconstructions using DFFS are valid.

3.7.1 Why the distance measures will not necessarily coincide with human similarity perceptions

The distance measures presented above measure the differences between two images, while humans are capable of giving a measure of the similarity or difference of the actual faces in the images. Human rating of similarity between face images will be relatively invariant to the scale, head position, angle of rotation of the face and facial distortions like smiling in the image. The distance measures presented above are all sensitive to these types of variations in the images.

These distance measures are based on measuring the differences between corresponding[5] pixels in the images. For them to have meaning with respect to similarity of faces, the corresponding pixels must be representing the same part of a face. Any of the above variations result in a shift of part of or all of the face within the image, then the corresponding pixels in the two images will be representing different parts of a face.

The effects of the above variations can be reduced by warps, but this is at the expense of more calculation and complexity. Another way to control for these variations is to try and limit their occurrence. This can be done by controlling the subject while the image is being taken.

3.8 The reconstruction capabilities of the eigenfaces

Not much was found in the literature on eigenfaces about their reconstruction capabilities. This is due to the fact that they are generally used for face recognition and not compression (which is directly related to reconstruction).

Sirovich and Kirby [29] performed principal component analysis on a set of 115 cropped face images. They found that using 40 eigenfaces they obtained good reconstructions of the images on which they performed principal component analysis. The rms error in the reconstructions of the cropped images was about 2%.

For a face recall system we would like to be able to generate facial likenesses of a wide range of people, not only those whose facial images were used to generate the eigenfaces. Thus Sirovich and Kirby's reconstructions are not an indication of the reconstruction capabilities of the system as it would be used.

The results of tests showing the reconstruction of new face images are presented in Section 5.2.2.

[5] The words *corresponding pixels* mean pixels in the same positions.

3.9 Relating the eigenfaces technique to human face recognition

There has been speculation that human face recognition is based on methods similar to the eigenfaces technique:

- The use of eigenfaces in face image construction results in the construction of a face image in a holistic manner. It is believed that humans perform their face recognition based on a holistic model of the face. They do not base their recognition on the analysis of facial features in isolation [28].
- Hancock, Bruce and Burton [12] state that there is considerable evidence that human representation of faces may be based upon relatively low level image features describing relative light and dark areas, rather than more abstract descriptions of facial features like feature separation and protuberances.
- Turk and Pentland [32] state that it appears likely that there must be a face recognition mechanism in humans that is based on low level, two dimensional image processing.
- There have been psychological experiments performed that relate human recognition performance on face images with the associated eigenface coefficients [23, 24, 13, 12]. In general the eigenface coefficients yield good predictions for some aspects of the human performances.

These seem suggest that there might be some sort of natural relationship between human face recognition and the eigenface technique for face image reconstruction. This could get exploited by a face recall system that uses eigenfaces to generate its faces.

Chapter 4

Optimisation

This chapter starts with a brief introduction into the field of optimisation. Then the task of finding the coefficients for the eigenfaces, which will result in a likeness to the perpetrator's face, is presented as an optimisation problem. This enables any of a myriad of optimisation algorithms to be used to find a likeness. These are evaluated and it is decided to use *Population Based Incremental Learning* (PBIL). The chapter ends with a description of the PBIL algorithm.

4.1 Function maximisation or minimisation

Optimisation is the field that covers methods that try to find the maximum or minimum point of a function over a specified domain. More specifically, given a function f which depends on one or more independent variables, optimisation involves finding the values of these variables where f takes on a maximum or minimum. The domain of f is known as the *search space.* The above problem will be discussed in terms of minimising the function f, since the problem of maximising any function f can be restated as that of minimising $-f$.

f is known as the *cost function* or the *objective function.* The functional relationship between the variables and the cost function does not necessarily have to be known. If this is the case then one must 'evaluate the function' by experimenting on the real object to be optimised or on a model of the object. The model can often be implemented on a computer and the experiments take the form of computer simulations.

The function f to be minimised can have both local and global minima. A local minimum is a minimum in a finite neighbourhood that does not include the full range of the independent variables. A global minimum is the the lowest value within the range of the independent variables. See Figure 4.1 for examples of maxima and minima, both local and global for one independent variable.

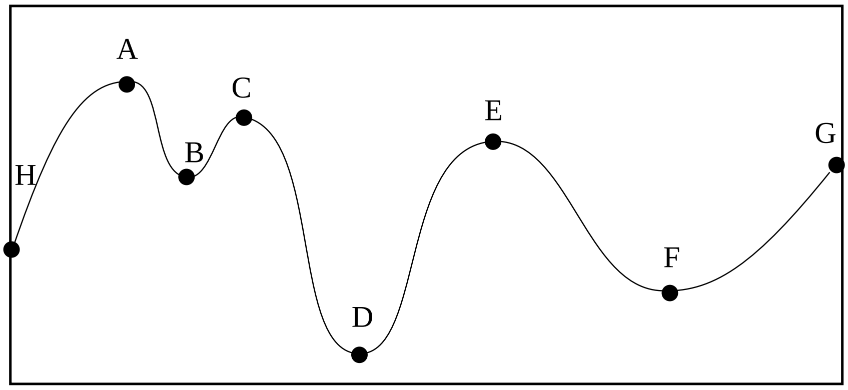


Figure 4.1: Plot showing examples of local and global minima and maxima. The above function of one independent variable is shown over its domain of interest. D is the global minimum, with H, B, F being the local minima. A is the global maximum, with C, E, G being local maxima.

There exists a massive number of different optimisation methods that can be used to try to find the optimum point of the cost function. Schwefel [27, page 3] states “it is a simple matter to collect several thousand published articles about optimisation methods”. The nature of the cost function is a primary factor in determining which of these different optimisation algorithms to use. The main factors which influence this choice are the form in which the cost function is expressed and can be evaluated, and the shape of the cost function.

4.2 Structuring the task of recalling a face as an optimisation problem

Figure 3.2 shows that it is possible to reconstruct good approximations to new face images (face images not in the set on which the principal component analysis was performed) with eigenfaces. (More examples are shown in Figure 5.14.) This is done by projecting the new face image onto the principal component axes to obtain the coefficients for the face image along these axes. The approximation is then generated by multiplying the eigenfaces with the coefficients along these axes and summing the products with the average face. This was shown in (3.32), which written out is

$$\tilde{\Upsilon} = \Psi + \omega_1\mathbf{u}_1 + \omega_2\mathbf{u}_2 + \omega_3\mathbf{u}_3 + \ldots + \omega_{M'}\mathbf{u}_{M'}, \tag{4.1}$$

$\tilde{\Upsilon}$ being the approximate reconstruction, Ψ the average face, M' the number of eigenfaces to use in the reconstruction, $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{M'}$ the eigenfaces and $\omega_1, \omega_2, \ldots, \omega_{M'}$ the coefficients to multiply the eigenfaces by.

So it should be possible to reconstruct a face image that looks like that of the perpetrator. The problem is to find the correct coefficients $\omega_1, \omega_2, \ldots, \omega_{M'}$, since we do not have a face image to project onto the principal component axes. The search for the correct coefficients can be viewed as an optimisation problem that could perhaps be solved using an optimisation algorithm with the witness performing the evaluation of the cost function and feeding the evaluation back to the algorithm. This could be done by showing face images, corresponding to points in the search space, on the screen and letting the witness evaluate them in some sort of manner yet to be defined.

The reason that it possible to use an optimisation strategy to find a likeness to the perpetrator is that the dimensions of the search space are now M'. This is manageable. If optimisation was performed directly on the pixels of the digital image the number of independent variables would be $N = \textit{height} \times \textit{width}$. This means that the search space would be much larger and include areas which do not result in face-like images. These areas could result in car like images, house-like images etc., though most of these areas would probably result in images that do not look like anything at all. Performing the optimisation on a search space like this would not work.

4.2.1 Constraints imposed due to a human cost function

The cost function in this optimisation problem cannot be expressed in terms of a functional relationship which can be evaluated by a computer at different points in the search space. Neither can it be modelled on a computer, which could then evaluate it or perform a simulation to evaluate it. These cases are the norm in optimisation problems. The cost function in this problem exists in the human's brain and he/she must perform the evaluation of it.

Because a human is 'evaluating his/her own cost function' there are a number of issues which must be taken into account when choosing the optimisation technique. These issues will be used to rule out different optimisation strategies for the task at hand.

1. There is no way to get or approximate the partial derivatives of the cost function.
2. The cost function cannot be evaluated to give a number.
3. The task of evaluating the cost function must be made as easy as possible for the witness.
4. The number of evaluations of the cost function, or faces that the witness is exposed

to, must be kept to a minimum. This is because the concentration of the witness will decrease with the number of faces he/she is shown.

5. There will be 'human error' associated with the judgement of similarity of the faces. Different people exposed to the same perpetrator will give different judgements of similarity to the same face images.

6. There will be human variability associated with the judgement of similarity of faces. The same person shown the same face images at different times will not always give the same judgement of similarity to them. This can be viewed as noise in the cost function.

As stated above it is important to make the process of dealing with the optimisation algorithm as easy as possible for the witness. What is wanted is to get some form of evaluation of points in the search space that is easy for the witness to give, as impartial as possible to variability in the witness evaluations and yet still provides enough information for the optimisation strategy to try find the optimum or a point close enough to it. The best envisaged way to achieve this would be to show the witness a set of faces and let him/her select the face that seems to him/her to be the most similar to the perpetrator's.

This suggests an evolutionary optimisation algorithm (see Section 4.4.4) that generates a population of trial solutions which can be shown to the witness. In this case the following issues must also be considered:

7. It is not ideal to make the witness associate a similarity measure with each face in the population as Johnston [5, 15] did. This is time-consuming for the human to perform and is vulnerable to human error and variability. What would be a better approach would be for the human to choose what he/she perceives as being the best match from the trial solutions in each generation. This would be less time-consuming and also less vulnerable to human variability and error.

8. The best match chosen will also not necessarily have its coefficients being the closest, in terms of Euclidean distance, to the optimal set of coefficients. The optimal set of coefficients being the coefficients calculated from projecting a face image of the perpetrator (if there was one available) using (3.31) onto the principal component axes.

9. There will still be a variability associated with the human's choice of best match. Given the same set of trial solutions, the user might not always choose the same solution as being the closest match. But this variability should be less than that associated with other measures of similarity, like giving ratings of similarity to different face images.

4.3 Encoding solutions as bit-strings

Many evolutionary methods (see Section 4.4.4) and others like the stochastic hill climber (see section 4.4.3) encode the trial solutions using binary numbers. This reduction of cardinality decreases the number of values that each digit can hold to two, which is the smallest number of values that can be used. This simplifies the manipulation of each bit (binary digit): it can either be left as is or flipped.

The encoding can be done in the following manner. Let n be the number of binary digits to be used for each independent variable. Divide up the domain of the independent variable into 2^n equal sized intervals. Associate the start positions of the intervals with increasing binary numbers using n bits and starting at $0_1 0_2 \dots 0_n$. This is shown in Figure 4.2. By increasing n the resolution of the binary coded representation of the independent variable is increased. Fig 4.2 shows that there is no binary digit associated with the maximal value that was used in the calculation of the divisions. This is not important as long as n is of a sufficient size, because by increasing n the size of the gap between the last position represented by a binary number and the maximal value of the domain decreases. If this is still considered to be a problem a slightly larger maximal number can be used.

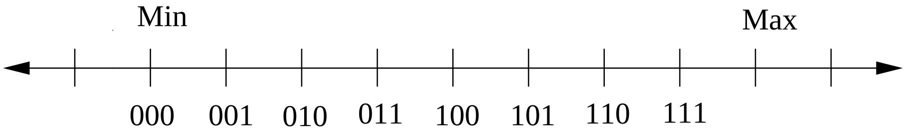


Figure 4.2: Using binary numbers to represent the search space. Here $n = 3$. Min is the start of the domain and Max is the end.

The conversion of a trial solution to a decimal value is then done as follows

$$\text{decimal} = \text{Min} + (b_0\ b_1\ b_2\ \dots\ b_{n-1}) \begin{pmatrix} \frac{2^{n-1}}{2^n} \\ \frac{2^{n-2}}{2^n} \\ \frac{2^{n-3}}{2^n} \\ \vdots \\ \frac{2^0}{2^n} \end{pmatrix} (\text{Max} - \text{Min}) \tag{4.2}$$

With Min being the start of the domain of the independent variable, Max being the end of the domain and $(b_0\ b_1\ b_2\ \dots\ b_{n-1})$ the binary encoded trial solution.

The trial solutions for each of the independent variables can be concatenated to form a bit-string

$$b_0\ b_1\ b_2\ \dots\ b_{n-1}\quad b_0\ b_1\ b_2\ \dots\ b_{n-1}\quad \dots\quad b_0\ b_1\ b_2\ \dots\ b_{n-1} \tag{4.3}$$

The searching part of the optimisation algorithm then manipulates the bit-string as a whole without consideration as to which bit is associated with which independent variable. The trial solutions are separated from the bit-stream before each evaluation in order that (4.2) can be used to generate a decimal value, which will be used in the evaluation of the cost function.

4.4 Optimisation strategies

This section analyses different optimisation strategies with respect to their suitability for finding the eigenface parameters to generate a likeness to the perpetrator. It would be impossible to investigate all the different optimisation strategies, so what has been attempted is to categorise them and then comment on the categories' suitability for the optimisation problem. Schwefel [27] provides a reasonably comprehensive overview of different optimisation techniques.

4.4.1 Classical analytical method

If the cost function can be expressed as a functional relationship then the following could be done: calculate the partial derivatives of the function, set them equal to zero and solve to find the turning points (where the partial derivatives will be zero). These are necessary conditions for the existence of a minima. The function can then be evaluated at all these points to find the point that results in the smallest evaluation. This is then the global minimum.

This method is ruled out because there is no way to express the cost function as a functional relationship.

4.4.2 Exhaustive search

The following simple search method is guaranteed to find the values of the independent variables, to within a predefined quantisation error, that will result in the optimum value of the cost function.

Define a step size according to the acceptable quantisation error in the location of the global minimum. Divide up the search space into hypercubes of width step size. Evaluate the cost function at the corners of these hypercubes. The point that evaluates to the lowest value is considered to be the global minimum. Fig 4.3 shows an exhaustive sampling of the search space of two independent variables.

This method would require too many evaluations for the human to perform. This is due both to the large size of the search space and the fact that humans are only able to perform very

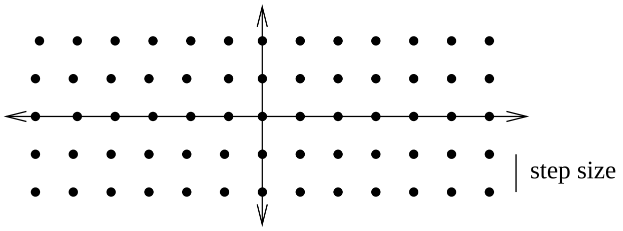


Figure 4.3: The sampling of the search space in an exhaustive search.

few evaluations.

4.4.3 Hill climbing methods

Here the optimisation task is discussed in terms of the finding of maxima, because the analogy here is of climbing upwards until the top of a mountain is found. The problem with these methods as a whole is that only one mountain is climbed at a time and and the algorithms get stuck on local optima.

Methods using gradient

These include two families of algorithms [25, page 293]. The first goes under the name of the *conjugate gradient methods*. This family is typified by the *Fletcher-Reeves algorithm* and the *Polak-Ribiere algorithm*. The second family goes under the name *quasi-Newton* or *variable metric* methods. This family is typified by the *Davidson-Fletcher-Powell (DFP)* algorithm and the *Broyden-Fletcher-Goldfarb-Shanno (BFGS)* algorithm.

These methods are ruled out because there is no way to obtain the gradient of the cost function that exists in the witness's mind.

Methods not using gradient

Downhill simplex method

Nelder and Mead's downhill simplex method is a slow but often extremely robust method of finding a minimum [25, page 305] (once again we are talking in terms of minima because of the algorithm's name and analogy). A simplex is the geometrical figure consisting, in T dimensions, of $T+1$ points (or vertices) and all their interconnecting line segments. The way the algorithm works is by moving the vertices of the simplex through the search space trying to find a minima. The cost function is evaluated at each of the vertices and the vertex that evaluated the highest is moved through the opposite face of the simplex to a lower point.

Using the simplex method for this problem would mean that the witness would be shown faces corresponding to the vertices of the simplex and the witness would have to identify the face that looked the least like the perpetrator's face. Identifying a the face that looks the least similar is not a task that the witness could perform well. This rules out the downhill simplex search.

Stochastic hill climbing

Stochastic hill climbing is a simple iterative optimisation algorithm that outperforms more complicated evolutionary optimisation algorithms on certain types of problems [2].

```
currentPos ← randomly generate solution vector
Best ← evaluate(CurrentPos)

loop NO_ITERATIONS
    testPos ← FlipRandomBit(CurrentPos)
    if(evaluate(testPos) > Best )
         Best ← evaluate(testPos)
         currentPos ← testPos
    end if
end loop
```

Figure 4.4: The basic stochastic hill climbing algorithm.

Fig 4.4 shows the basic stochastic hill climbing algorithm, it works as follows: define a coding of the solution to the problem in the form of a bit-string as described previously. Randomly populate the bit-string with 0 and 1, each having equal probability. Call it `currentPos` and evaluate it. Flip a bit in a randomly chosen position in `currentPos` and call this new bit-string `testPos`. If `testPos` evaluates to a better solution than `currentPos` then let `currentPos = testPos`, else let `currentPos` remain as is. Generate a new `testPos` vector from `currentPos` as before. Repeat until `NO_ITERATIONS` have passed.

The algorithm can be restarted at a random location many times, with the highest evaluating solution from all the starts being the final solution. The algorithm would now be called *multiple restart stochastic hill climbing.*

An obvious improvement to the basic algorithm shown in Figure 4.4 maintains a list of the positions of the bits which were flipped without improvement for each `currentPos`. This is used to stop the same `testPos` from being evaluated more than once for each `currentPos`.

Another variation is to make `currentPos = testPos` if `testPos == currentPos`[1]. This will stop the algorithm from getting stuck when it reaches a plateau. Instead it will wander around the plateau trying to find a higher evaluating position.

The stochastic hill climbing algorithm could be modified so that it would generate a population of trial solutions and then the witness could choose the best match. This would be done by generating a face corresponding to the flipping of each bit in `currentPos` and then updating `currentPos` with the face chosen as the best match by the witness.

The problem associated with this would be that the witness would be exposed to too many faces at a time. If 7 bits are chosen to represent each eigenface coefficient and the faces are built up using 50 eigenfaces each, this would result in $7 * 50 = 350$ bits in `currentPos`. This means that the witness would have to be shown 350 face images at a time, to choose the best match from. This is too many. Also, because of the noise in the evaluation of the cost function there will not be nice smooth hills to climb and the hill being climbed might be a local optimum.

4.4.4 Evolutionary methods

Evolutionary optimisation strategies are strategies that imitate the principals of evolution and natural selection for solving optimisation problems. Evolutionary methods can be grouped into 3 fields that were developed independently from each other. These fields are *evolution strategies* introduced by Rechenberg and developed by Schwefel [27], *evolutionary programing* started by Fogel, Owens and Walsh and *genetic algorithms* invented by Holland. These fields were all started in the 1960's [21, page 2].

Evolutionary strategies try and evolve a population of candidate solutions that will evaluate to the minimum or near minimum of the cost function. The evolution of the population towards the minimum is done through analogies to natural genetic variation and natural selection.

In the last several years there has been widespread interaction among researchers studying various evolutionary computation methods and the boundaries between genetic algorithms, evolution strategies and evolutionary programing have broken down to some extent [21, page 3].

The only evolutionary system that only needs to know which is the highest evaluating trial solution in the population (as put forward in section 4.2.1) appears to be *population based incremental learning* (PBIL). This is a little-known reformulation of the genetic algorithm, that has been shown to outperform the genetic algorithm on some optimisation problems [3, 2].

[1] = symbolises assignment, while == symbolises comparison

What follows here is an explanation of the genetic algorithm because it is similar to population based incremental learning and because Johnston's face recall system uses it. Population based incremental learning is then introduced to show its position within the hierarchy of different optimisation algorithms. Population based incremental learning is presented fully in section 4.5.

Genetic algorithms (GAs)

The following is the algorithm for a standard GA [21, page 10].

1. Start with a random population of n sample solutions. Each of these solutions is created by forming a bit-string as explained in section 4.3 and randomly placing 1 or 0 with equal probability in each position of the bit-string. The bit-strings are called chromosomes.

2. Calculate the fitness of each sample solution in the population by evaluating the cost function at the position it encodes, as specified in section 4.3.

3. Repeat the following steps until n offspring have been created.

    (a) Select a pair of parent chromosomes from the current population, the probability of selection being an increasing function of fitness of the chromosome. The same chromosome can be selected more than once to become a parent.

    (b) With probability p_c (the crossover probability) cross over the pair at a randomly chosen point (chosen with uniform probability) to form two offspring. This is called one point crossover as shown in Figure 4.5. If no crossover takes place, form two offspring that are exact copies of their parents.

    (c) Mutate the two offspring at each position with probability p_m (the mutation probability) and place the resulting chromosomes in the new population. The mutation is performed by flipping the bit at the position to be mutated.

4. Replace the current population with the new population.

5. Until some terminating condition is met, goto step 2.

The terminating condition could be a limit on the number of generations the GA is to run for, or it could be when the GA has converged to a certain degree. Convergence occurs when all the chromosomes in the population become almost identical. This will result in the children looking very much like their parents and therefore each generation being similar to the previous. A GA is typically iterated for hundreds or thousands of generations.

	Parent A	0000000 00000
One Point	Parent B	1111111 11111
Crossover		^
	Child A	0000000 11111
	Child B	1111111 00000

Figure 4.5: Demonstrating one point crossover.

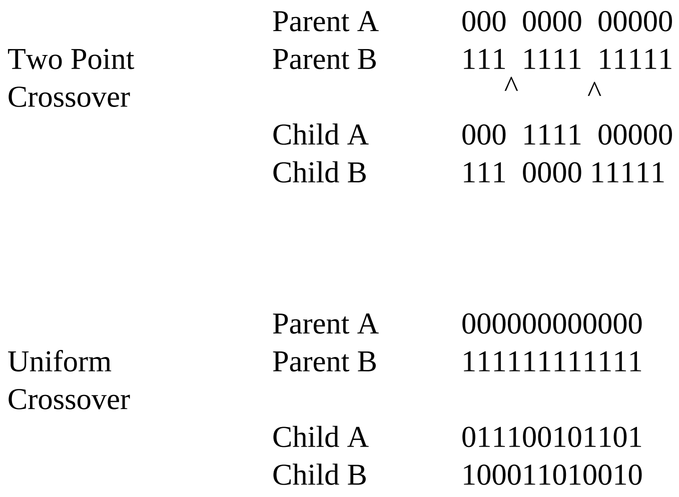
Due to the random factors involved in producing children chromosomes (step 3b above), the children may, or may not, have higher fitness values than their parents. However, because of the selective pressure applied through a number of generations, the overall trend is towards higher fitness chromosomes [3]. This process of survival of the fittest can over many generations result in the optimal or a close to optimal point being found in the cost function.

The population size n, crossover probability p_c and mutation probability p_m are all parameters that have to tuned according to the fitness function and the type of GA used. This is a trial and error process and unfortunately the speed of convergence of the GA onto the minimum is dependent on the settings of these parameters. Mutations are used to help preserve diversity in the population or, stated differently, to stop premature convergence.

There are many variations of the GA adapted for different problems. Amongst these variations are different types of crossover and mutation to the ones described above. Two-point and uniform crossover are shown in Figure 4.6. In two point crossover two points are randomly chosen and the contents of the chromosomes between these points are swapped to generate the two children. In uniform crossover the parent for each bit position is chosen randomly for the first child, the bits that were not used by the first child are then used to create the second child. The selection of parents to use in the production of children is also a process that has many different strategies.

Another variation is to use elitist selection. If it is used the best chromosome from generation G is automatically carried to generation $G+1$. This will cause the fitness of the best solution in each generation to monotonically increase. Without elitist selection the best solution in generation $G+1$ could be worse than the best solution in generation G.

Johnston's system [5, 15] uses a genetic algorithm to search for the optimum parameters for its face model. The witness evaluates each face in the population on a similarity scale according to its similarity to the perpetrator. This is the fitness associated with that member of the population. He uses a population of size 20 and claims to get good likenesses in as few as 10 generations [5]. This still necessitates the user having to provide a similarity measure for 200 face images.



	Parent A	000 0000 00000
Two Point Crossover	Parent B	111 1111 11111
		^ ^
	Child A	000 1111 00000
	Child B	111 0000 11111
	Parent A	000000000000
Uniform Crossover	Parent B	111111111111
	Child A	011100101101
	Child B	100011010010

Figure 4.6: Demonstrating two point and uniform crossover.

It has been decided (see section 4.2.1) that it would be easier for the witness to choose a winner (most similar face) from the population as opposed to evaluating each face in the population. For this reason the GA has been ruled out for use in the face recall system to be developed.

Population based incremental learning (PBIL)

PBIL is an evolutionary optimisation algorithm that only needs to know the winner of the population in order to generate a new population. This, the fact that it has been shown to outperform the GA, and simplicity to implement, are the reasons why PBIL was chosen as the optimisation algorithm for the face recall system. PBIL is explained in full in the next section, Section 4.5.

4.5 Population based incremental learning (PBIL)

PBIL is an optimisation algorithm that was developed by Baluja in 1994 [1, 3, 2]. This section presents his algorithm. It starts with a conceptual overview and then explains how this is implemented. Finally a comparison is made with the GA, since the underlying mechanisms are based on similar principles.

4.5.1 Conceptual overview

The following is a brief conceptual overview of how PBIL works:

1. Set up a random number generator with an initial uniform *probability density function* (pdf) across the space to be optimised.
2. With the random number generator, generate a population of trial points and evaluate each one.
3. Adjust the pdf of the random number generator to slightly favour the trial point which evaluated to be the best.
4. Until some terminating condition is met, return to step 2.

4.5.2 Implementation

Figure 4.7 shows how the above conceptual overview is implemented. The pdf is maintained in a probability vector PV of length LENGTH, the length of the encoded bit-string (see Section 4.3). Each position in PV holds a real number between 0 and 1, representing the probability of a 1 being generated when the PV is sampled at that position. The sampling process is shown in Figure 4.8. What is done in Figure 4.8 is to generate a random number between 0 and 1 for each position of the PV. If the random number is less than the corresponding number in the PV, then the corresponding value in the returned vector will be 1, else it will be 0. This is the implementation of the random number generator with the pdf that was presented in the conceptual overview above.

During initialisation PV is set to have 0.5 in all its positions. This means that when PV is sampled in this state, there is equal probability of 0 or 1 being generated for each position in PV. Thus the initial sampling of the search space will be uniform.

Generate NO_TRIALS trialVectors by sampling the PV. After each trialVectors is created evaluate it; this is done by first decoding the bit-string into the decimal variables and then feeding the variables into the cost function. Let bestVector be the trialVector that evaluates the best. Update the probability vector so that in the next generation of trials the probability of trialVectors looking more like bestVector is increased. This is done by increasing the positions in the PV that correspond to 1's in bestVector and decreasing the positions that correspond to 0's. The amount that PV is increased and decreased is relative to LEARNING_RATE.

The mutation in Figure 4.7 is to try and stop premature convergence on a particular point in space. Premature convergence could result in the search space not being properly explored and the algorithm converging on a local minimum. The mutation is performed after PV has been updated. Each bit in PV has a probability of MUTATION_PROBABILITY of being shifted. The bits that do get shifted are done so by MUTATION_SHIFT either up or down, each direction having a probability of $\frac{1}{2}$.

```
****** Initialise the Probability Vector ******
for i = 1 to LENGTH
    PV[i] = 0.5;
end for

while(NOT termination condition)

    ****** Generate Trials ******
    for i = 1 to NO_TRIALS
       trialVectors[i] = sample(PV);
       evaluations[i] = evaluateSolution(trialVectors[i]);
    end for
    bestVector = findBestVector(trialVectors, evaluations);

    ****** Update the Probability Vector towards the best solution ******
    for i = 1 to LENGTH
       PV[i] = PV[i] * (1.0 - LEARNING_RATE) + bestVector[i]*LEARNING_RATE;
    end for

    ****** Mutate the Probability Vector ******
    for i = 1 to LENGTH
       if (random(0,1) < MUTATION_PROBABILITY)
          if(random(0,1)>0.5)
              mutateDirection = 1;
          else
              mutateDirection = 0;
          end if
          PV[i] = PV[i]*(1.0 - MUTATION_SHIFT) + mutateDirection * MUTATION_SHIFT
       end if
    end for

end while
```

Figure 4.7: Pseudo-code for population based incremental learning.

Once the `PV` has been updated and mutated, a new population of `trialVectors` is created and the process repeats itself. The above iteration carries on until some terminating condition is reached. This could be after a maximum number of trials, or after the evaluation of `bestVector` is sufficiently high, or when the population generated by PBIL has converged to a certain degree.

The probability vector can be viewed as a prototype vector for generating trial solution vectors which will have high evaluations. The knowledge of which trial vectors will have high evaluations is gained from the feedback obtained from the previous generations.

As the number of generations increase, the values in the `PV` should converge on values close to 1 or 0. This will result in 1's and 0's respectively being generated with a high probability in the corresponding positions of the `trialVectors`.

```
sample(PV)
    for i = 1 to LENGTH
       vector[i] = PV[i] < random(0,1);
    end for
    return vector;
```

Figure 4.8: Pseudo code for the function `sample` used in Figure 4.7. The function `random(0,1)` generates a random number between 0 and 1 with a uniform probability density distribution.

4.5.3 Comparison of PBIL to the GA

Population based incremental learning (PBIL) is a simplification of the genetic algorithm, which matches or outperforms the standard GA on problems commonly explored in GA literature [1, 2]. This has also been shown to be true for a problem designed to be easy for the purported mechanisms of the traditional GA [3]. The results achieved by PBIL are more accurate and are attained faster than a standard genetic algorithm, both in terms of the number of evaluations performed and the number of processor clock cycles used. The reduction in clock cycles is due to the simplicity of the algorithm [1].

The GA uses a population to maintain the information learnt about the points in the search space sampled in previous generations and then utilises this information by using operators on the population to produce offspring. The learning is achieved by using the fittest members of the population with a higher probability when generating offspring. Thus as the generations increase the population should consist more and more of members in areas of the search space that have been evaluated to have a high fitness.

PBIL, instead of using a population to maintain the information learnt about the points in the search space already sampled, uses a probability vector which is in effect a pdf. After each generation the pdf is updated, so that the likelihood of the winner of that generation being generated when the pdf is sampled is increased. The pdf will still reflect the changes to it from previous generations, this is how the information about the search space is learnt and kept. The using of a pdf to maintain this information is a more direct approach and is easier to implement than that of the GA.

Another advantage of PBIL is that it is less sensitive to its free parameter settings than the GA is to its own.

Chapter 5

Creating and Analysing the Face Space

This chapter analyses the search space to be navigated by the PBIL algorithm in its search for a good likeness. This is to ascertain the difficulty of the task and to find ways in which to make it easier. It also demonstrates the concepts presented in Chapter 3.

The chapter starts by describing how the eigenfaces, used in the developed system, are generated. This includes a description of the standardisation process and why it is needed. Then the face reconstruction capabilities of the generated eigenfaces are explored in order to assess whether a good likeness can be generated if the optimal coefficients are found. Finally the smoothness of the cost function and the distribution of the faces in face space are investigated.

5.1 Defining/creating the face space used in the study

5.1.1 The face images used

Figure 5.1 shows 10 face images from a set of 50 face images which were obtained from *The Psychological Image Collection at Stirling (PICS)*, maintained by the University of Stirling Psychology Department. The image collection is available from the Internet via *http://pics.-psych.stir.ac.uk/*.

As can be seen in Figure 5.1 these are all grey-scale images of young British males looking directly into the camera. The background around the neck has been blackened with the use of a black piece of material.

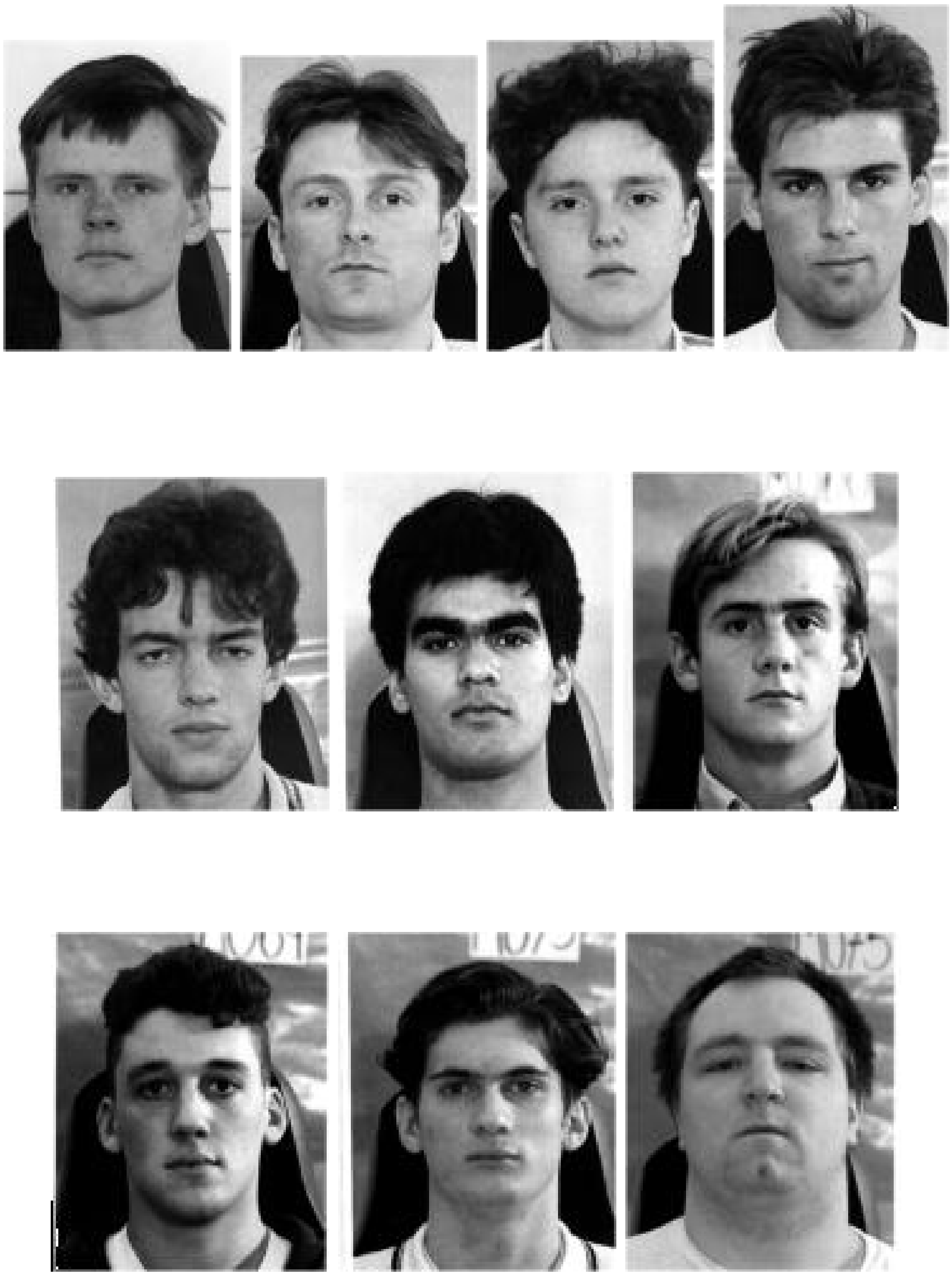


Figure 5.1: Examples of the face images obtained from the University of Sterling

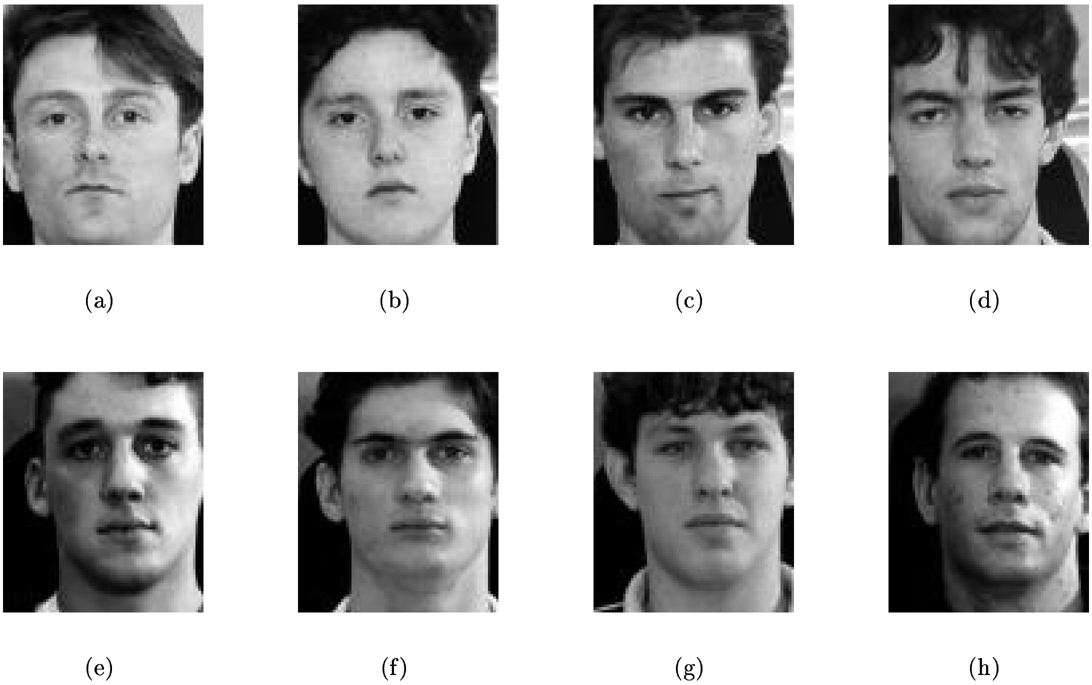
.

5.1.2 Standardising the face images

For the eigenfaces technique to work the face images, on which principal component analysis will be performed, have to be standardised so that each face is in the same position within the image. The hair also has to be removed — this is because there exists too much variation in hair-styles and colour to be captured by a small set of eigensurfaces.

Demonstration of the need for standardisation

The reason for requiring standardisation can be demonstrated with the extreme case where the set of face images are divided into two equal size groups. The one group has the faces positioned in the bottom left hand corner of the image, samples of this group are shown in Figures 5.2 a-d. The other group has the faces positioned in the top right hand corner, samples of which are shown in Figures 5.2 e-h.



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

(e) (f) (g) (h)

Figure 5.2: Samples of the shifted face images. The images in the top row are a sample of the 25 images with the face shifted to the bottom left of the image. The images in the bottom row are a sample of the 25 images with the face shifted to the top right of the image.

These two groups of face images will be used to show that if the images are not standardised then eigenfaces generated from them will not just describe the variations within the faces, as would be desired. They must then also describe the variations due to the different position of the faces within the image.

If principal component analysis is applied to the whole set of images at once the eigenfaces generated are those shown in Figure 5.3. The value associated with each pixel position in the eigenfaces can be both positive or negative. The colour-map that maps the values to the colours is a smoothly increasing range of grey-scale values from black to white. Black in the colour-map is associated with the most negative number in the eigenface, while white is associated with the most positive number in the eigenface. The mid-level gray is associated with 0. Therefore each eigenface deviates from the uniform gray where there is variance among the set of training face images. The eigenfaces can be considered to be some sort of map of the variations between face images.

In Figure 5.3 it can be seen that the average face is made up of two different average faces, one in the bottom left hand corner and one in the top right hand corner. The first eigenface mainly describes the variance due to the shift of the faces from the bottom left hand corner to the top right in the images. This can be seen in the white rectangles at the top and the right of the first eigenface.

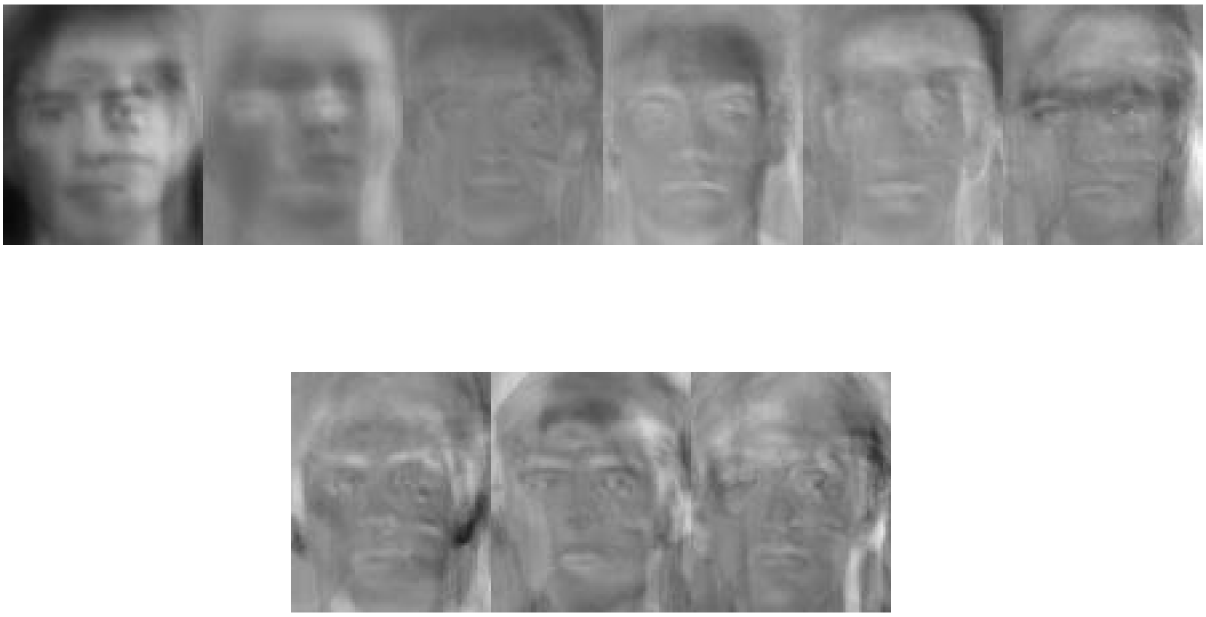


Figure 5.3: The average face and the first 8 eigenfaces generated from the shifted face images. Examples of the shifted face images are shown in Figure 5.2. The order of the images is left to right, top to bottom.

That the first eigenface describes the shift in the face positions is substantiated in Figure 5.4.

This plot shows the coefficients for the first eigenface plotted against those for the second eigenface. These coefficients were obtained by projecting the shifted set of face images onto the eigenfaces using (3.31). (These are the coefficients that would be used in the eigenface reconstruction of the shifted face images, an example of which is shown in Figure 5.5.) The faces in the shifted left-down set have negative coefficients for the first eigenface and those in the shifted top-right set have positive coefficients for the first eigenface. This clearly demonstrates that the first eigenface describes the shift.

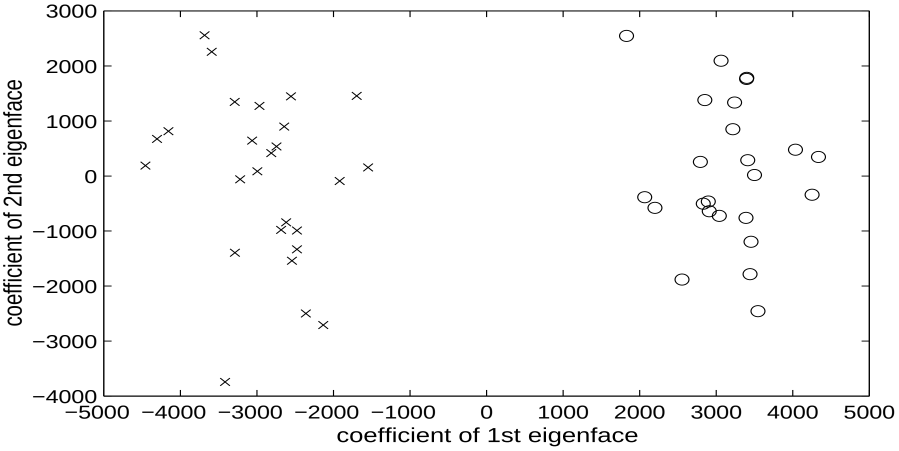


Figure 5.4: The coefficients of the second shifted eigenface versus those of the first. The coefficients are those obtained from projecting the 50 face images onto the face space. The x's show the 25 face images that are shifted left-down. The o's show the 25 face images that are shifted right-up.

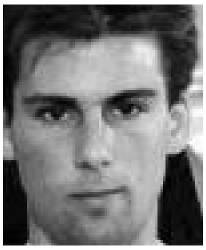


Figure 5.5: Reconstruction of the face image in Figure 2(c) using the average face and all the eigenfaces derived from the shifted faces. The face image being reconstructed was in the training set.

Figure 5.6 shows the results of a reconstruction of a face image from the set of faces on which principal component analysis was performed. This was done using only the average face and the first 10 eigenfaces. Figure 5.12 is also a reconstruction with the average face and the first 10 eigenfaces but these were calculated on standardised face images. By comparing the two figures it can be noted that the reconstruction in Figure 5.12 is much better and that there

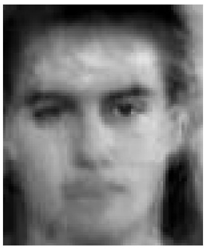


Figure 5.6: Reconstruction of face image in Figure 2(c) using only the average face and the first 10 eigenfaces derived from the shifted faces. The face image being reconstructed was in the training set.

is not as much blurring in it. This is due to the fact that if the face images are standardised there is less variation between them and the variation that does exist, is due to the differences between the face and not their positions. If there is less variation, then the variation can be described in fewer eigenfaces.

Another issue is that the face recall system could try and generate a face image from any point in face space. It can be derived from Figure 5.4 that not all points in face space will result in face images. The empty space in the face space in Figure 5.4 will result in some kind of mixture between a top right and a bottom left face.

Standardising the face images

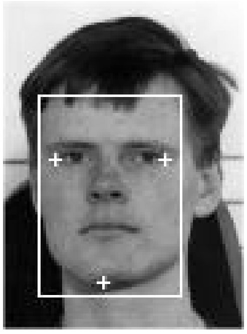
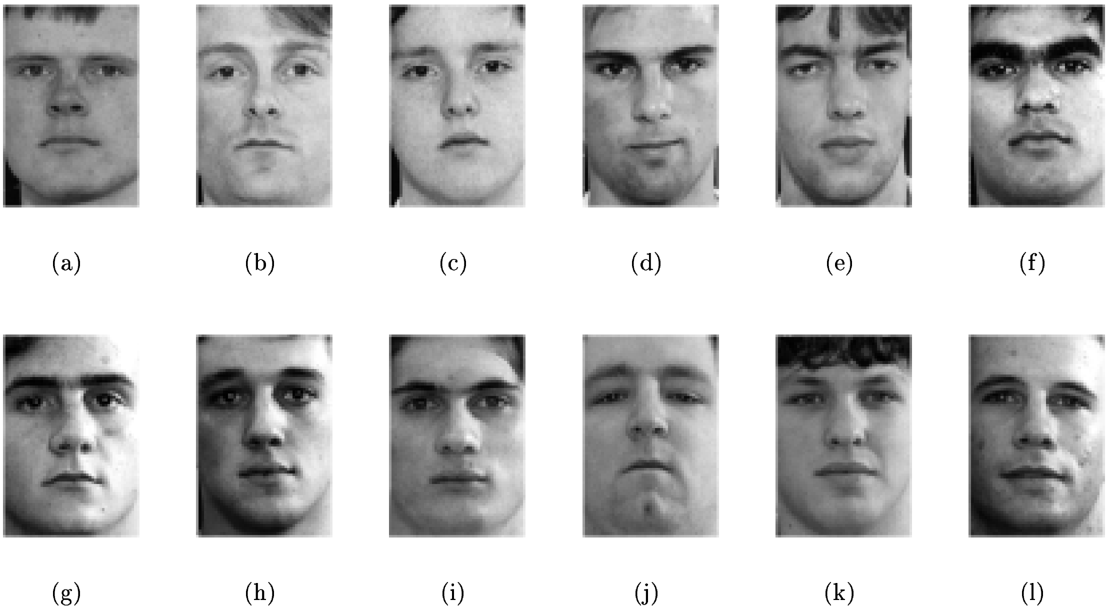


Figure 5.7: Showing how images in Figure 5.1 are cropped to obtain those in Figure 5.8. The crosses are points specified by a user and the box shows where the image is to be cropped.

The 50 face images obtained from PICS were standardised in the following manner: The outside corners of the eyes and the bottom of the chin were specified manually. These points are shown with the crosses in Figure 5.7. The drawn-in box shows where the image was cropped relative to the crosses. The sides of the vertical edge of the cropping rectangle are 22 pixels to left and right of the left and right eye X's respectively. The top horizontal edge is 100 pixels above the average y-position of the eye crosses and the bottom horizontal edge is 10 pixels below the y-position of the chin cross. The images were then resized to 51 by 76 pixels (width by height). Figure 5.8 shows some of the face images after being cropped and resized.



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

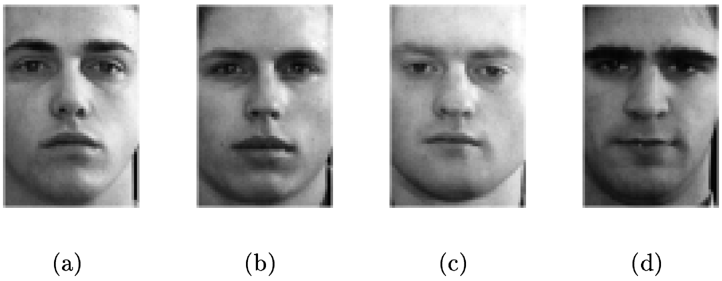
(g) (h) (i) (j) (k) (l)

Figure 5.8: Examples of the standardised face images used to obtain the eigenfaces.

5.1.3 Performing principal component analysis

Principal component analysis was performed on a set of 46 of the standardised face images, examples of which are shown in Figure 5.8. The whole set of standardised face images is shown in appendix A. The four faces shown in Figure 5.9 were left out, so that the reconstruction capabilities of the face space could be tested on face images not in the set on which principal component analysis was performed. The reason for this is that the developed system will be expected to construct faces that were not in the set of faces from which the eigenfaces were calculated. Using terminology from pattern recognition, the set of 46 faces on which principal component analysis is performed will be called the *training set* and the 4 faces left out will be called the *test set.*

There are two factors that can make the system perform worse in generating faces in the test set compared to those in the training set. Firstly, it is not known how well a new face can be reconstructed with the eigenfaces. Secondly, face images created with the eigenfaces will probably have a disposition to be likenesses to the faces in the training set. This would be



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

Figure 5.9: The test set of face images. Face images excluded from the set of images on which the principal component analysis was performed

.

due to large parts of the face space resulting in face images that are likenesses to the faces in the training set.

Figure 5.10 shows the eigenfaces generated from applying principal component analysis on the set of 46 standardised face images. In comparison to the eigenfaces of the shifted faces in Fig 5.3, it can be seen that the areas of variance are more localised. The deviations from the average grey (towards black and white) are localised to the positions of the facial features and the positions of these are generally much the same, in the eigenfaces and the average face.

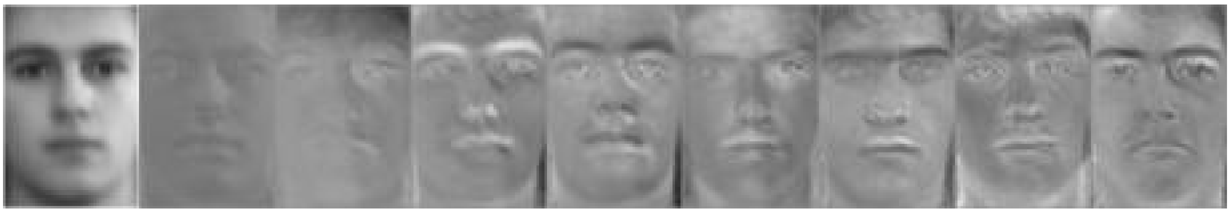


Figure 5.10: The average face and the first 8 eigenfaces generated from applying principal component analysis on the training set of 46 face images.

5.2 The reconstruction capabilities of the eigenfaces

It is important to know how well the eigenfaces reconstruct face images not in the training set, since this is what a face recall system will be expected to do. This is tested in Section 5.2.2 .

Section 5.2.1 shows the reconstruction capabilities of the eigenfaces on face images in the training set. Though this has no relevance on the face recall system it is included to demonstrate eigenface reconstructions of images in face space and other DFFS issues.

The reconstructions in Sections 5.2.1 and 5.2.2 that only use the first few eigenfaces are there to demonstrate the effects of increasing the DFFS.

5.2.1 Reconstructing face images in the training set

This subsection shows that face images in the training set can be perfectly reconstructed if all the eigenfaces are used.

Using the average face and all the eigenfaces

If a face image in the training set is to be reconstructed with all its $M = 46$ associated eigenfaces, then the reconstruction will be perfect, because the face space includes all the face images that were used in the principal component analysis. This is can be seen in Figure 5.11 which is a reconstruction of Figure 5.8(d) — the two images are identical.

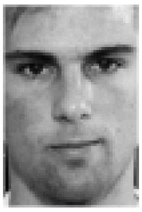


Figure 5.11: Reconstruction of the face image in Figure 5.8(d) using the average face and all the eigenfaces.

Using the average face and only some of the eigenfaces

If a face image in the training set is to be reconstructed using only some of its associated eigenfaces, then the reconstruction is unlikely to be perfect. This is because the face space is now smaller, introducing a non zeros DFFS between the image to be reconstructed and its projection onto the now smaller face space. The reconstruction of Figure 5.8(d) using the average face and the first 10 eigenfaces is shown in Figure 5.12. It can be seen that even though the two faces images are not identical they are similar.

5.2.2 Reconstructing face images in the test set

This subsection shows that the eigenfaces used in this thesis can be used to reconstruct face images in the test set adequately if all the eigenfaces are used. This is used as an indication that face reconstruction with eigenfaces can be used to reconstruct new face images.

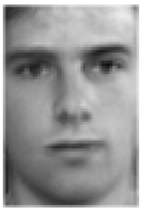


Figure 5.12: Reconstruction of face image in Figure 5.8(d) using only the average face and the first 10 eigenfaces derived from the training set

Using the average face and all the eigenfaces

If a face image in the test set is to be reconstructed with all the M eigenfaces then the reconstruction will not be perfect. This is due to the fact that the face image in the test set is almost certainly a point in the original image space that is not included in the face space defined by the eigenfaces. The reconstruction of Figure 5.9(b) using the average face and the first 10 eigenfaces is shown in Figure 5.13. It can be seen that the thickness of the lips of the person in Figure 5.9(b) have not been properly reproduced in Figure 5.13 — here they are thinner. There also exists an overall fuzziness in Figure 5.13. There still exists a remarkable similarity between the faces, which is an indication that face reconstruction with eigenfaces has got good generalising capabilities to faces not in the training set.

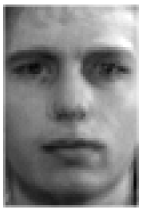


Figure 5.13: Reconstruction of face image Figure 5.9(b), from the test set, using the average face and all the eigenfaces derived from the training set.

This statement is further validated by the reconstructions shown in Figure 5.14. These are the reconstructions of the rest of the faces in the test set shown in Figure 5.9. Once again a good similarity exists.

Using the average face and only some of the eigenfaces

If a face image in the test set is to be reconstructed with only some of the eigenfaces, then the reconstruction will poorer than a reconstruction with all the eigenfaces shown in Figure 5.13. This is because on top of the fact that the face image does not fall into the full face space defined by the $M = 46$ principal component axes, we are now using a subspace of the

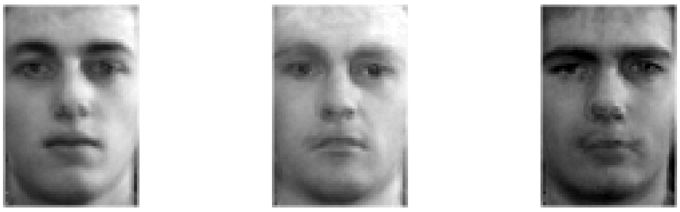


Figure 5.14: Reconstruction of the rest of the face images in the test set, shown in Figure 5.9. This was done using the average face and all the eigenfaces derived from the training set.

face space. Thus test set face images will be even further outside the subspace of face space than they were outside the full face space. Figure 5.15 shows the reconstruction of Figure 5.9(b), from the test set, using the average and the first 10 eigenfaces. It can seen that the fuzziness has increased in comparison to the reconstruction, with all the eigenface, in Figure 5.13, but the likeness is still strong.

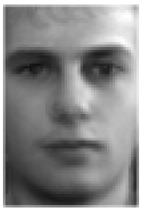


Figure 5.15: Reconstruction of the face image in Figure 5.9(b), from the test set, using only the average face and the first 10 eigenfaces derived from the training set

5.3 Moving through face space

In order to get an idea of what the faces in the face space looked like and how they changed when the space was traversed, the coefficients for two face images, in the test set, were calculated and then the face images 'in between' were generated. This was done by traversing equal distances along the straight line between the calculated coefficients of the two face images and generating images with the coefficients at these intermediate points. The results of this can be seen in Figure 5.16. Figure 5.16 shows that this traversal of the face space could be used as a morph from one face image to another.

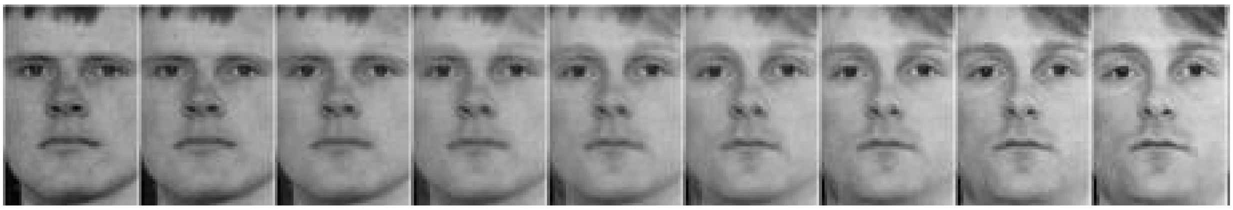


Figure 5.16: Images generated with coefficients obtained by traversing the face space in a straight line from one face image to another.

5.3.1 The shape of the cost function

In Figure 5.16 it can be seen that there is a smooth transition from one face to another, that it is easy to note the similarity between a face and its neighbours and that there are no faces in between that are significantly different from those enclosing it. These factors all contribute to making the witness' cost function reasonably smooth. The smoother the cost function, the easier it is for an optimisation algorithm to traverse to its optimum or a close-to optimum point.

5.3.2 The shape of the cost function of Johnston's system

Johnston's face recall system as described in section 2.1.3 builds up its faces through positioning a forehead, a set of eyes, a nose, a mouth and a chin together. The dimensions of the face space of Johnston's system are then an axis for each of the facial features described above and an axis for the positions of each of the facial features.

To achieve this, each forehead in the set of foreheads must have a number associated with it, so that the foreheads can be ordered along the forehead axis. This ordering must result in some sort of progression from the forehead at the start of the axis to the forehead at the end of the axis. The same must be done for the other facial features. There is no natural ordering of the features and it must be done manually based on the subjective decisions of the sorter.

This, combined with the fact that the there are a limited number of each of the facial features with no intermediate versions, stops the cost function from being smooth. This makes it harder for the genetic algorithm to find the optimum or a close-to optimum point.

5.3.3 Artifacts due to the non alignment of facial features

In the middle face images in Figure 5.16 it can be seen that there are double eyebrows. This is due to the eyebrows of the face image on the right hand side being slightly higher than those on the left hand side. So as we traverse the face space from one face to the other, there has to be some form of transition. This transition results in the double eyebrows.

The reason for there being eyebrows in different positions is that the images were standardised to have the outside part of each eye in the same position. Thus even though the eyes are set to the same position, the eyebrows can be in different positions if they are in different positions relative to the eyes. The face on the right has eyebrows which are higher relative to the eyes than those of the face on the left. For the same reason, other facial features can be in different positions in the standardised training set. This is the cause of the other artifacts that can be seen in the generated face images displayed in the GUI, shown in Figure 6.5.

These double features in the generated images could be removed by using more landmark points in the standardisation process. The problem with this is that all generated faces would have their features in the same same position and the faces would be the same shape. The information held in the position of the facial features and in the shape of the face would then be lost to the system. All generated faces would be of the same overall shape.

5.4 Distribution of faces in face space

Moghaddam and Pentland [22] state that, in their experience, the distribution of faces in face space is accurately modelled by a single Gaussian distribution. For a distribution to be Gaussian each variable must have a Gaussian distribution and all linear combinations of the variables must be Gaussianly distributed. This is not readily tested because it is impractical to test an infinite number of linear combinations of variables for normality [31]. Even though it is not possible to verify that the distribution is Gaussian, by analysing the distribution it is possible to determine if the distribution seems to approximate a Gaussian distribution. This was done by looking at the distribution of the coefficients of the eigenfaces used to reconstruct the images in the training set.

Firstly the distribution of the coefficients of each eigenface was plotted in a histogram. Figure 5.17 shows the distribution of the coefficients of the first eigenface. It can be seen that the distribution seems to approximate a Gaussian distribution. The distributions of the coefficients of the other eigenfaces look on average similar.

Secondly, scatter-plots were generated showing the distribution of the coefficients with respect to each other. Figure 5.18 shows the distribution of the coefficients of the first eigenface plotted against those of the second eigenface. The distribution looks like it is a sampling of a Gaussian distribution. Other combinations of the coefficients were looked at and they had on average the same spread of points.

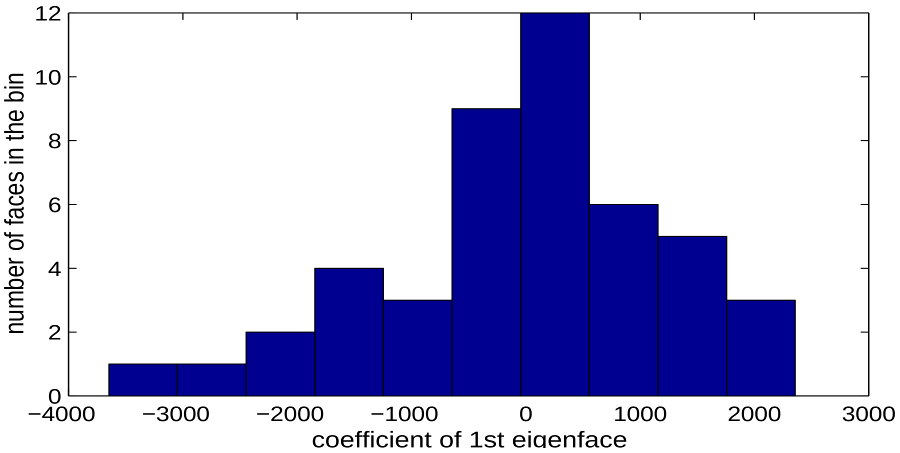


Figure 5.17: Histogram showing the distribution of the coefficients of the first eigenface, these were obtained by projecting the images in the training set onto it. It seems to approximates a normal distribution.

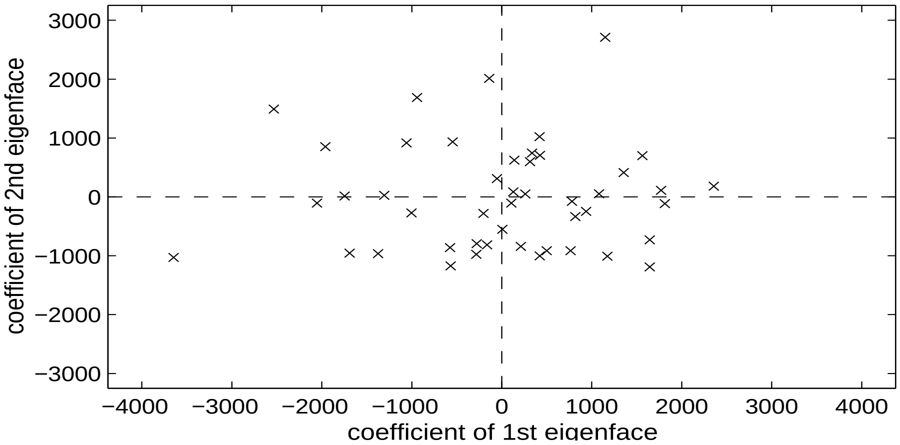


Figure 5.18: Scatter plot showing the distribution of the coefficients of the first and second eigenfaces. These were obtained by projecting the images in the training set onto them.

Although the it cannot be confirmed that the distribution faces in face space is a Gaussian distribution, it can be seen from looking at the above plots (Figure 5.17 and Figure 5.18) that a ‘best guess’ approximation to their distribution would be a Gaussian distribution. If this is assumed, then the best approximation to the true distribution would be the multidimensional Gaussian parameterised by the covariance matrix of the eigenface coefficients. As a result of face space being described in terms of the principal component axes of the face images, the covariance matrix is a diagonal matrix with the eigenvalues going down the diagonal from largest to smallest.

Chapter 6

The Developed Face Recall System

This chapter describes the implementation and use of the developed face recall system. Also included are the results of automated simulations that were performed. Chapter 7 presents the results of trials with humans trying to generate faces.

The chapter starts by explaining why the system was implemented in MATLAB. Then it explains how PBIL was modified to search through a solution space with a Gaussian distribution. The results of running simulations with the computer, instead of a human, choosing the best match face in the population are also presented. These were for preliminary feasibility trials and to find settings for PBIL's free parameters. The chapter ends with simple explanation of how to use the developed face recall system.

6.1 Platform of implementation

The developed face recall system was implemented in MATLAB. This decision was taken because of the powerful mathematical capabilities of MATLAB, due to its easy matrix manipulation and built-in functions. MATLAB is a high-level interpreted prototyping language which enables the rapid development of algorithms and programs. It has a similar structure to compiled procedural languages like C. This makes translation of the MATLAB code into C straightforward — each MATLAB function can be translated into an equivalent C function. This one to one translation of functions simplifies the tasks of debugging and checking for errors, since the new function can be checked against the MATLAB one. There also exists a MATLAB compiler that translates MATLAB code to C++ and the MATLAB C++ math library which is a library of C++ classes which provide the functionality of most of the built in MATLAB functions.

6.2 Transforming PBIL's pdf to start as a Gaussian pdf

The initial sampling of the face space by PBIL is uniform. This is inefficient. We have already noted in section 5.4 that the distribution of face images in face space seems to be a Gaussian with a mean of zero. The algorithm will therefore waste generations learning what we already know. This initial uniform sampling of face space results in some very unface-like 'face images' appearing in the initial population.

To rectify this the random variables generated by the PBIL algorithm are transformed. What this transform does is to make the initial pdf of the random number generator in PBIL Gaussian and then, as the winner of each generation is found, this pdf will get modified accordingly.

To do this the inverse-transform method is used [19, page 242]. The inverse transform method is generally used to generate random numbers with a non-uniform pdf f from a random number with a uniform pdf. Given f's cumulative distribution function (cdf) F and its inverse F^{-1}, to perform the transform the following operation is performed:

1. Generate U, a random number with a uniform distribution between 0 and 1.
2. Set $X = F^{-1}(U)$.

X is now the random number which will have a non uniform pdf f. U will always fall into the domain of F^{-1} which is $[0, 1]$. This is because by definition the range of F is $[0, 1]$.

Figure 6.1 shows a graphical representation of what is done above.

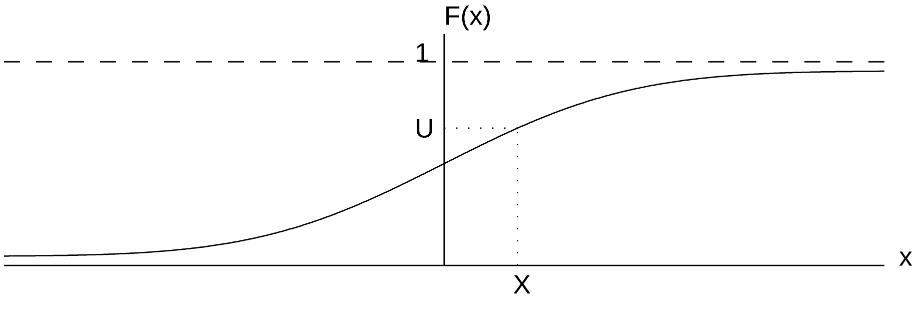
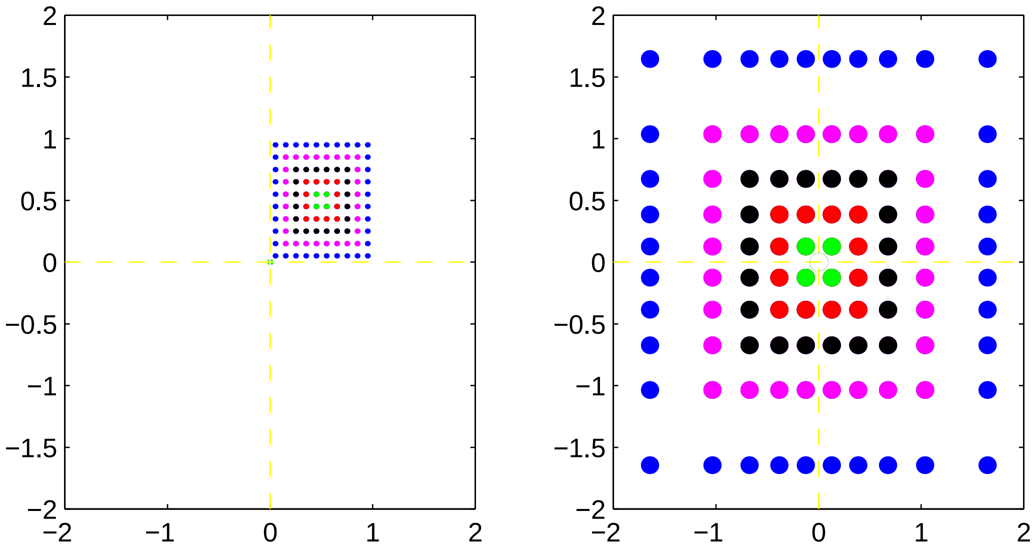


Figure 6.1: Figure showing the inverse transform method. U is the uniform input random variable and X is random variable that has been transformed, $F^{-1}(U)$.

Another way of thinking about the transform in this context is that it is a mapping from one part of a space to another. This mapping is such that a uniform sampling of the space will result in a Gaussian sampling after the transform. This is shown in Figure 6.2, here the inverse transform method was used to transform a two dimensional uniform sampling between zero and one, to a two-dimensional Gaussian sampling of the space. Equation (6.9), which is derived in Section 6.2.1, was used to perform the transform.



(a) A uniform sampling in (0,1)

(b) The transform of the points in (a).

Figure 6.2: The transform of a uniform sampling between zero and one to a two-dimensional Gaussian sampling with a mean zero and standard deviation of one along each axis. This was done using (6.9).

6.2.1 Calculating the inverse transform

The cdf of a Gaussian with mean m and standard deviation σ is

$$F_X(x) = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{x} e^{-(x'-m)^2/2\sigma^2} dx'. \tag{6.1}$$

To obtain F^{-1}, F will first be written in the form of the error function `erf(x)`. Then to obtain the inverse, the inverse error function `erfinv(x)` will be used. `erfinv(x)` is a MATLAB function that does a fast numerical approximation to the inverse of the error function.

The form of the error function is [30, page 746]

$$\texttt{erf(x)} = \frac{2}{\sqrt{\pi}} \int_{0}^{x} e^{-t^2} dt \tag{6.2}$$

Rewriting the Gaussian cdf (6.1) in terms of the error function (6.2) is done as follows. Letting

$$p = \frac{x' - m}{\sqrt{2}\sigma}, \tag{6.3}$$

we have

$$dp = \frac{1}{\sqrt{2}\sigma} dx' \Rightarrow dx' = \sqrt{2}\sigma dp. \tag{6.4}$$

So

$$F_X(x) = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\frac{x-m}{\sqrt{2}\sigma}} e^{-p^2} dp. \tag{6.5}$$

Since we can assume that the mean of the distribution of faces in face space is zero we can let $m = 0$. This is because we have subtracted out the average face before the calculation of the eigenfaces. Then

$$\begin{aligned}
F_X(x) &= \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\frac{x}{\sqrt{2}\sigma}} e^{-p^2} dp \\
F_X(x) &= \frac{1}{\sqrt{\pi}} \int_{-\infty}^{0} e^{-p^2} dp + \frac{1}{\sqrt{\pi}} \int_{0}^{\frac{x}{\sqrt{2}\sigma}} e^{-p^2} dp \\
F_X(x) &= \frac{1}{2} + \frac{1}{\sqrt{\pi}} \int_{0}^{\frac{x}{\sqrt{2}\sigma}} e^{-p^2} dp \\
F_X(x) &= \frac{1}{2} + \frac{1}{2}\texttt{erf}(\frac{x}{\sqrt{2}\sigma}) \\
\texttt{erf}(\frac{x}{\sqrt{2}\sigma}) &= 2(F_X(x) - \frac{1}{2}). && (6.6) \\
& && (6.7)
\end{aligned}$$

Now that the cdf of the Gaussian has been obtained in terms of `erf(x)`, we can use `erfinv(x)` to get the inverse

$$\begin{aligned}
\frac{x}{\sqrt{2}\sigma} &= \texttt{erfinv}(2F_X(x) - 1) \\
x &= \sqrt{2}\sigma\ \texttt{erfinv}(2F_X(x) - 1) && (6.8)
\end{aligned}$$

As shown in Figure 6.1, substitute the random variable U for $F_X(x)$ will give X, the transformed random variable. So the transform to be used is

$$X = \sqrt{2}\sigma \, \texttt{erfinv}(2U - 1). \tag{6.9}$$

6.3 System implementation

The following is a description of how the system is implemented. Section 6.5 gives a description of how the implemented system works from the user's point of view.

6.3.1 General overview of the system implementation

The developed face recall system [26] uses PBIL as an optimisation algorithm to search through a face space defined by a set of eigenfaces, using a human to perform the evaluation of the cost function. The cost function evaluation that the witness performs is to look at the population of face images generated by PBIL and to specify the face that looks most like the perpetrator. The witness specifies the best match by clicking on the face in the GUI shown in Figure 6.5. The population size used by PBIL is *noFaces*, the number of faces shown on the GUI at one time. Since it is assumed that the distribution of faces in face space is a Gaussian the coefficients generated by PBIL are transformed to match this.

6.3.2 Detailed description of the system implementation

Figure 6.3 shows a block diagram of the system implementation. What follows is a description of the individual parts of the system.

PBIL

This is the implementation of the PBIL algorithm described in Section 4.5 with the pseudocode for its implementation shown in Figure 4.7. The implementation used is exactly like Figure 4.7 except that the function call `evaluateSolution(trialVectors[i])`[1] is not made. The task this function performed is now combined into `findBestVector(trialVectors)` (note that the function does not pass the argument `evaluations` as it has not been calculated and is not needed). The dashed box in Figure 6.3 performs the task of the function `findBestVector`. Therefore the output of the PBIL block is `trialVectors`. `trialVectors`

[1] This `font` refers to variables in the PBIL algorithm's pseudo-code, presented in Figure 4.7. While this *font* is for numbers used in the block diagram of the system, presented in Figure 6.3.

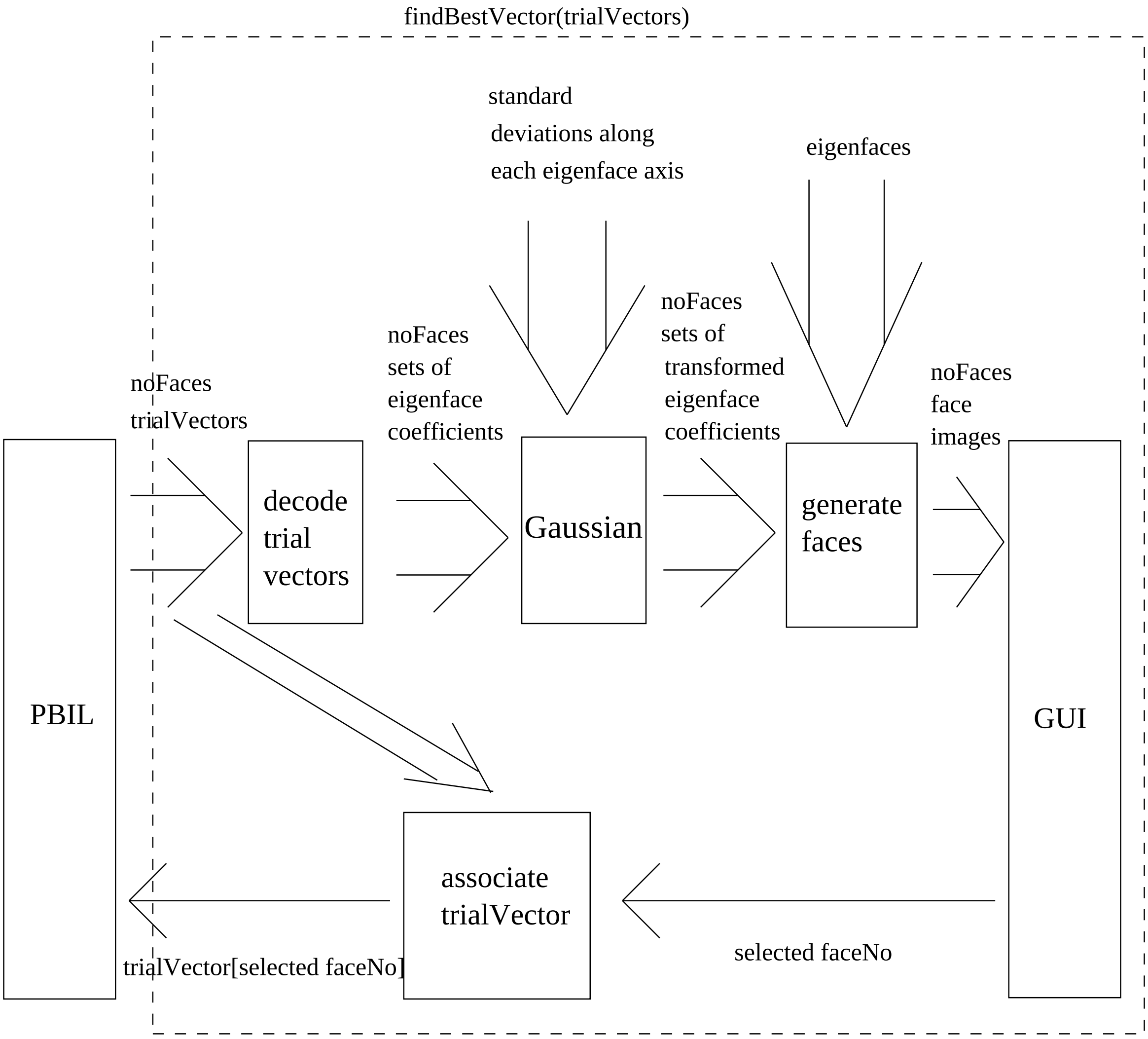


Figure 6.3: A block diagram of the implementation of the developed system.

contains *noFaces* encoded bit-strings. The return from the block `findBestVector` and input into the PBIL block is the trial vector that resulted in the face that the witness selected.

The free parameters in the PBIL algorithm shown in Figure 4.7 are set as follows: `LENGTH`, the length of the bit-string, is 1012 and is explained in the description of the block *decode trial vectors* below. `NO_TRIALS` is not set — the loop is modified to carry until the witness is satisfied with one of the faces in the population. He/she is then to press the stop button on the GUI. The `LEARNING_RATE` was set to 0.3. The arrival at this figure is described in Section 6.4. This is a relatively high setting, but is used because the number of generations must be kept to a minimum or the witness will loose concentration. `MUTATION_PROBABILITY` is set to 0.02 and `MUTATION_SHIFT` = `LEARNING_RATE`.

Decode trial vectors

Each trial Vector is encoded as a $46 \times 22 = 1012$ element bit-string. See Section 4.3 which describes the encoding and decoding of the bit-string. What is performed in this block is exactly as described in section 4.3. Each bit-string holds the information to generate one of the faces in the GUI, so 46 coefficients must be encoded and the precision of the encoding is 22 bits. (This is a precision is not necessary, but the system performance is fast enough.) The output of this block is *noFaces* sets of 46 real-numbered eigenface coefficients.

Gaussian

Each coefficient is transformed as described in Section 6.2 using (6.9). For each coefficient, σ is set to the standard deviation of the projected coefficients of the training set onto its eigenface axis. This is the eigenvalue associated with the eigenface.

Generate faces

This block multiplies each set of eigenface coefficients by their associated eigenfaces and then sums the products and the average face as shown in (3.32). This is repeated for each set of coefficients to generate *noFaces* face images.

GUI

The GUI displays the *noFaces* face images as shown in Figure 6.5. In this case *noFaces* = 20. The witness is then to click on the face perceived to be the most similar to the perpetrator's. The face number of this face (*selectedFaceNo*) is output from this block.

Associate trialVector

This block feeds back into PBIL the trial vector that provided the coefficients for generating the face image that the user selected. This trial vector is `trialVector`[*selectedFaceNo*].

6.4 Computer simulations

In the following experiments the computer chooses the winner of each population. The goal of the experiments is to try and find a face image in the test set.

This is done by letting the face recall system run as explained previously. The only difference is that instead of a human user choosing the closest match face image, the computer does. This is done by calculating the DIFS from the face image to be found to each face image in the population being displayed. The winner is the face image that has the smallest DIFS.

It must be noted that the face in the GUI that is the closest in terms of DIFS would not necessarily be the face that a human user would select as being the most similar. In fact in the later trials discussed in Chapter 7 it is seen that this is almost never the case.

The reasons that these experiments were performed with the computer choosing the best match is that its choices are consistent and could in a sense be called the optimal choices.

6.4.1 Assessing the convergence of the system to a target face

Generally optimisations are performed over thousands of generations. With a human choosing the best match this is not viable. The first experiment performed was to see how many iterations it would take PBIL to converge on an acceptable set of coefficients if the optimal best match was automatically chosen each time.

The face that was searched for was a face in the test set, Figure 5.9(b). It can be seen from Figure 6.4 that if the optimal best match is chosen as the winner of each population, then after 30 generations (the second last face in Figure 6.4) the best match face looks like the face that is being searched for. In fact it is almost identical to the closest approximation that can be made with the eigenfaces, which is shown in Figure 5.13. This is an extremely promising and surprising result as it usually takes thousands of generations for optimisation algorithms to find points close to the optimum. Unfortunately the optimal best matches that were automatically chosen are not the obvious choices for best matches to a human user.

Table 6.1 shows the DIFS between the best match face images in Figure 6.4 and the face image being searched for. It can be seen that the DIFS generally decreases over the generations, but

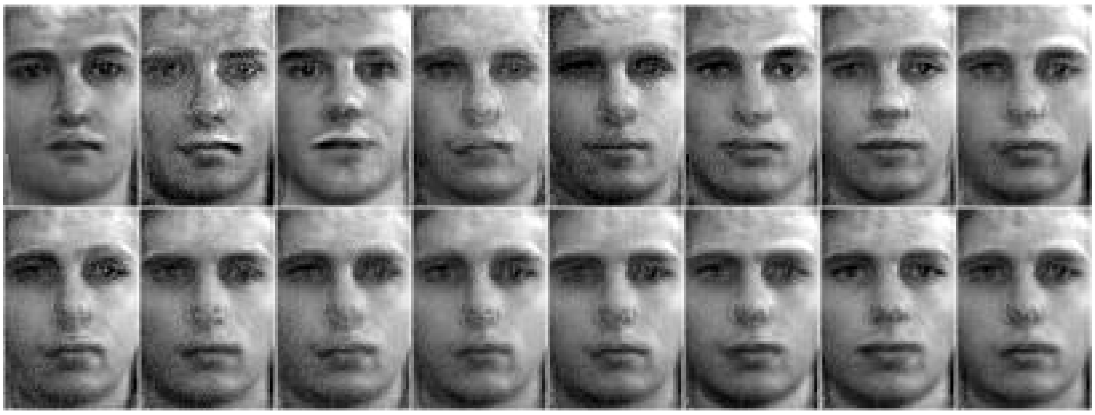


Figure 6.4: The optimal best match faces chosen in the automated simulation with a population size of 100 and a learning rate of 0.2. The face to be reproduced is Figure 5.9(b). The best match face for every second generation is displayed from left to right, top to bottom.

1578	1450	1435	1179	1121	1065	931	964
920	827	765	719	728	675	696	606

Table 6.1: The DIFS between the faces in Figure 6.4 and the face being searched for. The position of the numbers in the table match the positions of the images in Figure 6.4.

is not monotonic. It increases from 931 to 964 from the 14th to the 16th generation. This is a characteristic of PBIL, which exists because it does not keep a copy of the best match of the previous generation in the current generation. This helps to stop PBIL from converging prematurely on a local minimum.

6.4.2 Finding settings for the PBIL parameters

Values had to be chosen for the parameters `LEARNING_RATE` and `NO_TRIALS` (the population size) for the PBIL algorithm in the developed system before trials with humans could be run. The learning rate should be set high enough so that the system does not take long to converge on a good likeness, but it must not be so large that it results in premature convergence on a local optimum. The population size should be large enough that the update to the probability vector is based on a good sampling of the search space, but there should not be too many faces displayed on the screen. This is because it is difficult for human users to compare a large number of faces every generation. This was noted during the preliminary trials with humans using the system. It was further substantiated during the experiments carried out described in Chapter 7.

To choose settings for these parameters, simulations were run with the computer choosing the best match based on the smallest DIFS. These simulations were automated because it was desired to sweep through a large range of possible settings.

It would not be possible for a human to perform all these trials. Also there would be a large difference in the resultant face generated by a human, for the same parameter settings, over different trial runs. This difference would be even larger if different people performed the trials. To average out these differences many runs would have to be made for each combination of parameters. This is impractical.

Three runs were made for each combination of the parameters. The best match DIFS after 30 generations was averaged across the three runs and then noted. These averaged DIFS are shown in Table 6.2. The learning rate was swept from 0.05 to 0.4 and the population size from 10 to 50.

		Learning rate							
		0.05	**0.1**	**0.15**	**0.2**	**0.25**	**0.3**	**0.35**	**0.4**
	10	1631	1360	1069	985	965	914	958	902
	15	1407	1303	1048	963	808	870	940	905
	20	1478	1173	886	864	977	876	841	915
Population size	**25**	1381	1203	928	813	828	765	788	806
	30	1361	1171	937	776	717	**667**	669	791
	35	1346	1097	887	850	702	738	762	865
	40	1441	1079	856	796	713	616	773	766
	45	1359	993	840	714	717	752	715	699
	50	1349	1063	897	768	625	654	627	734

Table 6.2: DIFS measures after 30 PBIL generations with the learning rate and population size being changed. Each block is the average taken over three runs. In the simulations the best match was chosen according to the smallest DIFS.

Looking at the results shown in Table 6.2 it was decided to have a population size of 30 with a learning rate of 0.3. This is the fourth smallest DIFS in the table. Some combinations with populations sizes greater than 40 resulted in a lower final DIFS, but it was decided to present a smaller number of faces to the users at a time. These are the parameters settings used for the experiments in Chapter 7, where the results of humans using the system are presented.

6.5 Using the developed face recall system

Figure 6.5 shows the graphical user interface (GUI) of the developed system. The face recall process is started with the GUI displaying a random sample of faces. Using a mouse the witness will then click on the face that most resembles the perpetrator's face. A new set of faces will then appear on the screen which look more similar to the clicked-on face, but the

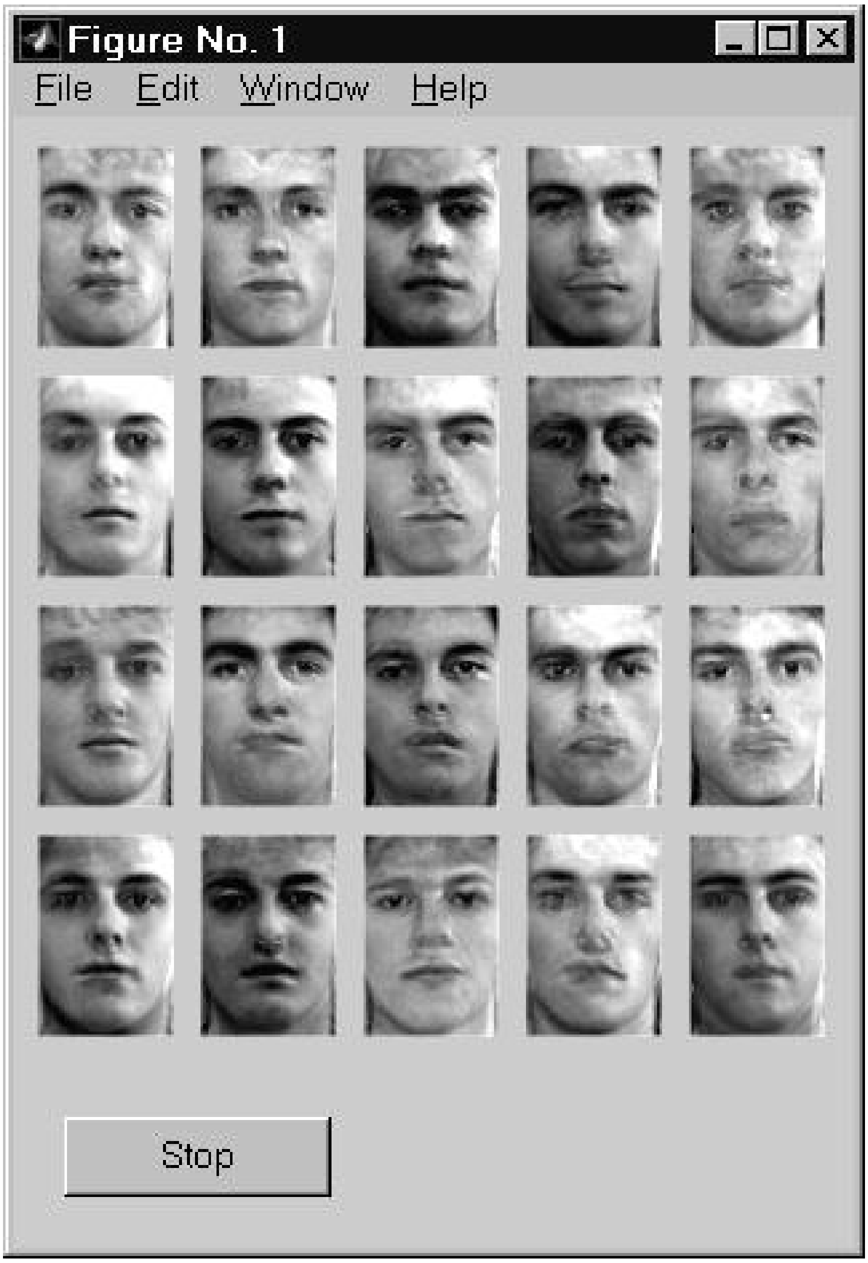


Figure 6.5: User interface of the developed face recall system after the user has made 1 choice.

spread of different looking faces will still be wide. Once again the witness is to click on the most similar looking face and a new set of faces will be displayed. With each iteration of the above process, the displayed set of faces will look more similar. This enables the witness to narrow in on a likeness of the perpetrator. This can be seen in Figure 6.5 which shows the second set of faces displayed, Figure 6.6 which is after 8 iterations and Figure 6.7 which is after 14 iterations.

Once the witness is satisfied that one of the displayed faces is a good enough likeness to the perpetrator, the process is stopped and that face can be printed onto paper or saved as a computer image to be used later.

The system described above is still a prototype system, it is only able to construct the face without a hair-style and other accessories such as beards, glasses etc. This could done using a similar method to that of a facial composition system, an example of which is shown in Figure 2.4. Different sets of eigenfaces should be used for different sex, race, age groups. This would restrict the search space and make it easier for the witness to find a good likeness face image.

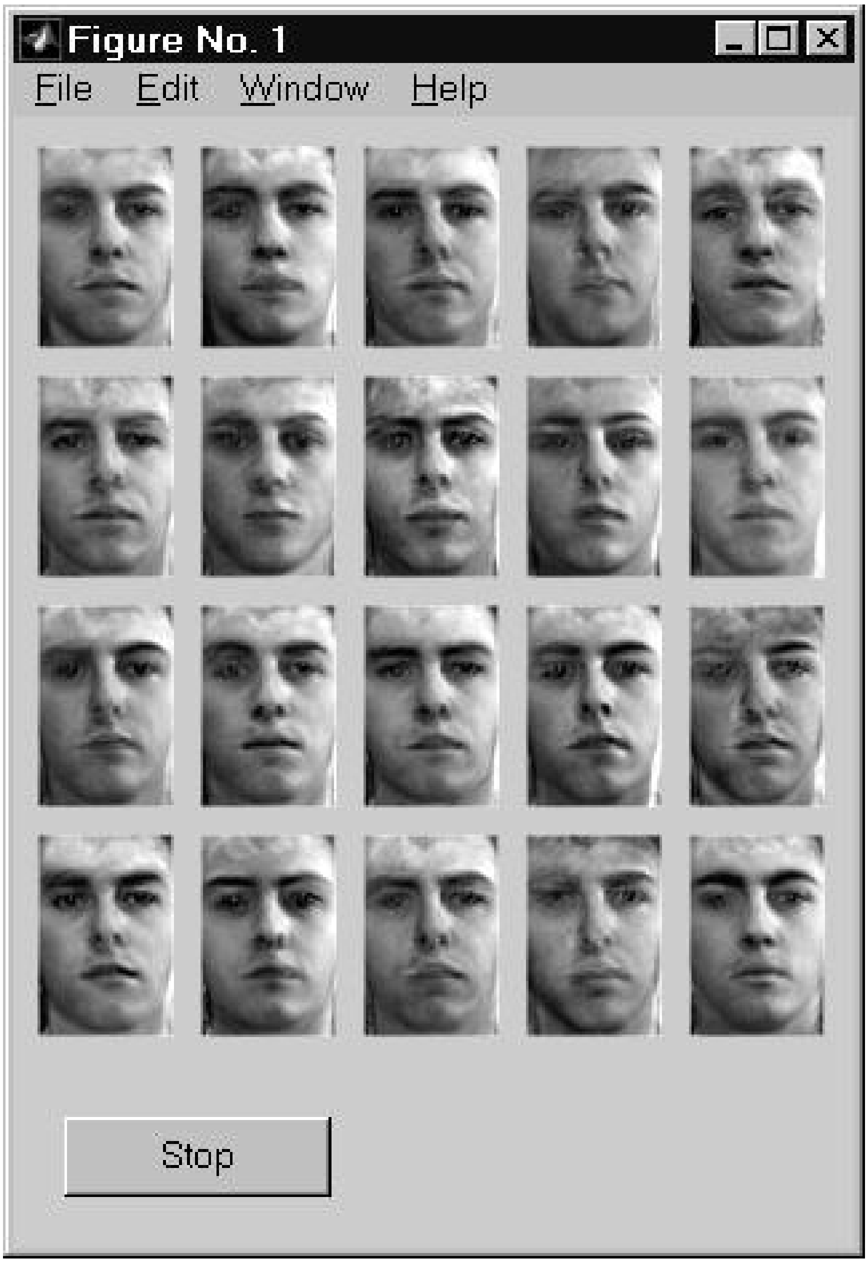


Figure 6.6: User interface of the developed face recall system after the user has made 8 choices.

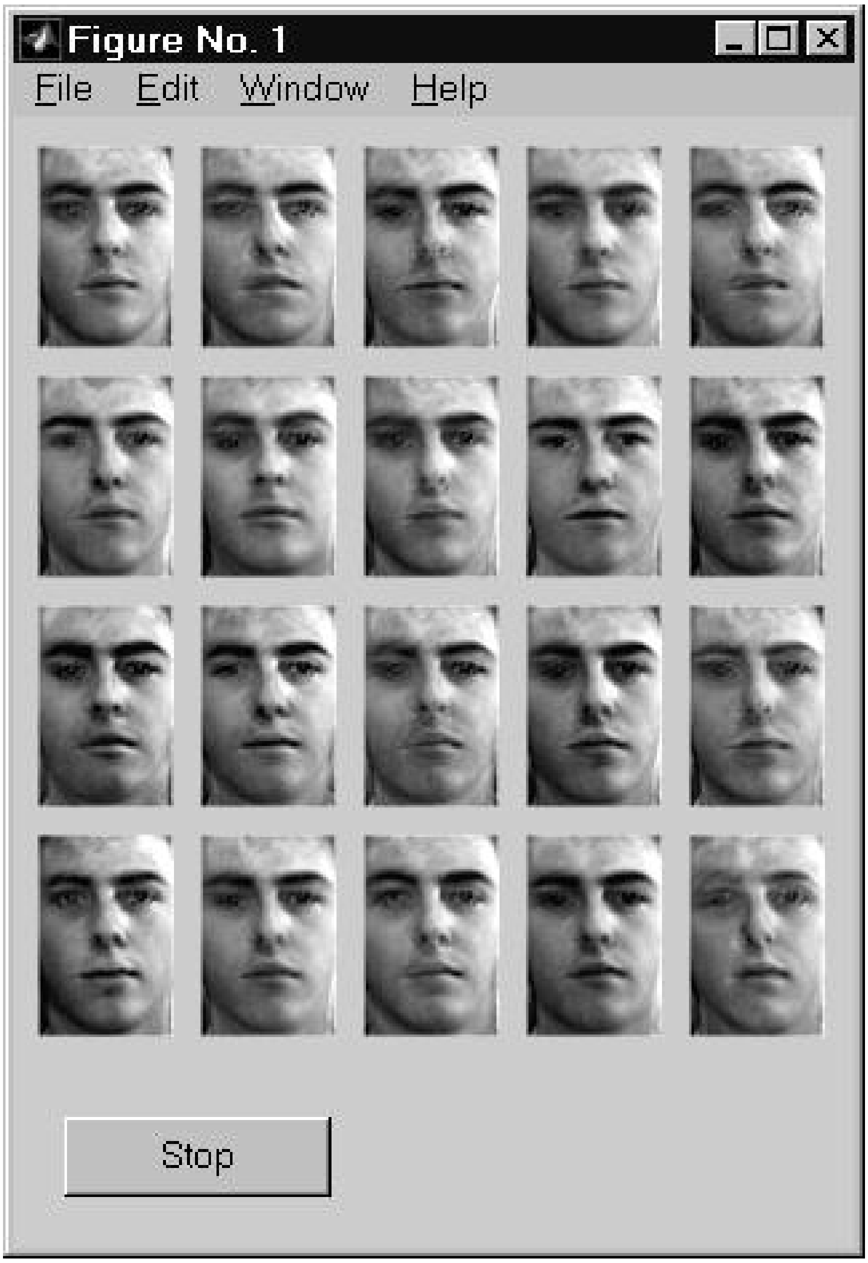


Figure 6.7: User interface of the developed face recall system after the user has made 14 choices.

Chapter 7

Experiments Assessing the System

This chapter presents the results of experiments run with the system. These experiments are with people interacting with the face recall system — unlike those in Chapter 6 where the computer simulated the optimal choices. Two different experiments are presented in this chapter. In the first the face to be reconstructed is viewable during the whole construction process. In the second the face to be reconstructed is only shown for a brief period before the reconstruction. This is the more realistic condition under which a face recall system would be used.

7.1 Experiment 1

Experiment 1 was constructed to test how the system would work under optimal conditions. The optimal condition would be a person with a perfect memory of a face trying to construct a visual likeness to it. This optimal condition is simulated by having in view of the user, while interacting with the system, a picture of the face to be found. The face to be reconstructed will be referred to as the *target* face. This is a test of a *human's* ability under optimal conditions to generate a good likeness to the target face while interacting with the developed face recall system. (It is already known that the system can generate good likenesses if the face with the smallest DIFS from the target is automatically selected each generation.)

This experiment is broken into two stages: the first is the *construction phase* where likenesses to the target faces are generated (Section 7.1.1), the second is the *assessment phase* where judgements are made on the similarities of the generated likeness images to the target faces (Section 7.1.2).

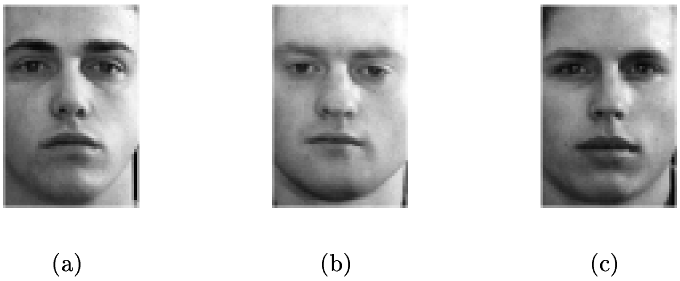
7.1.1 Experiment 1 construction phase

What follows is a description of the construction of likenesses to the target faces with the developed face recall system (in *procedure*). Then the generated likeness face images are presented with an analysis of their generation (in *results*).

Procedure

Fifteen undergraduate psychology students from the University of Cape Town volunteered to perform the constructions. Their makeup was 11 females and 4 males between the ages of 18 and 30. They will be referred to as the *participants* of the experiment.

The construction phase took place over three days, with the participants coming one at a time to perform their face image constructions. Their constructions were performed on the developed face recall system, described in Chapter 6. The system used the set of eigenfaces presented in Section 5.1.3. The participants were each given an explanation of what was expected from them and how to use the system. Then they were given a few minutes to familiarise themselves with the system.



(a) (b) (c)

Figure 7.1: The face images to be found in Experiment 1.

The experiment started with the displaying of one of the faces in Figure 7.1 on the computer screen next to the system GUI. The participant then tried to construct a likeness face image to it using the face recall system. She was told to stop when she was satisfied that she had found a good likeness to the target face. Her choices for best match in each generation was then displayed on the computer screen and the participant had to select the face image which she believed to be the most similar to the target face. This image will be referred to as the participant's *best choice*. This procedure was repeated for the other two faces in Figure 7.1. The order in which the faces were presented for reconstruction was such that the permutations of the order were as different as possible.

Results

Appendix B contains all the results obtained from Experiment 1. What is presentation here is a portion of the results and some summarising statistics with discussion.

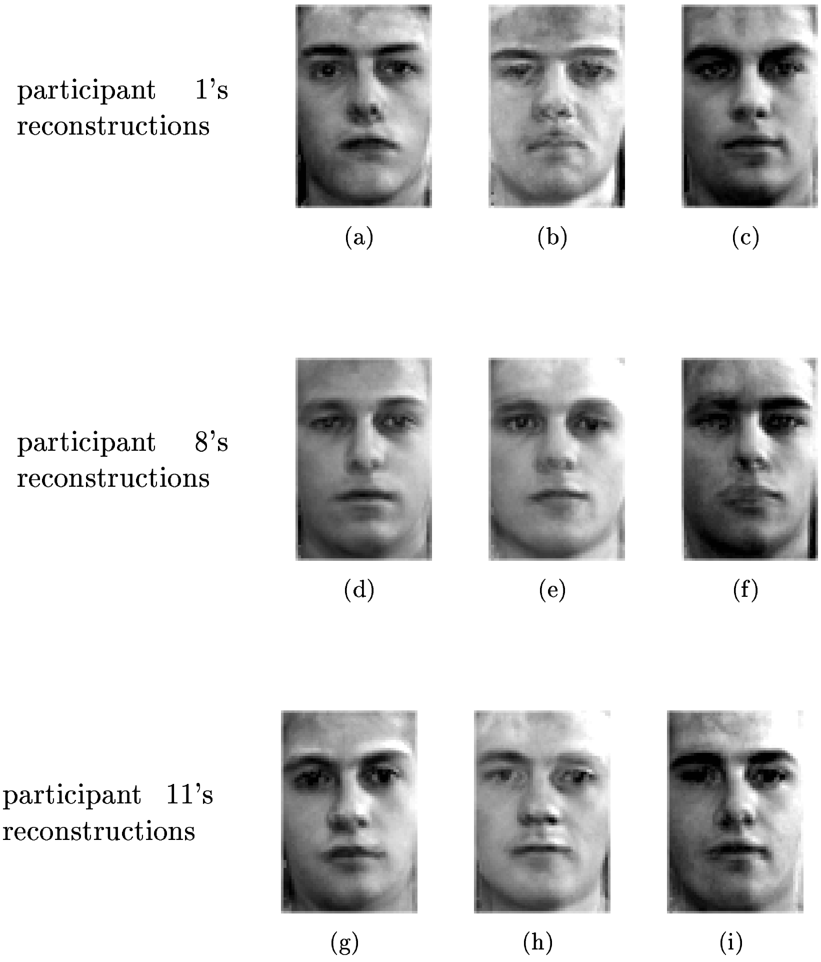

participant 1's reconstructions
(a)
(b)
(c)
participant 8's reconstructions
(d)
(e)
(f)
participant 11's reconstructions
(g)
(h)
(i)

Figure 7.2: Likenesses to the faces in Figure 7.1 found by participants 1, 8 and 11. The order of presentation of the face images in each line is: likeness to Figure 7.1(a), likeness to Figure 7.1(b) and likeness to Figure 7.1(c). All the likenesses generated in Experiment 1 are presented in Appendix B.

Figure 7.2 shows the best choices of some of the participants. All the best choice likenesses generated in Experiment 1 are shown in Figures B.2, B.3 and B.4 in Appendix B.

Figure 7.3 shows the best match faces chosen by participant 1 in her search for a facial likeness to the target, Figure 7.1(a). It can be noted that there is little difference in the last faces.

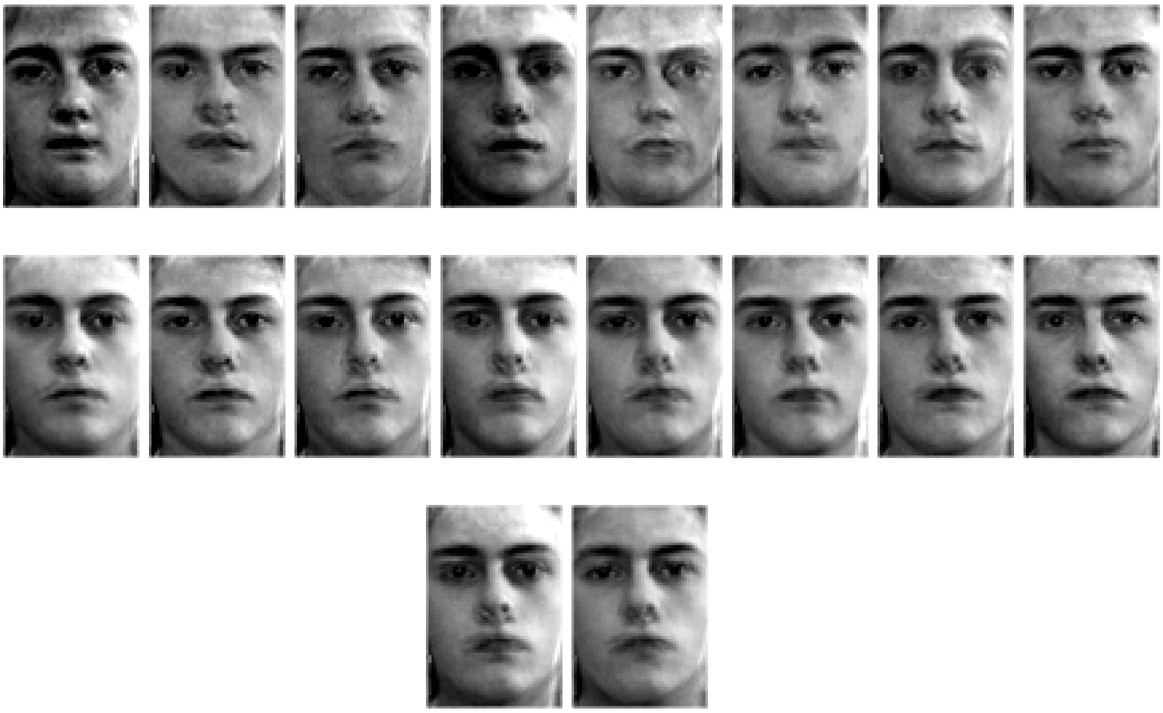


Figure 7.3: The best match faces chosen by participant 1 in her search for the face in Figure 7.1(a). The first face is her choice of best match for the first generation, the second face is her choice for the second generation and so on. The sequence goes from left to right, top to bottom. She made 18 choices in all and chose her 16th choice to be her best choice.

This is due to the convergence of the displayed population of faces in the latter generations.

generation	**DIFS of user's choice**	**smallest DIFS in population**	**faces in population closer than user's choice**
1	3036	1590	20
2	2177	1560	12
3	2320	1614	16
4	3011	1780	26
5	2700	1990	12
6	2407	1673	7
7	1983	1737	3
8	1844	1661	2
9	1780	1612	2
10	2046	1633	16
11	1954	1642	13
12	1701	1685	1
13	1930	1636	11
14	2185	1620	24
15	1816	1638	5
16	1884	1681	12
17	1772	1719	3
18	1642	1642	0

Table 7.1: A table showing the DIFS of participant 1's best match choices, shown in Figure 7.3, to the target Figure 7.1(a). The smallest DIFS to a face image in the displayed population and the number of faces in the population who have a smaller DIFS than participant 1's choice. This is shown for each generation.

Table 7.1 shows the DIFS from participant 1's best match choices to the target face. It also shows what the smallest DIFS from the target to a face in the displayed population was and the number of faces in the population a shorter DIFS away than the participant's choice. It can be seen that participant 1's choice of the most similar face is almost never the face that is the smallest DIFS away from the target face. The only time that they were the same was in the 18th generation. Generally a substantial number of faces are closer in terms of DIFS. The values shown in Table 7.1 are characteristic of the other reconstructions.

Table 7.2 gives an analysis of the times taken to find likenesses to each of the target faces. The average times turned out to be lower than expected.

Table 7.3 gives an analysis of the number of generations that were run through to find likenesses to each of the target faces. The average number of generations also turned out to be lower than expected.

Table 7.4 gives an analysis on the differences between the number of choices made and the

	mean (minutes)	**minimum (minutes)**	**maximum (minutes)**	**standard deviation (minutes)**
Figure 7.1(a)	10.5	5	25	5.0
Figure 7.1(b)	11.7	5	28	6.3
Figure 7.1(c)	10.3	5	24	5.8

Table 7.2: The mean, minimum, maximum and standard deviation of the time taken to find the facial likeness images in Experiment 1. The time for each reconstruction is given in Table B.1 in Appendix B.

	mean	**minimum**	**maximum**	**standard deviation**
Figure 7.1(a)	15.5	6	29	6.8
Figure 7.1(b)	15.0	6	28	6.0
Figure 7.1(c)	14.5	8	30	6.2

Table 7.3: The mean, minimum, maximum and standard deviation of the number of choices made to find facial likeness images in Experiment 1. The number of choices for each reconstruction is given in Table B.1 in Appendix B.

	mean	**minimum**	**maximum**	**standard deviation**
Figure 7.1(a)	2.4	0	12	3.4
Figure 7.1(b)	3.2	0	12	3.4
Figure 7.1(c)	2.9	0	11	3.4

Table 7.4: The mean, minimum, maximum and standard deviation of the difference between the number of choices made and the choice number of the best choice for each face search in Experiment 1.

choice number of the best choice. This is presented to assess the need for making a final best choice out of all the choices as opposed to taking the last choice as the resultant likeness face image. The relatively large mean, maximum and standard deviations of this difference shown in Table 7.4 seem to suggest that this a necessary step in the face recall process. Although this may seem to undermine the effectiveness of the PBIL algorithm, it is not so since a characteristic of PBIL is that a population will not necessarily have fitter members than the previous population. This is to stop premature convergence (see Section 4.5).

7.1.2 Experiment 1 assessment phase

This phase of Experiment 1 is to obtain a measure of the quality of the reconstructions. This must in some way measure the resemblance of the generated face image's likeness to its target face. This was done by presenting the generated face image above a lineup of faces which included the target face. *Judges* then specified the face in the lineup that they thought was the most similar to the generated face.

Procedure describes the judging procedure and *results* presents the results of the judging.

Procedure

For each of the target faces 6 other standardised face images were chosen to be in the lineup with it. For each participant a page was laid out with their three constructions each above the associated lineup containing the construction's target. The order of presentation of the constructions as well as the position of the targets within the lineup were changed in each participant's lineup page. This was done such that the permutations were as different as possible. Then 20 copies of each of these pages were made, resulting in 300 pages each containing three lineups. Two of the 15 different lineup pages are included at the end of Appendix B.

These 300 lineup pages were handed out to 300 first year students at the University of Cape Town at the start of their psychology lecture. The 300 judges were then told to specify which face in each lineup looked most like the generated face image above the lineup.

Results

267 of the 300 lineup pages were returned unspoilt. The judgements they contained are tabulated in Table 7.5. It shows the judgements for each of the generated facial likeness images.

Table 7.5 is summarised in Table 7.6. Table 7.6 shows that the constructions of Figure 7.1(a)

	Figure 7.1(a)		**Figure 7.1(b)**		**Figure 7.1(c)**	
	correct	**incorrect**	**correct**	**incorrect**	**correct**	**incorrect**
participant 1	**14**	2	8	8	4	12
participant 2	8	**11**	**12**	7	7	**12**
participant 3	**10**	5	**9**	6	0	**15**
participant 4	**12**	5	**16**	1	**11**	6
participant 5	**16**	3	4	**15**	3	**16**
participant 6	**10**	8	**12**	6	0	**18**
participant 7	**13**	2	**9**	6	7	7
participant 8	**14**	4	**15**	3	6	**12**
participant 9	**11**	7	8	**10**	1	**17**
participant 10	**11**	9	**11**	9	**11**	9
participant 11	**12**	6	8	**10**	**15**	3
participant 12	**14**	3	**11**	6	5	**12**
participant 13	**14**	4	**13**	5	2	**16**
participant 14	**13**	6	**11**	8	5	**14**
participant 15	**12**	8	5	**15**	1	**19**

Table 7.5: Tabulation of the results obtained from the evaluation of the Experiment 1 lineups. The table shows for each construction performed in Experiment 1 the number of judges that selected the correct face in the lineup and the number of judges that selected an incorrect face. For each construction the larger of the two is in bold.

	lineups judged	**targets identified**	**targets identified (%)**
Figure 7.1(a)	267	184	69
Figure 7.1(b)	267	79	30
Figure 7.1(c)	267	152	57

Table 7.6: The number and percentage of correctly selected target faces out of the lineups in Experiment 1.

enabled the judges to identify it, out of the used lineup, 69% of the time. The constructions of Figure 7.1(b) enabled a 30% correct identification and the constructions of Figure 7.1(c) enabled a 57% correct identification. These percentages should be compared to the one in seven chance resulting in a 14% probability of correctly guessing which is the target face if the likeness face image provides no information as to which is the target face in the lineup.

7.2 Experiment 2

Experiment 2 was constructed to test the system under more realistic conditions. This is to use the system to construct a face from memory.

Again the experiment is broken into two stages: the *construction phase* where likenesses to the target face are generated (Section 7.2.1) and the *assessment phase* where judgements are made on the similarities of the generated likeness images to the target faces (Section 7.2.2).

7.2.1 Experiment 2 construction phase

What follows is a description of the constructing of likenesses to the target faces with the developed face recall system (in *procedure*). Then the generated likeness face images are presented with an analysis of their generation (in *results*).

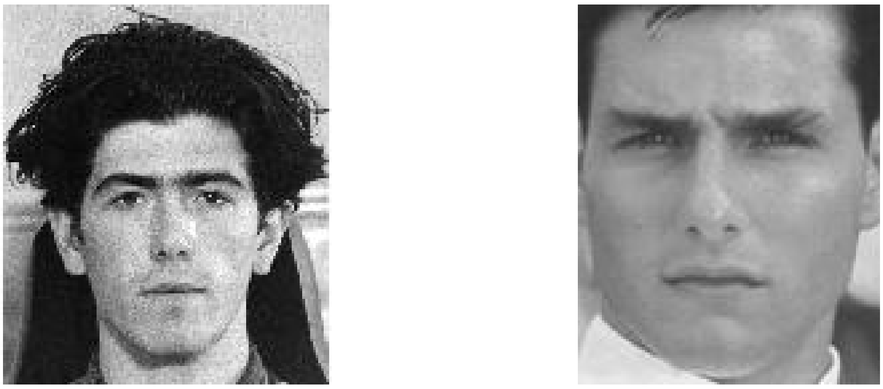
Procedure

Fifteen undergraduate psychology students from the University of Cape Town volunteered to perform the constructions. Their makeup was 13 females and 2 males between the ages of 18 and 22. They will be referred to as the *participants* of the experiment. None of the participants of Experiment 2 were also participants of Experiment 1.

The constructions performed during this experiment were made with the developed face recall system, which is described in Chapter 6. The system used a new set of eigenfaces generated for this experiment. The principal component analysis was performed on a combination of the training set and the test set with the exclusion of the face image shown in Figure A.1.36. This is an image of one of the faces to be constructed in this experiment.

The construction phase took place over 3 days, with the participants coming one at a time to perform their face image constructions.

The procedure was as follows: a short explanation of the purpose of the experiment and what was expected was given. One of the two face images in Figure 7.4 was shown to the participant for 15 seconds. A brief explanation of how to use the system was given. A lineup including the face presented previously was shown to the participant and she had to identify the face that was previously shown. This was to check that the participant remembered what the target face looked like. A five minute filler task was then given to the participant to perform, so that there would be a time lapse and intermediate task performed between her exposure to the target face and her construction of it. The participant was then told to start constructing the target face with the face recall system and to stop when she was satisfied that she had found a good likeness to the it. Her choices for best match in each generation was then displayed on the computer screen and the participant had to select the face image which she believed to be the most similar to the target. This image will be referred to as the participant's *best choice*. The participant was then shown another lineup containing the target face and was asked to identify it. This was done to see whether the exposure to the faces displayed by the face recall system had corrupted her memory of the target face. This procedure was repeated for the other face in Figure 7.4. The order in which the two faces for reconstruction were presented was alternated for each participant.



(a) Strange face.

(b) Familiar face.

Figure 7.4: The face images to be found in Experiment 2. The strange face is a face image that came from the same set of faces that where used to generate the eigenfaces. The familiar face is Tom Cruise, a well-known actor.

Results

Appendix C contains all the results obtained from Experiment 2. What is presented here is a portion of the results and some summarising statistics with discussion.

Figure 7.5 shows the best choices of some of the participants for the reconstruction of the unfamiliar face. All the best choice likenesses to the unfamiliar face generated in Experiment 2 are shown in Figure C.2 in Appendix C.

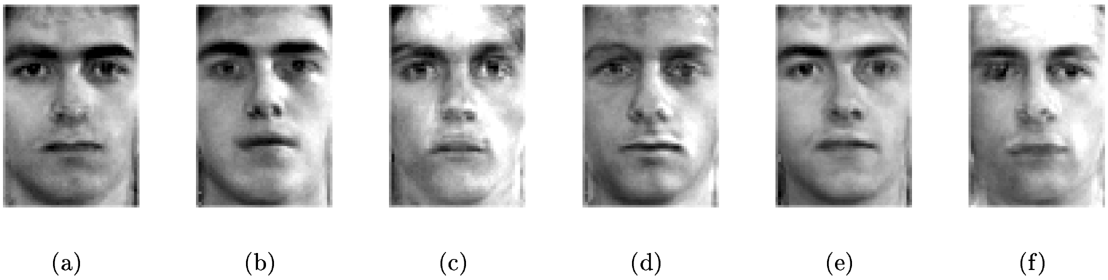


Figure 7.5: Face likeness images to the unfamiliar person found in Experiment 2. A facial image of the unfamiliar person is shown in Figure 7.4(a). The face images shown are the best choices of participants 2, 4, 7, 8, 12 and 15, displayed in the same order.

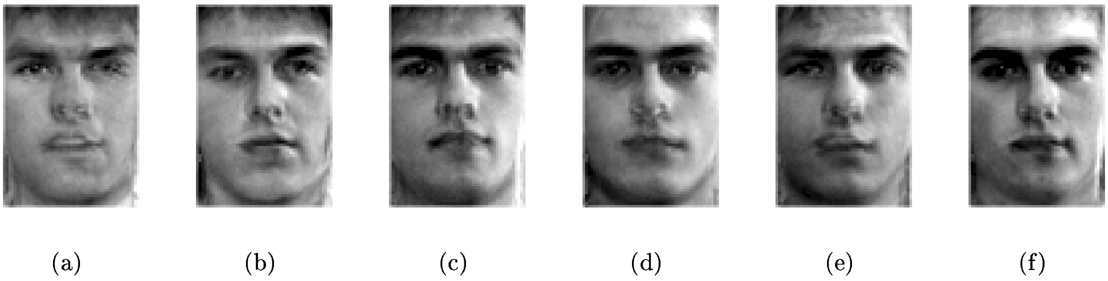


Figure 7.6: Facial likeness images to the familiar person (Tom Cruise) found in Experiment 2. A facial image of Tom Cruise is shown in Figure 7.4(b). The face images shown are the best choices of participants 5, 6, 8, 11, 12 and 13, displayed in the same order.

Figure 7.6 shows the best choices of some of the participants for the reconstruction of the familiar face. All the best choice likenesses to the familiar face generated in Experiment 2 are shown in Figure C.3 in Appendix C.

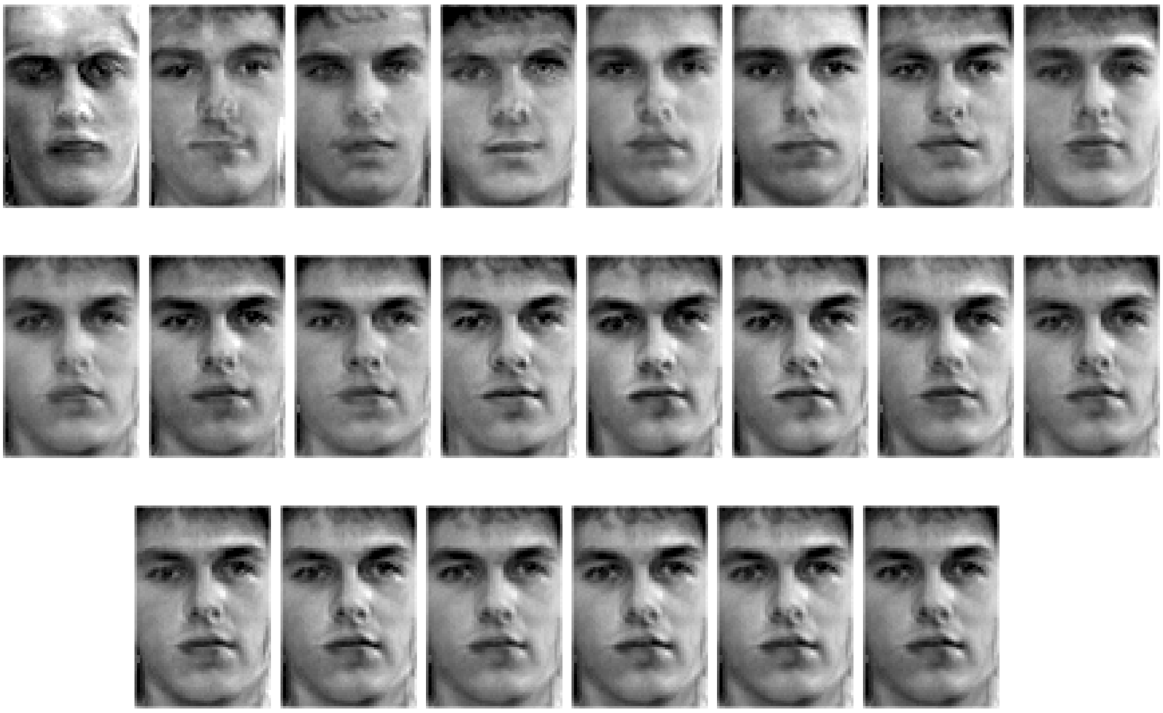


Figure 7.7: The best match faces chosen by participant 6 in her search for a facial image likeness to Tom Cruise. The first face is her choice of best match for the first generation, the second face is her choice for the second generation and so on. The sequence goes from left to right, top to bottom. She made 22 choices in all and chose her 17th choice to be her best choice.

Figure 7.7 shows the best match faces chosen by participant 6 in her search for a facial likeness to Tom Cruise. Again it can be noted that there is little difference in the last faces.

Table 7.7 gives an analysis of the times taken to find likenesses to each of the target faces. As in Experiment 1 the times were low.

Table 7.8 gives an analysis of the number of generations that were run through to find likenesses to each of the target faces. The mean number of choices made per construction is approximately 5 choices more than in Experiment 1.

Table 7.4 gives an analysis on the differences between the number of choices made and the choice number of the best choice. The mean difference is also larger than in Experiment 1.

	mean (minutes)	**minimum (minutes)**	**maximum (minutes)**	**standard deviation (minutes)**
strange face	12.3	6	28	5.4
familiar face	11.9	4	28	7.0

Table 7.7: The mean, minimum, maximum and standard deviation of the time taken to find the facial likeness images in Experiment 2. The time for each reconstruction is given in Table C.1 in Appendix C.

	mean	**minimum**	**maximum**	**standard deviation**
strange face	19.6	9	30	5.7
familiar face	21.5	10	30	6.1

Table 7.8: The mean, minimum, maximum and standard deviation of the number of choices made to find facial likeness images in Experiment 2. The number of choices for each reconstruction is given in Table C.1 in Appendix C.

	mean	**minimum**	**maximum**	**standard deviation**
strange face	5.7	0	15	5.3
familiar face	4.4	0	17	3.8

Table 7.9: The mean, minimum, maximum and standard deviation of the difference between the number of choices made and the choice number of the final choice for each face search in Experiment 2.

7.2.2 Experiment 2 assessment phase

Like in Experiment 1 the assessment was done by presenting the generated face image above a lineup of faces which included the target face. *Judges* then specified the face in the lineup which they thought was the most similar to the generated face.

Procedure describes the judging procedure and *results* presents the results of the judging.

Procedure

For each of the target faces 6 other face images were chosen to be in the lineup with it. For the unfamiliar face image the 6 face images were chosen from the original unstandardised face images obtained from the University of Sterling. For Tom Cruise's face 6 other face images of famous actors were selected. For each participant a page was laid out with their two constructions above the associated lineup containing the construction's target. The order of presentation of the constructions as well as the position of the targets within the lineup was changed in each participant's lineup page. This was done such that the permutations were as different as possible. Then 20 copies of each of these pages were made, resulting in 300 pages. Two of the 15 different lineup pages are included at the end of Appendix C.

These 300 lineups were again handed out to psychology students at the University of Cape Town. The 300 judges were told to specify which face in each lineup looked like the generated face image above the lineup.

Results

227 of the 300 lineup pages were returned unspoilt. The judgements they contained are tabulated in Table 7.5. It shows the judgements for each of the generated facial likeness images in Experiment 2.

Table 7.10 is summarised in Table 7.11. It shows that the unfamiliar face was correctly identified 17% of the time and the familiar face was identified 58% of the time.

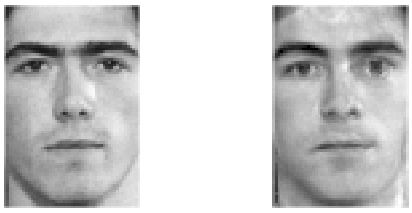
The 17% correct identification of the constructions of the unfamiliar face is not much better than the pure chance probability of 14%. The reason for the poor reconstruction was initially thought to be the poor reconstruction capability of the eigenfaces for the unfamiliar face. This was checked by projecting the unfamiliar face onto the eigenface axes and then using the obtained eigenface coefficients to reconstruct the face. The results of this reconstruction are shown in Figure 7.8 and this however shows that a good reconstruction of the unfamiliar face can be obtained with the eigenfaces.

	strange face		**familiar face**	
	correct	**incorrect**	**correct**	**incorrect**
participant 1	5	**8**	**9**	4
participant 2	7	7	**13**	1
participant 3	0	**16**	8	8
participant 4	3	**10**	**7**	6
participant 5	0	**16**	**10**	6
participant 6	6	**10**	**9**	7
participant 7	3	**14**	**12**	5
participant 8	1	**16**	**14**	3
participant 9	0	**14**	3	**11**
participant 10	2	**14**	5	**11**
participant 11	0	**15**	5	**10**
participant 12	4	**9**	**11**	2
participant 13	5	**11**	**12**	4
participant 14	1	**14**	**9**	6
participant 15	2	**14**	5	**11**

Table 7.10: Tabulation of the results obtained from the evaluation of the Experiment 2 lineups. The table shows for each reconstruction performed in Experiment 2 the number of judges that selected the correct face in the lineup and the number of judges that selected an incorrect face. For each reconstruction the larger of the two is in bold.

	lineups judged	**targets identified**	**targets identified (%)**
strange face	227	39	17
familiar face	227	132	58

Table 7.11: The number and percentage of correctly selected target faces out of the lineups in Experiment 2.



(a) Standardised unfamiliar face.

(b) Reconstruction.

Figure 7.8: The optimal reconstruction of the unfamiliar face in Experiment 2 with the eigenfaces used.

7.3 Feedback from participants

While the participants were reconstructing the target faces in Experiment 1 and Experiment 2 the following feedback was obtained from them:

- Even though they were told to concentrate on the face as a whole when making their choice of best match, most of the participants said that they chose their best match according to particular features. 15 participants said they based their decision mainly on the eyes and eyebrows, 6 said the mouth, 5 said the nose, 4 said the jaw and chin and 2 said the overall face shape. Many said they based their best match on the most prominent feature of the face to be found.
- Approximately half the participants said their method of looking through the faces in the GUI was to look at all the faces at once and wait for a face to 'jump out' at them. The rest said they scanned the faces row by row and assessed each face individually.
- Most participants found it hard to distinguish between face images displayed in the GUI after the population started converging. The convergence results in the displayed images looking very similar, this is shown in Figure 6.7.
- Most participants commented on the variation of the displayed face images. The variations noted were the lighting and the orientation of the head and the position of the pupils of the eyes.
- Most participants commented on the fuzziness of some of the displayed faces.
- Many participants said they would like to go back to a previous generation of displayed faces.
- Most participants expressed the wish to change a particular facial feature once the population had converged.
- Some participants said that it would be easier for them if fewer faces were displayed at time.

7.4 Problem of cut out hair

A major shortcoming of the developed face recall system in its present implementation is that the face images it generates do not include the full face image. The sides of the face are cut away and the face images do not include hair.

This results in very similar looking faces. The effects of cropping away the hair and parts of the face on the resultant look of the face can be seen by comparing face images in Figure 5.1, which are full face images, to those Figure 5.8, which are the same face images cropped. It can be seen that there is much more information content in the full face image.

This issue is further emphasised in Figure 7.9, which shows the same face with different hairstyles. That the faces look like those of different people shows the importance of the hair in face images.

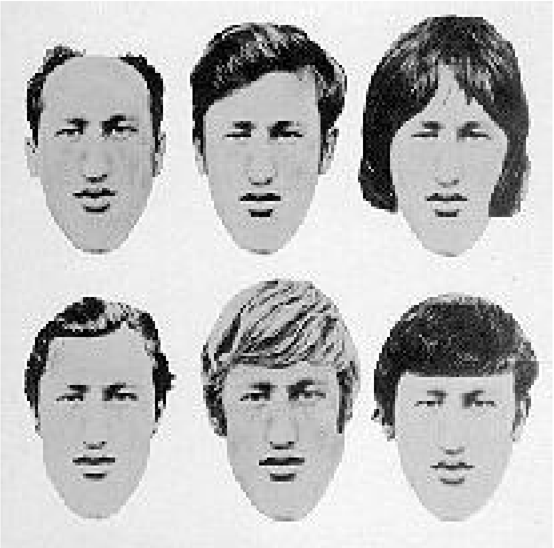


Figure 7.9: The effect of changing the hair on the same face image (*Ellis [8]*).

Because of this major shortcoming in the current system, not too much can be read into these experiments.

7.5 Discussion of experiment results

The identification of the target faces in the face lineups were far from as high as would be wanted for an operational face recall system used by police forces. This is true for both the unlikely optimal condition in Experiment 1 and the more realistic condition of Experiment 2.

The reconstruction of the unfamiliar face from memory in Experiment 2 was the more realistic conditions under which a face recall system would be used. The judged quality of its reconstructions were especially low. The 17% correct identification is just above chance. It would seem like the generated face likeness images did not provide any information at all.

The reconstructions of the familiar face from memory were fair with a 58% correct identification in the lineups. Some of the likeness images generated for it were extremely good. This can be seen in judgements of each likeness image in Table 7.10.

What can also be noted is that some people seem to be able to generate better quality likeness face images than others. This can be seen in Tables 7.5 and 7.10.

Some of the target faces seemed to be harder to construct with the system than others. This can be seen in the Tables 7.6 and 7.11, which show that there existed a big difference in the quality of the reconstructions for the different targets faces.

The short times and small number of generations taken by the participants to arrive at their best choice is a good feature. Because of the short times the face generation process could be run repeatedly to try and come up with a good likeness to the target.

Chapter 8

Comparison with Existing Systems

This chapter compares the developed face recall system with the existing face recall systems presented in Chapter 2. The developed face recall system is shown to address many of the shortcomings of the existing systems (these were laid out in Section 2.2).

8.1 Comparison to normal composite based systems

The three main problems in face recall systems put forward in Section 2.2 are addressed by the developed system as follows:

1. The expressive capabilities of the system, if implemented properly, should be sufficient. This is because by using the correct coefficients in the linear combination of the eigenfaces a good likeness to the desired face can be generated (see Section 5.2.2).
2. There is no need to have a skilled intermediary due to the simplicity of the system.
3. It is recognition based, so there is no selection of individual features in isolation. It is a true holistic solution to the problem.

Thus this system addresses the three main problems which sketch artists and normal composite-based systems (not recognition-based, like Johnston's) did not properly address.

Johnston's and Brunelli's systems, even though they have not been used for law enforcement purposes, seem to address the three main problems better than their predecessors. Both systems have unique problems associated with their methods which might stop them from being viable face recall systems. The developed face recall system is based on a combination of the ideas used in both systems and thereby addresses the unique shortcomings of both.

8.2 Comparison to Johnston's system

The expressive capabilities of the developed system is better due to the fact that the system can potentially generate any likeness if implemented properly.

It should be easier for an optimisation algorithm like PBIL or a genetic algorithm to search through faces in the developed system than in Johnston's (see Section 5.3). This should result in fewer generations needed before an acceptable facial likeness is generated. By restricting the number of generations the witness is exposed to, he/she has to judge fewer faces. This makes it easier for him/her to maintain concentration. By exposing the witness to fewer images the chances of corrupting their memory should also be reduced. Memory corruption occurs when the witness's mental image of the perpetrator gets modified due to the exposure to other faces.

The user only has to click on the best likeness image in each generation instead of evaluating *every* face image that is displayed as has to be done in Johnston's system. This once again reduces the load on the witness and should help them maintain concentration. It also reduces the time needed to arrive at a facial likeness.

8.3 Comparison to Brunelli's system

Internally Brunelli's system also uses linear combinations of eigensurfaces to produce a face image. However, instead of using one set of eigenfaces to produce its face image, it uses eigeneyes, eigennoses and eigenmouths which are positioned and smoothed together to form a face image. A set of eigeneyes was shown in Section 3.3.1.

Because of the reconstruction capabilities of the eigensurfaces technique, both Brunelli's and the developed system should be able to produce sufficiently good facial likenesses *if* the correct coefficients are found for the systems. The deciding factor in the feasibility of both methods is whether these coefficients can be found for the system.

The differences between the developed system and Brunelli's are :

1. In Brunelli's system each facial feature is made independently of the others and then positioned in the face, as opposed to the developed system where the whole face image is made up of one set of eigenfaces.

   This has the following consequences for Brunelli's system:

   (a) There are more coefficients to set. There must now be sets of coefficients for *each* eigenfeature and also coefficients associated with the positioning of these features, instead of just one set of coefficients for the eigenfaces as in the developed system. This results in a much larger search space, which makes finding an optimum or close to optimum point more difficult.

(b) The system is not truly holistic as only one feature is modified at a time, even though this modification can be seen in context of the face built at that point.

(c) The intrinsic statistical information about the allowable matching and positioning of different-looking features in real faces is captured to a large degree in the eigenface technique. This information is lost using the eigenfeature technique. It is possible with Brunelli's system to have a face that has eyes and mouth that could both exist, but which would never be found on the same face. Thus in the developed system all the possible faces generated conform to the statistics of real faces with respect to the matching and positioning of different-looking features, while with Brunelli's system this is not the case. So while using Brunelli's system the witness will have to sort through faces that could not exist, this adds to the difficulty of the task.

2. In Brunelli's system there is no systematic searching strategy — it is up to the witness to try and devise some sort of searching method and then to execute it. The system is also restricted by the witness's memory as a result of the following: to reduce the number of faces to be exposed to, it is necessary to remember the faces presented by the system and their associated coefficients. This can be used to stop the witness from being exposed to faces similar to those they have already seen but do not look like the perpetrator. It can also be used to direct the witness to faces that are close to faces that they rated as being similar to the perpetrator's. The remembering of many different combinations of scroll bar settings and their associated faces is something which humans will not do well.

   In the developed system the searching strategy is implemented by the system. The search algorithm, PBIL, performs the task of choosing faces for the user to evaluate better than the user can. Because it is implemented on computer it can have a memory of how previously displayed faces and their associated coefficients were evaluated.

Chapter 9

Conclusion

The significant findings and results obtained in this thesis are summarised and presented in this chapter.

The literature survey performed in this thesis led to the conclusion that existing face recall systems do not solve the task of translating a face in a person's memory to a visual image as effectively as would be desired. This prompted the development of a new face recall system.

It was shown in this thesis that using a linear combination of eigenfaces it is possible to generate acceptable face likeness images to new face images, with different coefficients in the linear combination resulting in different looking faces. The novel idea of using an optimisation algorithm to search through these coefficients to try and generate a face likeness image to a face in memory was tried. PBIL was the evolutionary optimisation algorithm used and the evaluation of the cost function is performed by the human user trying to generate the visual image.

The developed face recall system has a simple user interface because all that is required from the user is to repeatedly specify which of a set of face images looks most like the face that he/she is trying to construct. This is the evaluation of the cost function. It is also shown that if the correct coefficients for the linear combination of eigenfaces are found the system is capable of generating good photographic quality likenesses.

A drawback of this technique is that eigensurfaces cannot be used to generate realistic hair in the face images, so all the face images produced by this system show the face up to the start of the hairline above the forehead. This makes the task of constructing a face likeness image harder for the user and the resultant generated face likeness image does not have hair — this detracts from its ability to be a good representation of the face in memory.

Experiments were performed with the developed face recall system. These experiments each had 15 participants who generated face likeness images. In the first experiment the participants generated three face likeness images with the target images to be reconstructed viewable throughout the generation process. In the second experiment the participants generated the face images from memory. Here they were briefly shown a picture of a familiar and an unfamiliar face and had to construct them from memory using the face recall system. Each of the face likeness images created in the two experiments where then judged by a different set of 13 or more judges. This was done by presenting the generated face likeness image above a lineup of 7 face images containing the target face that was reconstructed. The judges then had to try and identify the target face in the lineup.

The number of generations and time used by the participants to generate their face likeness images were low. The highest mean number of generations for a target image was 20. This is extremely low considering that evolutionary optimisation algorithms normally run for thousands of generations. The highest mean time for a target image was 22 minutes.

In the experiments where the face likeness images were generated from memory by the participants, the results were good enough for the judges to identify the target image in the lineup only 17% of the time. This is just above the 14% probability of the target face being selected if the face likeness image provided no information as to which was the target face in the lineup. The reconstructions of the familiar face were better with a 58% correct identification of the target in the lineup.

The results obtained from experiments performed using this system show that in its present state it is not a viable system, but if these are interpreted as being preliminary feasibility studies the results are promising.

Bibliography

[1] S. Baluja. Population-based incremental learning: a method for integrating genetic search based function optimisation and competitive learning. Technical report, School of Computer Science, Carnegie Mellon University, June 1994.

[2] S. Baluja. An empirical comparison of seven iterative and evolutionary function optimisation heuristics. Technical report, School of Computer Science, Carnegie Mellon University, September 1995.

[3] S. Baluja and R. Caruana. Removing the genetics from the standard genetic algorithm. In *Proceedings of the Twelfth International Conference on Machine Learning*, July 1995.

[4] R. Brunelli and O. Mich. Spotit! an interactive identikit system. *Graphical Models and Image Processing*, 58(5), September 1996.

[5] C. Caldwell and V. Johnston. Tracking a criminal suspect through "face-space" with a genetic algorithm. In *Proceedings of the 4th International Conference on Genetic Algorithms*, 1991.

[6] S. E. Comish. Recognition of facial stimuli following an intervening task involving the identi-kit. *Journal of Applied Psychology*, 72(3):488–491, 1987.

[7] G. Davies and D. Christie. Face recall: an examination of some factors limiting composite production accuracy. *Journal of Applied Psychology*, 67(1):103–109, 1982.

[8] H. Ellis. Practical aspects of face memory. In *Eyewitness testimony - psychological perspectives.* Cambridge University Press, 1984.

[9] H. D. Ellis, G. M. Davies, and J. W. Sheperd. A critical examination of the photofit system for recalling faces. *Ergonomics*, 21(4):297–307, 1978.

[10] J. Freund and G. Simon. *Modern Elementary Statistics.* Prentice Hall, eighth edition, 1992.

[11] M. L. Gillenson and B. Chandrasekaran. A heuristic strategy for developing human facial images on a CRT. *Pattern Recognition*, 7:187–196, 1975.

[12] P. Hancock, V. Bruce, and A. Burton. A comparison of two computer-based face identification systems with human perceptions of faces. *Vision Research*, 38:2277–2288, 1998.

[13] P. Hancock, A. Burton, and V. Bruce. Face processing: human perception and principal component analysis. *Memory and Cognition*, 24(1):26–40, 1996.

[14] I. Jolliffe. *Principal Component Analysis.* Springer-Verlag, 1986.

[15] V. S. Jonston. Method and apparatus for generating composites of human faces. United States patent number 5,375,195, December 1994.

[16] M. Kirby and L. Sirovich. Application of the Karhunen-Loeve procedure for the characterisation of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1), January 1990.

[17] M. B. Kovera, S. D. Penrod, C. Pappas, and D. L. Thill. Identification of computer-generated facial composites. *Journal of Applied Psychology*, 82(2):235–246, 1997.

[18] K. R. Laughery and R. H. Fowler. Sketch artist and identi-kit procedures for recalling faces. *Journal of Applied Psychology*, 65(3):307–316, 1980.

[19] A. Law and W. Kelton. *Simulated Modeling and Analysis.* McGraw-Hill, 1982.

[20] W. Maschner. Facette, face design system. Technical report, SOFTWEHR police software products, www.facette.com, wmaschner@csi.com, 1998.

[21] M. Mitchell. *An Introduction to Genetic Algorithms.* MIT Press, 1996.

[22] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):696–710, July 1997.

[23] A. O'Toole, H. Abdi, K. Deffenbacher, and D. Valentin. Low-dimensional representation of faces in higher dimensions of the face space. *Journal of the Optical Society of America A.*, 10(3):405–411, March 1993.

[24] A. O'Toole, K. Deffenbacher, D. Valentin, and H. Abdi. Structural aspects of face recognition and the other-race effect. *Memory and Cognition*, 22(2):208–211, 1994.

[25] H. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes in C, the Art of Scientific Computing.* Cambridge University Press, 1988.

[26] Y. Rosenthal, G. de Jager, and J. Greene. A computerised face recall system using eigenfaces. In *Proceedings of the Eighth Annual South African Workshop on Pattern Recognition*, pages 53–57, November 1997.

[27] H. Schwefel. *Evolution and Optimum Seeking.* Wiley, 1995.

[28] J. Sergent. An investigation into component and configural processes underlying face recognition. *British Journal of Psychology*, 75:221–224, 1984.

[29] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterisation of human faces. *Journal of the Optical Society of America, A*, 4(3):519–524, March 1987.

[30] F. Stremler. *Introduction to Communication Systems.* Addison Wesley, third edition, 1990.

[31] B. Tabachnick and L. Fidell. *Using Multivariate Statistics.* Harper Collins, 1989.

[32] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1), 1991.

[33] M. S. Wogalter and B. D. Marwitz. Face composite construction: in-view and from-memory quality and improvement with practice. *Ergonomics*, 34(4):459–468, 1991.

Appendix A

Face images used to obtain the eigenfaces

The following croped face images are the standardised (see section 5.1.2) set of face images that were used to generate the eigenfaces. These are 46 of a set of 50 face images from *The Psychological Image Collection at Stirling (PICS)* maintained by the University of Stirling Psychology Department. The image collection is available from the internet via *http://pics.psych.stir.ac.uk/*. The other 4 images are shown in Fig. 5.9. The set of 46 faces shown below is reffered to as the training set of face images and the set of 4 in Fig 5.9 is reffered to as the test set.

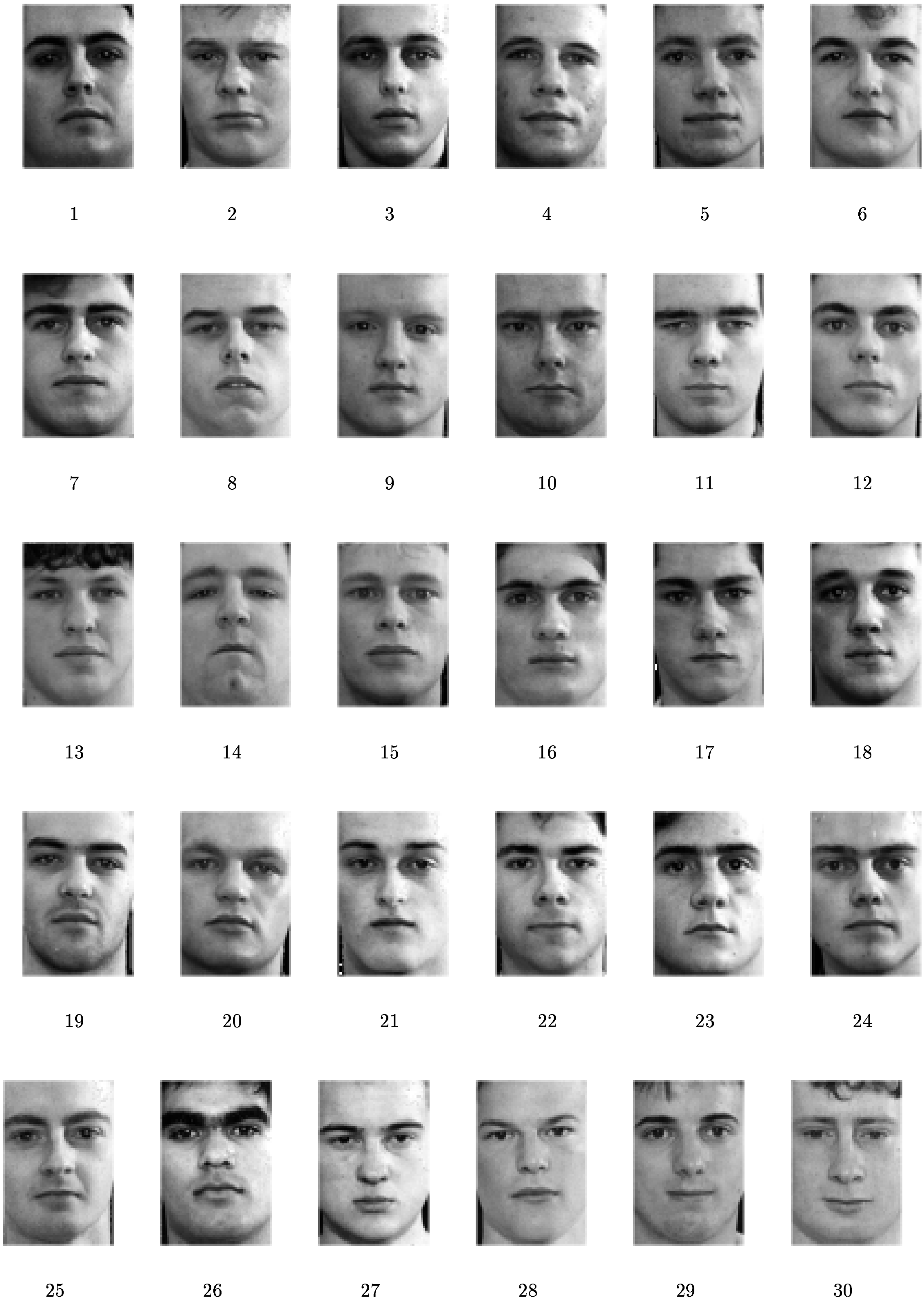
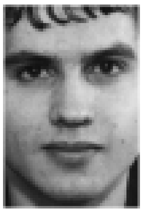
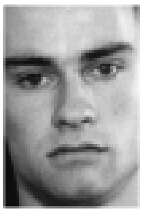


Figure A.1: Part A. Face images used to obtain the eigenfaces

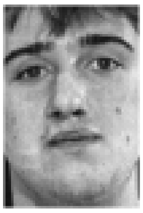
.

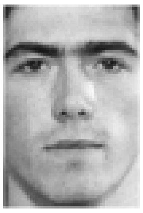
31

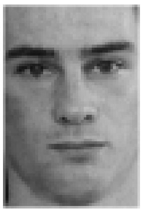

32

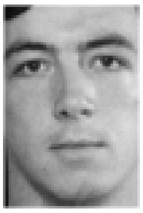

33

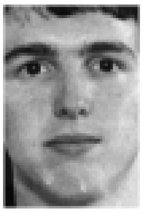
34

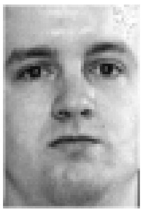

35

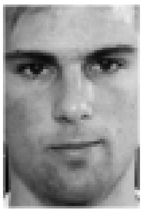

36

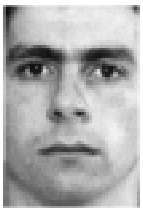

37


38

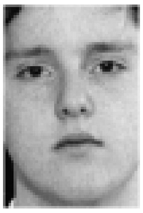

39

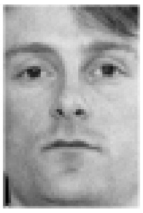

40


41


42

43


44


45

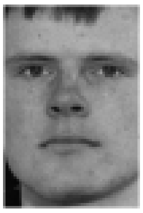

46

Figure A.1: Part B. Face images used to obtain the eigenfaces

.

Appendix B

Experiment 1

This appendix contains the results obtained from experiment 1 and some of the inputs used in it. Some of these results are in Chapter 7, but have been included here for completeness.

All the target faces to be reconstructed are shown in Figure 7.1. The 15 reconstructions of each target are shown in Figures B.2, B.3 and B.4. Table B.1 shows the number of generations used, the best choice number and the time taken for each face likeness image generated in this experiment. Table B.2 gives the judgements obtained on the reconstructions. The last two pages of this appendix contain the lineup pages used in the judging of participants 3 and 11.

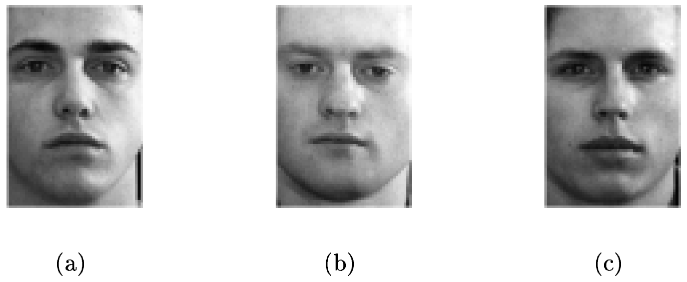

(a) (b) (c)

Figure B.1: The face images to be found in Experiment 1.

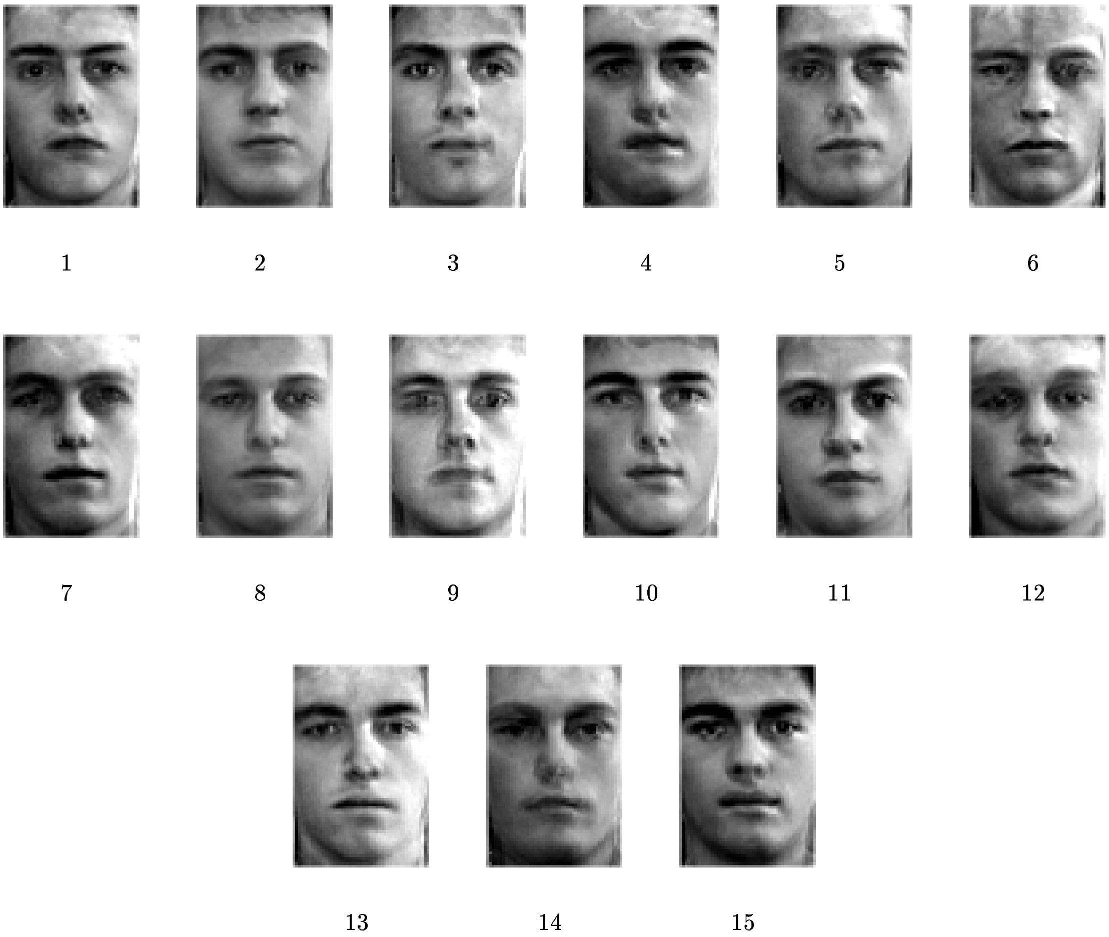


Figure B.2: The results of trying to find the face in Figure B.1(a). The number under each face image is the number of the participant who generated the face likeness image.

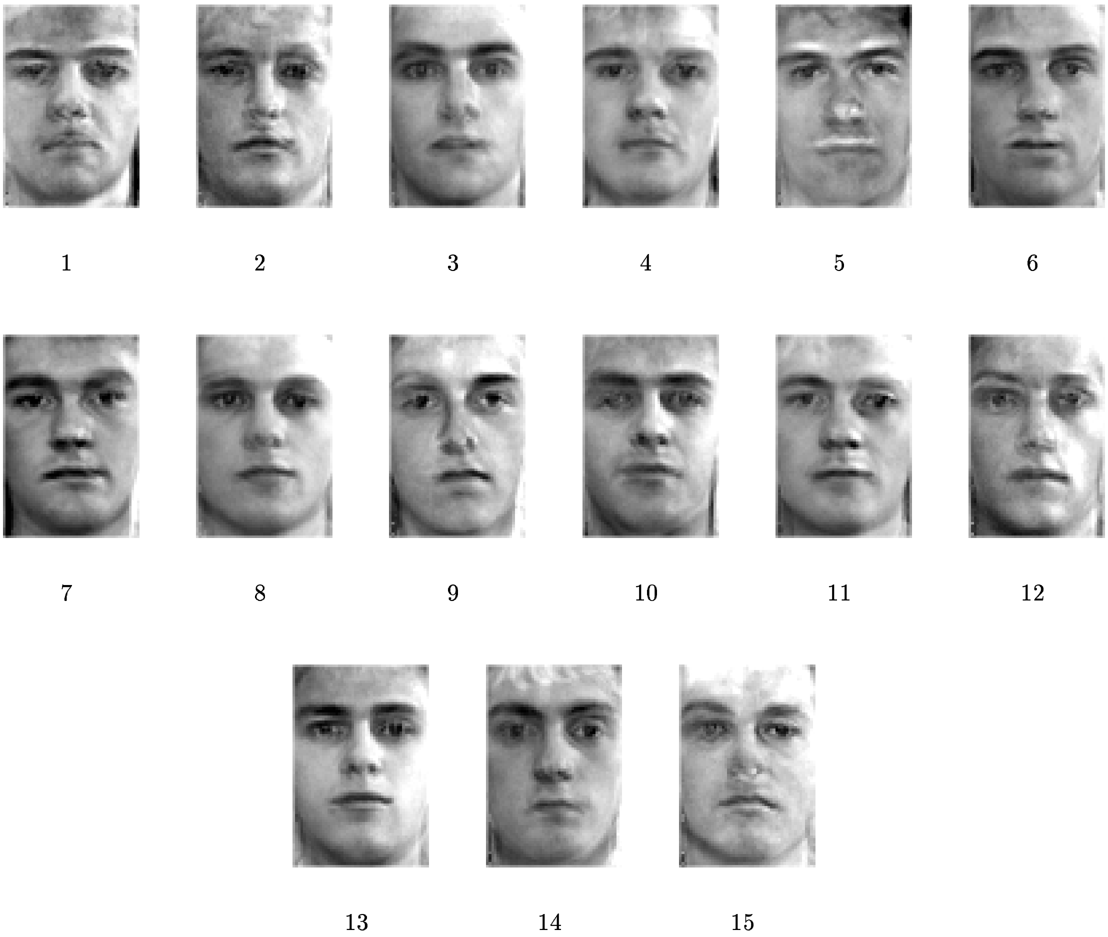


Figure B.3: The results of trying to find the face in Figure B.1(b). The number under each face image is the number of the participant who generated the face likeness image.

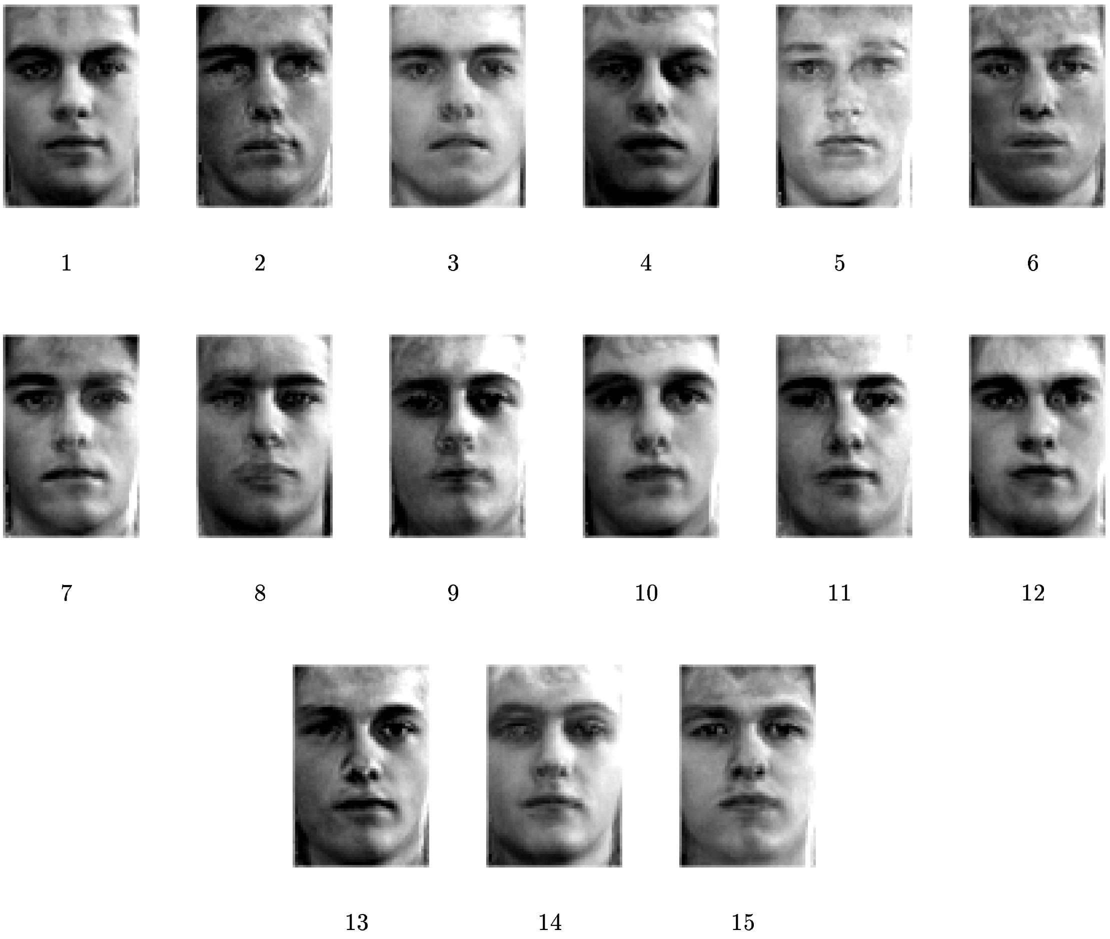


Figure B.4: The results of trying to find the face in Figure B.1(c). The number under each face image is the number of the participant who generated the face likeness image.

	Fig. B.1(a)			**Fig. B.1(b)**			**Fig. B.1(c)**		
	choices	**final choice**	**time (m)**	**choices**	**final choice**	**time (m)**	**choices**	**final choice**	**time (m)**
participant 1	18	16	15	12	9	22	13	11	18
participant 2	9	9	14	7	2	15	8	3	7
participant 3	10	10	10	13	13	15	8	8	9
participant 4	10	10	8	7	5	5	8	6	7
participant 5	6	5	5	7	5	7	9	9	9
participant 6	17	17	5	17	17	6	19	18	5
participant 7	20	18	8	15	13	6	19	10	10
participant 8	11	11	7	21	21	18	9	9	7
participant 9	15	15	12	16	14	16	17	14	28
participant 10	15	13	13	19	14	24	13	13	10
participant 11	27	15	25	16	15	11	22	15	10
participant 12	9	6	11	6	1	11	13	13	13
participant 13	24	20	10	21	21	6	13	11	6
participant 14	12	11	7	20	11	8	17	15	10
participant 15	29	20	8	28	16	6	30	19	8

Table B.1: The number of generations used, the best choice and the time taken for each reconstruction in Experiment 1. The time is from when the first population of faces was displayed on the GUI until the participant pressed stop.

	Fig. B.1(a)		**Fig. B.1(b)**		**Fig. B.1(c)**	
	correct	**incorrect**	**correct**	**incorrect**	**correct**	**incorrect**
participant 1	**14**	2	8	8	4	12
participant 2	8	**11**	**12**	7	7	**12**
participant 3	**10**	5	**9**	6	0	**15**
participant 4	**12**	5	**16**	1	**11**	6
participant 5	**16**	3	4	**15**	3	**16**
participant 6	**10**	8	**12**	6	0	**18**
participant 7	**13**	2	**9**	6	7	7
participant 8	**14**	4	**15**	3	6	**12**
participant 9	**11**	7	8	**10**	1	**17**
participant 10	**11**	9	**11**	9	**11**	9
participant 11	**12**	6	8	**10**	**15**	3
participant 12	**14**	3	**11**	6	5	**12**
participant 13	**14**	4	**13**	5	2	**16**
participant 14	**13**	6	**11**	8	5	**14**
participant 15	**12**	8	5	**15**	1	**19**

Table B.2: Tabulation of the results obtained from the evaluation of the Experiment 1 lineups. The table shows for each reconstruction performed in Experiment 1 the number of judges that selected the correct face in the lineup and the number of judges that selected an incorrect face. For each reconstruction the larger of the two is in bold.

lineup1

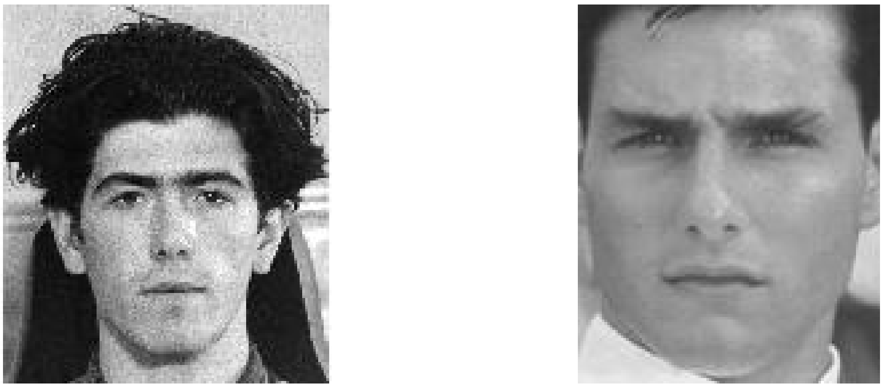
lineup2

Appendix C

Experiment 2

This appendix contains the results obtained from Experiment 2 and some of the inputs used in it. Some of these results are in Chapter 7, but have been included here for completeness.

All the target faces to be reconstructed are shown in Figure 7.4. The 15 reconstructions of each target are shown in Figures C.2 and C.3. Table C.1 shows the number of generations used, the best choice number and the time taken for each face likeness image generated in this experiment. Table C.2 gives the judgements obtained on the reconstructions. The last two pages of this appendix contain the lineup pages used in the judging of participants 6 and 13.



(a) Strange face.

(b) Familiar face.

Figure C.1: The face images to be found in Experiment 2. The strange face is a face image that came from the same set of faces that where used to generate the eigenfaces. The familiar face is Tom Cruise, a well known actor.

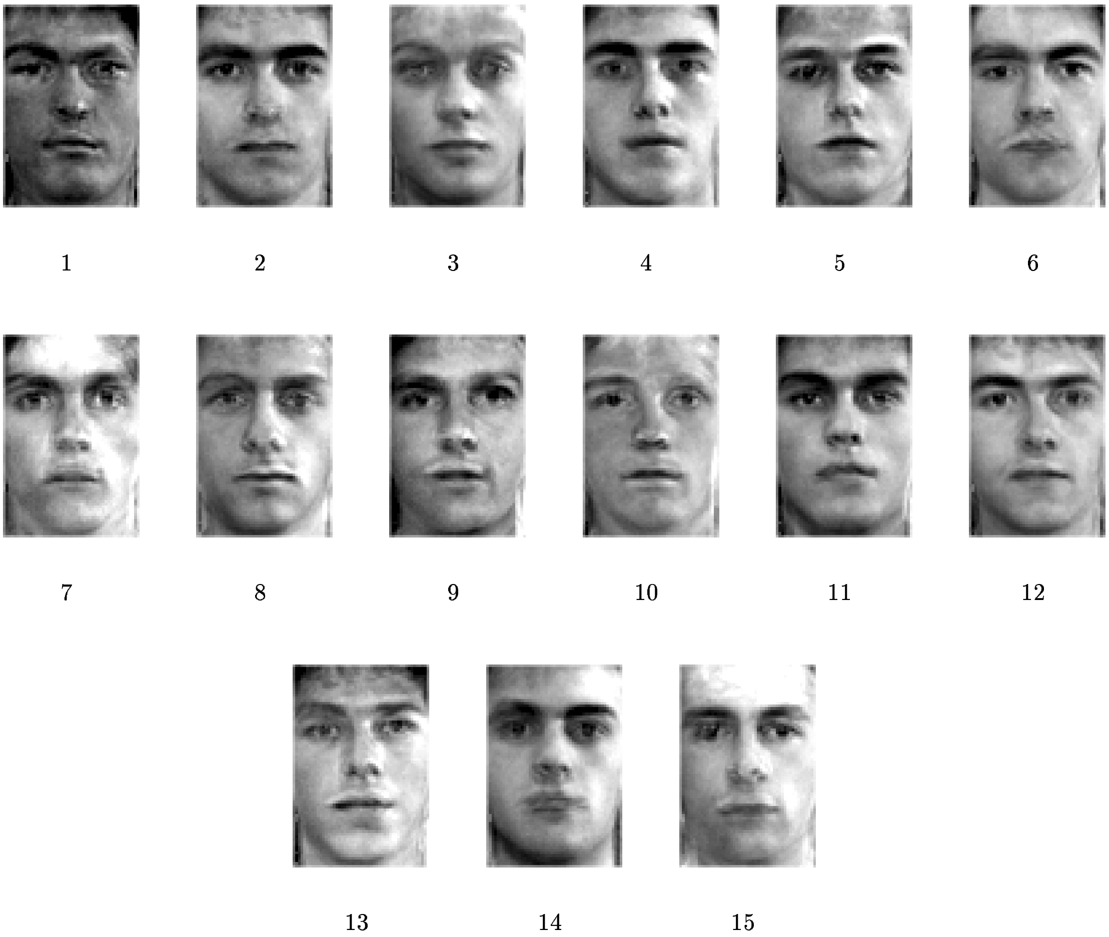
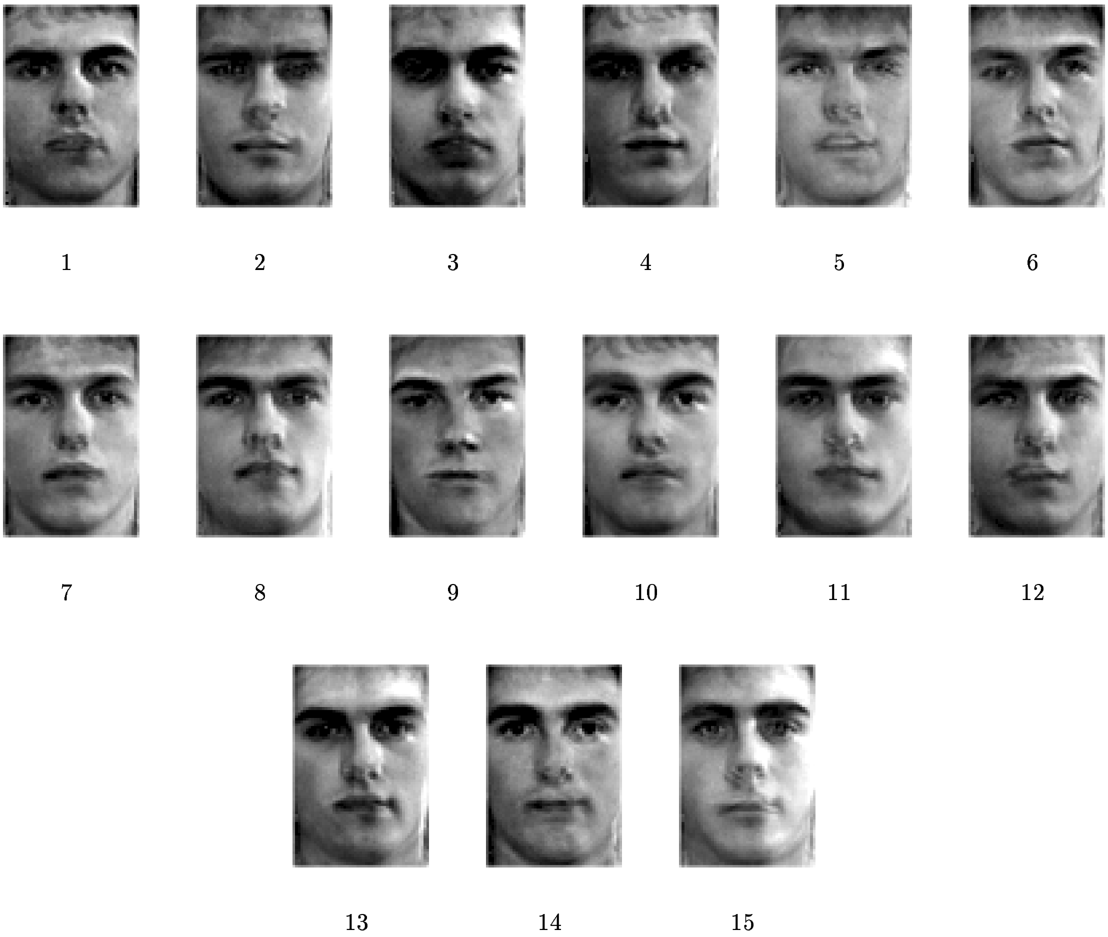


Figure C.2: The results of trying to find the unfamiliar face from memory. Figure C.1(a) is the face which is to be reconstructed. The number under each face image is the number of the participant who generated the face likeness image.



1 2 3 4 5 6

7 8 9 10 11 12

13 14 15

Figure C.3: The results of trying to find the familiar face from memory. The familiar face was Tom Cruise's. Fig C.1(b) was shown to the participants to refresh their memory. The number under each face image is the number of the participant who generated the face likeness image.

	strange			**familiar**		
	choices	**final choice**	**time (min)**	**choices**	**final choice**	**time (min)**
participant 1	25	25	11	30	26	8
participant 2	30	19	11	30	23	9
participant 3	12	8	6	15	11	4
participant 4	14	14	10	10	9	7
participant 5	17	11	12	21	19	13
participant 6	22	7	12	22	17	12
participant 7	18	17	15	21	19	9
participant 8	16	14	18	28	25	28
participant 9	18	5	8	30	25	9
participant 10	17	8	14	19	15	16
participant 11	20	16	12	25	8	13
participant 12	29	25	28	23	21	27
participant 13	9	7	8	13	7	9
participant 14	24	9	6	16	16	6
participant 15	23	23	13	20	16	9

Table C.1: The number of generations used, the best choice and the time taken for each reconstruction in Experiment 2. The time is from when the first population of faces was displayed on the GUI until the participant pressed stop.

	strange face		**familiar face**	
	correct	**incorrect**	**correct**	**incorrect**
participant 1	5	**8**	**9**	4
participant 2	7	7	**13**	1
participant 3	0	**16**	8	8
participant 4	3	**10**	**7**	6
participant 5	0	**16**	**10**	6
participant 6	6	**10**	**9**	7
participant 7	3	**14**	**12**	5
participant 8	1	**16**	**14**	3
participant 9	0	**14**	3	**11**
participant 10	2	**14**	5	**11**
participant 11	0	**15**	5	**10**
participant 12	4	**9**	**11**	2
participant 13	5	**11**	**12**	4
participant 14	1	**14**	**9**	6
participant 15	2	**14**	5	**11**

Table C.2: Tabulation of the results obtained from the evaluation of the Experiment 2 lineups. The table shows for each reconstruction performed in Experiment 2 the number of judges that selected the correct face in the lineup and the number of judges that selected an incorrect face. For each reconstruction the larger of the two is in bold.