EigenFIT: A statistical learning approach to facial composites

Stuart James Gibson

School of Physical Sciences
University of Kent

Thesis submitted for the Degree of Doctor of Philosophy at the University of Kent

· July 2006 ·

Supervised by Dr. Christopher John Solomon

Abstract

This thesis describes the technical concept, design and implementation of a novel facial composite system which exploits the intrinsic human capacity for facial recognition and comparison.

Existing commercial systems for the computer generation of facial composites suffer from some well-documented weaknesses, notably the tendency to focus exclusively on featural information. The scientific, psychological and operational motivation for a new approach is first outlined and then the basic design presented. A system based on this novel approach, known as EigenFIT, is adaptable and exploits the concept of knowledge integration in which global, featural and semantic information supplied by a witness can all be seamlessly incorporated into the composite construction process.

The work presented is primarily concerned with the generation and manipulation of plausible face stimuli which are displayed in a manner that is matched to the needs of witnesses and composite operators. A model of facial appearance based on statistical learning procedures is outlined. The thesis shows how this model can be combined with suitable image processing techniques to generate near photo-realistic composite faces across gender, ethnic origin and age.

Also described is advanced functionality which allows a witness to directly manipulate individual features, automatically age faces, combine faces and alter perceived facial attributes in a simple and direct fashion. A novel exploration of the caricature effect and its potential impact on effective composite production is also presented. Preliminary results of both laboratory testing and trials of the prototype system and its experimental, operational use by U.K. police forces is discussed.

> *A computer terminal is not some clunky old television with a typewriter in front of it. It is an interface where the mind and body can connect with the universe and move bits of it about.*
>
> Douglas Noel Adams

Publication list

Publications emanating from the work presented in, or related to, this thesis.

Papers published in journals and refereed conference proceedings

- C.M. Scandrett, C.J. Solomon, and S.J. Gibson. A person-specific, rigorous aging model of the human face. *Special Issue Pattern Recognition Letters: Vision for Crime Detection and Prevention*, 2006 (in press).

- S.J. Gibson, C.J. Solomon, A. Pallares-Bejarano. Non-linear, near photo-realistic caricatures using a parametric facial appearance model. *Behavior Research Methods*, 2005, 37, 170-181.

- S.J. Gibson, A. Pallares-Bejarano, C.J. Solomon, M.I.S. Maylin Synthesis of photographic quality facial composites using evolutionary algorithms. *Proceedings of the British Machine Vision Conference 2003*, Editors R. Harvey and J.A. Bangham, pp221-230.

- V.S. Johnston, C.J. Solomon, S.J. Gibson, A. Pallares-Bejarano. Human Facial Beauty: Current Theories and Methodologies. *AMA Archives of Plastic Surgery (Special Topics)*, 2003, 5, 371-377.

- S.J. Gibson, A. Pallares-Bejarano, C. J. Solomon. Facial Attribute Manipulation for Composite Systems and Computer Graphics Applications. *Proceedings of the 3rd IASTED international conference Visualisation, imaging and image processing, VIIP 2003*, Benalmadena, Malaga, España. Sept. 2003.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank the following people:-

- My supervisor, Christopher Solomon for embracing the true spirit of academic research and also for his support and understanding.
- Alvaro Pallares Bejarano for his work on evolutionary algorithms that contributed to the EigenFIT project.
- Matthew Maylin who recoded the original MATLAB prototype of EigenFIT in C++.
- Victor Johnston for his early work on use of evolutionary techniques in facial composite applications and also the support and enthusiasm he has shown for the EigenFIT project.
- Graham Pike and his colleagues at the Open University for performing cognitive tests on EigenFIT.
- Alvaro, Matthew and Catherine Scandrett for their friendships during my period of postgraduate research at the University of Kent.
- I would especially like to thank Rana for her love, support and patience.

Chapter 1

Introduction to facial composites

In the event of a crime, police officers often rely to some extent on a witness to provide a comprehensive account of the incident. In some circumstances, the witness has to convey a description of the perpetrator based only on a brief encounter. The pertinent question is how do you accurately convey the perpetrator's face when the image only exists as a memory in the witness' mind?

This corresponds well to the typical circumstances under which a trained police officer will subsequently work with the victim (or other witness to a crime) in an attempt to produce a facial likeness or facial composite of the perpetrator. Unless the witness is a gifted artist it is unlikely that he or she will be able to provide a reliable sketch of the offender/perpetrator. Typically, assuming of course that the attacker is unknown to the witness, he or she will first be asked to provide a detailed verbal description of the attacker and to recount the incident in as much detail as possible. When the interview is complete, an attempt is then made to produce a likeness under the guidance of a specially trained operator. Whilst sketch artists are still used widely in the U.S., this process will most likely (in the U.K. at least) use some form of computerized *facial composite system.* A facial composite system is therefore a tool designed to allow the expression of the facial appearance retained in the witness' memory in some tangible form such as a digital image or computer print-out. The desired outcome is that the generated composite be of sufficient accuracy that subsequent display to members of the public will result in direct recognition and that the details of the suspect will be supplied to the police. In many cases, a generated composite may not be accurate enough to produce a definite 'hit' but will nonetheless provoke members of the public who recognize basic similarities to provide the names of possible suspects. In most cases, it is the combination of the composite with other basic information such as age, build, domicile and the type of crime that results in the provision of suspect names.

The process by which a witness and a composite operator arrive at a final facial composite is a complex interplay of computer imaging and human cognitive function and the final result depends on a number of factors. The overall success of the composite process is, first and foremost, reliant on the witness' ability to retain some memory of the face in question. Undoubtedly some people are better equipped to perform this task than others. Other factors such as the witness' state of mind (i.e. they may be in various degrees of shock as a result of the crime), the period of time over which the crime took place, the proximity of the perpetrator to the witness during the crime, and the time elapsed between the crime and the composite construction will also affect the memory. From a scientific and technological perspective, there are critical aspects to consider in the design of an effective composite system. It should provide sufficient flexibility of use and image quality to meet the needs of different witnesses and operators and should be constructed, as much as possible, to match their normal cognitive processes.

In the absence of any photographic evidence such as CCTV footage, the witness' memory of the perpetrator will be the only means for constructing a resemblance to the face and it is vital to the successful progress of many criminal investigations that the best possible result be obtained. Although systematic methods for remembering faces have been outlined by Penry, the inventor of the PhotoFIT system [71] it is unreasonable and impractical to expect that potential witnesses and victims will be well trained in these techniques. Rather, the emphasis must be on the design of composite systems and associated interview techniques which allow the best evidence to be produced by ordinary members of the public. A substantial body of psychological research (which is discussed in more detail in the following sections) suggests that the vast majority of existing facial composite systems may not operate in a way which allows this.

The key point emerging from this work is the relative *weakness of human-beings at the process of recall and description* of faces as contrasted with their remarkable capacity for face recognition. In simple terms, to first recall and then accurately describe a face, even that of a family member or a close relative, is cognitively difficult. The facial composite systems currently favoured by international police forces rely on a construction process in which individual facial features (eyes, nose, mouth, eyebrows, etc) are selected one at a time from a large library and then electronically 'overlaid' to make the composite image.

Although substantial improvements have been made over earlier systems such as IdentiKit I and PhotoFIT, a considerable body of evidence now suggests that the task of face recognition and synthesis does not lend itself to simple decomposition into features and is, at least partly, a global process [100, 12, 67] which relies on the inherent spatial/textural relations between all the features in the face. Our ability to visualize facial features is highly variable and the accurate verbal description of faces notoriously difficult (the vocabulary of the verbal medium is simply not matched to the direct cognitive experience of seeing and recognising a face). The human face perception mechanism seems designed primarily to recognise faces and it is accepted that the recollection and visualization of even familiar faces (let alone unfamiliar faces seen only briefly) is more difficult. Indeed, previous research has suggested that the need for the witness to recall and verbally describe the face to the operator could be the weakest link in the composite construction process [94]. These two basic facts currently mean that the generation of facial composites using isolated facial features is time-consuming (sometimes taking many hours) and requires tasks (recall and verbalization) that the witness may find very difficult.

The work described in this thesis is an attempt to explore and develop an alternative approach to facial composite production which directly exploits the human capacity for tasks related to facial recognition. The basis of this approach consists of two essential parts - i) A generative (and near photo-realistic) model of human facial appearance which is able to randomly produce plausible human faces and ii) An interactive evolutionary algorithm in which new groups of faces are produced as a result of the witness' responses to previously generated faces. This thesis also describes techniques which effectively allow the direct incorporation of specific featural and semantic information provided by a witness.

Chapter 2 outlines the central mathematical concepts and foundations on which much of this thesis is based. The essential conceptual and computer implementation of this approach to composite construction (termed EigenFIT) is described in Chapter 3. Chapter 4 describes four additional tools that enhance the functionality of the composite procedure detailed in the preceding chapter, by integrating semantic and feature based knowledge with holistic representations of the face. A novel technique for fast geometric transformation (warping), which has important practical implications for a system based on this approach, is described in Chapter 5. Chapter 6 outlines work on non-linear caricature transformations, which can be described using the facial appearance model, and its implications for composite production and beyond. Finally, Chapter 7 gives a summary of the work to date, and a discussion outlining proposed future developments.

In the remainder of this chapter, the focus is on the motivation for this work. First, an overview of the better known existing commercial composite systems is offered and then the current developments in the field are reviewed.

1.1 Review of established composite methods

1.1.1 The artist's sketch

One approach to generating a likeness to a perpetrator is to employ a police sketch artist. Sketch artists perform the same function as a composite system and its operator by translating a witness' verbal description into a pencil drawing of the perpetrator's face. The exact technique varies from artist to artist. The procedure adopted by American sketch artist Gil Zamora [101] is given as a specific example. The witness is interviewed, beginning with a few preliminary questions about the age, race and body build of the suspect. Once these questions have been answered, more specific enquiries are made about the face shape, hair and ears. The internal facial features are the last part of the sketch to be drawn, working around the face and concluding with the eyes. The witness remains seated during the interview with their eyes closed during the first five minutes of questioning. This helps them to relax and focus. After the initial sketch is finished the witness must comment on its accuracy and make suggestions for improvements. The whole process takes approximately 45 minutes. Since no dataset is required, the range of different faces that can be produced is limited only by the artist's knowledge of natural facial variation. The drawbacks are that the artist needs to be highly skilled in both their drawing ability and interview technique. Artistic ability is not a skill that is easily acquired, therefore, unlike many other methods used in criminal investigations, it can not be easily taught. Furthermore, the artist's sketch is by its very nature a subjective interpretation of the witness' verbal description, and as such is liable to inaccuracies unless a strong understanding is established between the witness and artist.

1.1.2 Identikit

Developed by P.J. Dunleavy and released in 1959, Identikit I was the first alternative to the sketch artist. Indentikit I was a mechanical system based on line drawings of individual facial features. The line drawings were printed on transparencies and overlaid to produce the composite image. Hence a likeness

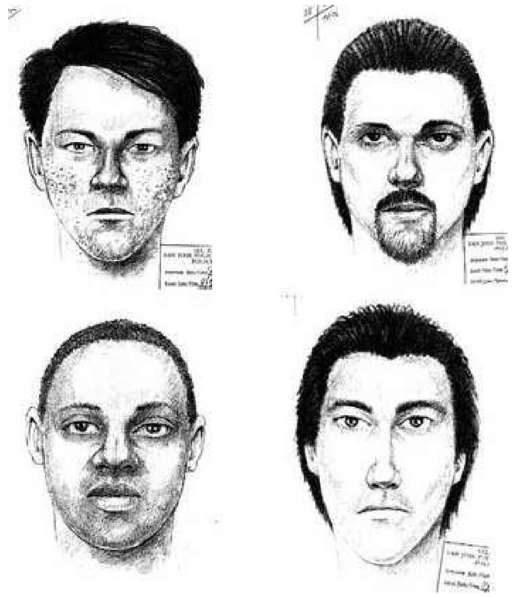


Figure 1.1: Likenesses to offenders, produced by sketch artist Gil Zamora.

to a chosen face could be 'composed' from a selection of constituent features or parts, leading to the term facial composite. The system contained a limited library (for example, there were only 130 hairstyles) from which the witness was required to choose a set of appropriate facial features. The features could not be resized or moved in relation to each other. Identikit I was superseded in 1975 by Identikit II which used monochrome photographic features rather than line drawings. These were the predominant systems of choice in the U.S., a similar system called PhotoFIT being more prevalent in the U.K.

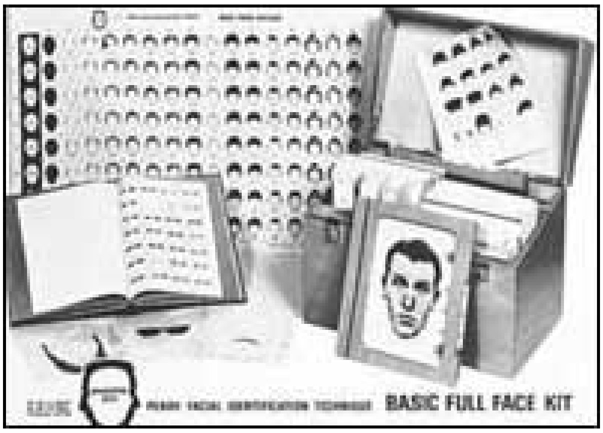


Figure 1.2: Identikit composite pack. Facial features were printed on card. The chosen features were slotted into a frame, forming the composite image.

1.1.3 PhotoFIT

PhotoFIT is an acronym for Photographic Facial Identification Technique. This mechanical composite system was invented by Jacques Penry and introduced to U.K. police forces in April 1970 with the backing of the Home Office. As the name *Photo*FIT suggests, this system used monochrome photographs of exemplar faces. The photographs were reproduced on card and cut into five separate pieces relating to specific facial regions. The available regions were chin, mouth, nose, eyes with eyebrows, and a single card including the forehead, hair and ears. 550 different features were provided for selection by the witness with guidance from a trained operator. Sorting through all 550 cards would be cumbersome, so a book referred to as a visual index was used for selection purposes. The cards corresponding to the features selected from the visual index were then arranged in a frame to form the composite face. If required, additional details such as scars were drawn on transparencies and overlaid on the composite image. In the UK, the name PhotoFIT became synonymous with the term facial composite. It is occasionally used today when referring to composites, despite the fact that the system itself has been superseded by computer software packages that perform the same task more efficiently. One of the main benefits of PhotoFIT over other systems was its photographic feature library. Ironically, this was also one of its disadvantages since the joins between different face regions remained visible in the final composite. This problem was not so apparent in the less ambitious systems which used simpler line drawings.

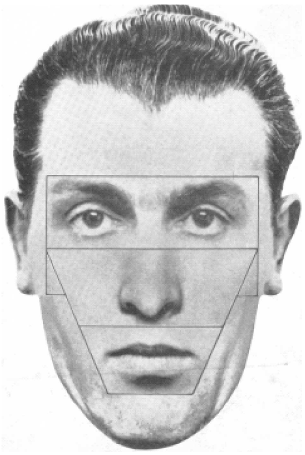


Figure 1.3: Diagram indicating the feature components used in the card based PhotoFIT system.

1.1.4 E-FIT (for Windows)

E-FIT (Electronic Facial Identification Technique) [3] is a computer software package that runs on the Windows operating system. E-FIT is underpinned by the same feature based methodology as PhotoFIT, but has considerably more functionality, and also attempts to address some of PhotoFIT's psychological deficiencies. Bennett [3] outlines the steps taken to build a more reliable composite system as requested by the U.K. Home Office and recommended by the Aberdeen University Psychology Department [29]. Since its inception E-FIT has become the most advanced commercially available composite software package to date, and is used by police forces and security services across the world. Due to the complexity of this package, it is essential that a trained operator work with the witness. The first and arguably most important step in creating a composite using the E-FIT system is for the operator to interview the witness in order to acquire a verbal description of the suspect. According to Aspley, the manufacturers of E-FIT, an operator who has received comprehensive training in cognitive interviewing methods can expect to obtain 40% more information than normally obtained using standard interview techniques. An extensive library of facial features (in this context referred to as a database) is provided comprising exemplar images of hair, eyebrows, eyes, nose, mouth, ears and face-shape (including chin) regions. Searching the entire database for suitable features would be prohibitively time consuming, hence the description is entered into the system by means of a series of radio buttons from which the system's 'Intellisearch' algorithm produces an initial exploratory composite image. This 'first guess' likeness functions as a starting point from which the final composite can be created. The witness instructs the operator to swap or modify specific features with which he/she is unsatisfied, thereby improving the likeness to the suspect. In this respect, the facial features are manipulated in context, i.e. within the face. This is a departure from systems that require the witness to select features using only a visual index. There are psychological advantages in favour of modifying the features in-situ. It should be made clear that the process described here is a pseudo-global technique. The E-FIT system, like the mechanical systems that pre-date it, is inherently a feature-based approach to facial composite production. However, it does have some advantages over other systems. Features can be moved and rescaled (independently in the horizontal and vertical directions) and blended together to produce almost seamless joins between different face regions. It also has a wide range of beards, moustaches and other facial appendages that can easily be added to the composite. At present, E-FIT sets the benchmark for composite systems.

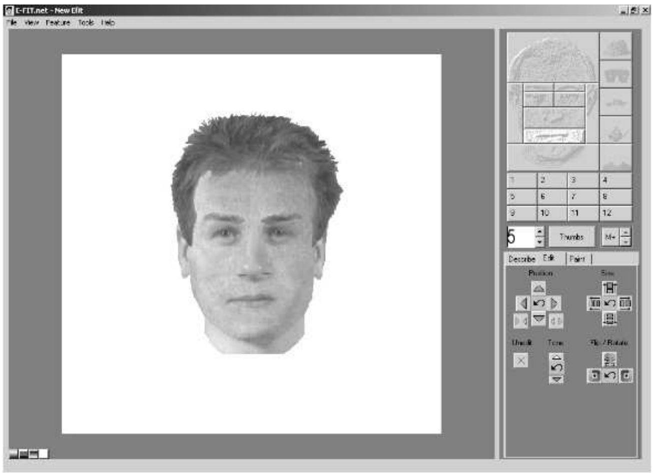
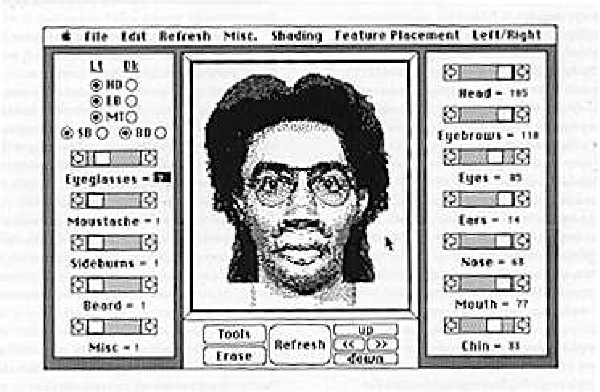


Figure 1.4: Graphical interface for the E-FIT composite system.

1.1.5 Mac-a-Mug

Developed by Shaherazam, this system was introduced to U.S. police forces in the mid 1980s. The software, as the name indicates, only runs on Macintosh computers. The graphical user interface is depicted in figure 1.5. The system is based on the use of sketch-like individual facial features that are overlaid on top of a template forming a homogeneous composite without feature boundaries. Each feature can be independently resized, rotated and translated. This feature transformation option was a major improvement over previous systems, allowing a large number of combinations to be created from a small library of approximately 500 base features and facial appendages.

The first study on the performance of the Mac-a-Mug system was undertaken by Cutler, Stocklein and Penrod [24]. In this study, an expert operator created composites of different targets, which were always visible during the composite process. Participants in the experiment were asked to match the composites to photographs of the real target faces. An astonishing 49% accuracy was recorded in this experiment, implying that the Mac-a-Mug system was highly successful in creating realistic composite images. Later studies carried out by Koehn and Fisher [58], in which the target faces were not shown during the compositing process, indicated that real performance of the system was a mere 4%. In the same experiment, a trained operator also created composites from life, increasing the performance of the system to 77%, emphasizing that the problem with reconstruction lies in the capacity of the witness to remember a face.



Eyeglasses =
Moustache = 1
Sideburns = 1
Beard = 1
Misc = 1
Head = 105
Eyebrows = 118
Eyes = 85
Ears = 14
Nose = 63
Mouth = 77
Chin = 33

Figure 1.5: Mac-a-Mug GUI.

1.1.6 ComPHOTOfit

ComPHOTOfit is distributed by American based company, Sirchie Inc. [81]. The most recent version contains 1,500 colour images of facial features and appendages. Features can be positioned, resized, moved, copied and modified as requested by the witness. Accessories such as glasses, hats, moustaches and beards can be added and some functionality for introducing aging effects is also provided. The software is currently used by over one thousand law enforcement agencies worldwide and is the most widely used composite system in America at present.

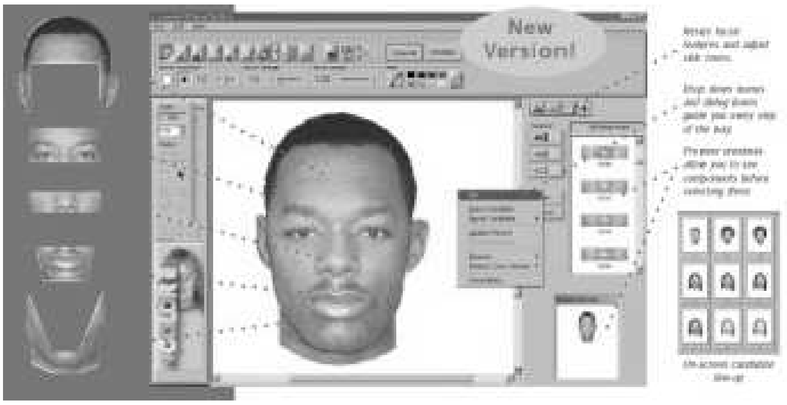

New Version!
Resize facial features and adjust skin tones.
Drop down menus and dialog boxes guide you every step of the way
Preview windows allow you to see components before selecting them
On-screen candidate line-up

Figure 1.6: Comphototfit graphical user interface.

1.1.7 PROfit

PROfit is a feature based system currently available and marketed by ABM [1]. It replaced ABM's previous composite software, CD-fit. PROfit contains a comprehensive database of 20,000 facial features spanning Afro-Caribbean, Caucasian, Mediterranean, North African, and Far Eastern ethnic groups. Features can be re-sized, repositioned, lightened or darkened. An internal drawing package allows modification of existing facial features and the addition of distinctive marks such as scars. Composites of the same subject, produced by different witnesses, can be morphed together with the aim of producing a more accurate likeness. 3/4 fits can also be made where the witness only had a partial view of the offender. The software interfaces with another ABM product called Profile, which compares the composite image to a database of face images.

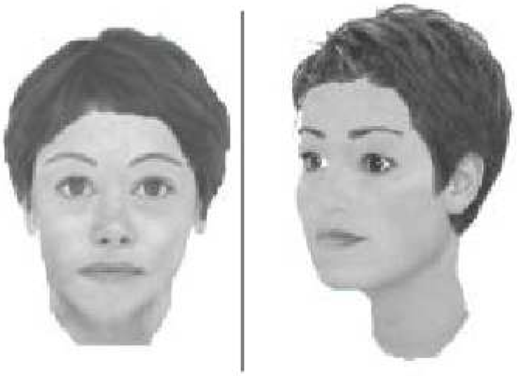


Figure 1.7: PROfit is capable of producing 3/4 view composite images.

1.2 The psychology of generating facial composites

1.2.1 Face recognition and retrieval

An essential requirement of any composite system is its ability to represent facial variation from an appropriate population. For instance, a system that can only represent white females will be ineffective for creating likenesses to Chinese males. Composite systems to date have been limited in this respect due to the extent of their feature libraries/databases. Conversely, a sketch artist is only restricted by knowledge of different face types and an ability to capture the essence of these faces in a sketch. Later systems (notably E-FIT) address this problem by compiling extensive databases of facial features. With databases increasing in size, the problem of organizing and accessing the components for building composite faces becomes more complex. Storing, and to some degree retrieving, images of faces is a task which the human mind performs countless times each day. It is an essential part of our social interaction required for everyday life.

Geometric models of how the mind organises face imagery have been proposed. The generic term for such a model is *face-space*, first proposed by Valentine [96]. In a face-space model, the face of an individual subject occupies a particular position within the space, and spatial relationships between subjects can be used to explain effects including distinctiveness, caricature, inversion and race. Since the process of generating a composite is effectively an attempt to retrieve a face from memory, advantages may be gained in developing a composite system that operates in a manner that is harmonious with face-space.

Valentine formalized two abstractions of face-space; *norm based coding* (NBC) in which distinctiveness is governed by the distance from a central prototype face, and *exemplar based coding* (EBC) in which the face is judged purely by its proximity to other faces. Making a distinction between these two representations can be problematic. If the exemplars are normally distributed in face-space, the region of maximum exemplar density will correspond to a central prototype as described by the NBC model. Both models predict that typical faces occur in densely populated regions of face-space and distinctive faces are located in sparsely populated regions of face-space. Valentine and Endo [97] found EBC to be superior when explaining the effect of race on face processing. Conversely, the effect of caricature is explained more simply by NBC than EBC. It appears that these two variations of the face-space model need to be unified to form a single, succinct, representation that accounts for all aspects of face recognition and retrieval. The exact nature of the dimensions of face-space are also a point of debate.

Studies have attempted to determine a relationship between statistical properties of images and human face processing. Hancock et al [44] performed a principal components analysis on a set of suitably aligned digital face images to form a multidimensional space in which each axis corresponds to a specific global (referring to the whole face, not individual features) face property. These axes are determined purely on the statistics of a sample of face images and, as such, are not ordered in terms of perceptual importance. Subsequent psychological experiments were performed to relate the principal components to psychological aspects of face perception. The early components were shown to embody very general information regarding the appearance of faces, the suggestion being that the higher components of the analysis were more important in determining if a face is memorable. The PCA approach lends some weight to the validity of NBC since faces perceived as distinctive tended to occur at greater distances from the mean than common/indistinct faces.

1.2.2 Local features vs global methods

To date, composite systems have depended on libraries or databases of individual features from which composites faces can be constructed. The preference of individual facial features over global (whole face) components is due primarily to technical issues regarding implementation and possibly due to the path that the historical development of composite systems has taken. Nevertheless, there is no *a priori* reason to suppose that a piecewise arrangement of individual facial features is the best approach [73]. In fact, there is strong psychological evidence to suggest that faces are recognised as a whole rather than as a sum of their constituent parts [79] [12]. The implication is that the configuration of facial features should be acknowledged as an important factor in any reliable composite strategy.

Tanaka and Farah [89] performed an experiment in which subjects were presented with twelve different faces. Subjects were then asked to identify facial features belonging to the original face stimuli from a scrambled arrangement of features. The results of the experiment indicated that individual facial features were 10% more likely to be correctly identified when they are displayed in their normal configuration within the face. Similar experiments were performed using images of inanimate objects segmented into regions analogous to features. In this case, recognition rates were unaffected by viewing the image segments in isolation. This implies that the configuration of features is important when recognising faces but plays little or no part in the recognition of other objects, an effect known as face superiority.

Turner et al [94] performed three studies investigating the effects of local vs global when using the E-FIT system. The aim was to determine whether constructing a composite within the context of a whole face would offer any advantages over constructing a composite on a feature by feature basis. The first study involved participants creating composite images using ’piecemeal’, ’jigsaw’ and standard E-FIT approaches for selecting features. In the piecemeal method each of the facial features were selected in isolation with the other features hidden from view. The jigsaw process required the witness to add one feature at a time to the composite image in a manner akin to building a jigsaw. In the standard E-FIT approach, features were manipulated in-situ.

According to the participant’s own subjective evaluation, there was no appreciable difference between the correctness of the composites created by the piecemeal and jigsaw methods. Conversely, when the composite images were assessed by independent participants the jigsaw method was judged to be better than the piecemeal method and the standard E-FIT procedure was considered the best method. The results can be interpreted as evidence that working within the context of a whole face-image can lead to more perceptually accurate composites, although it appears that the witness is unable to provide an objective measure of the goodness of the three methods.

Previous work by Haig [42] found the hair and head shape to be highly salient. This finding was corroborated by Young et al [99] who showed that the external features are more important for recognition than the internal features when the face is unfamiliar to the witness. Conversely, when the face is familiar to the witness, the internal features are more significant. In Turner's third experiment feature saliency was investigated, and how the order in which the features were selected affected the quality of composite images. The relative feature saliency was assumed to be determined by the order in which the witness described the components of the target face. For example, if the witness described the eyes before the mouth then this was interpreted as the eyes being more salient than the mouth. One group of participants generated composites starting with the most salient feature (first to be described) and finishing with the least salient feature (last to be described). Another group worked in order from the least salient to the most salient. A third group were allowed to work through the features in any order they desired, as would be the case in normal E-FIT use. The results of this experiment provided evidence to suggest that having the witnesses work on the higher saliency features early in the construction process produced the best quality likenesses.

1.2.3 Verbalization of facial descriptions

Current methodologies for producing composite imagery require the witness to provide verbal descriptions of the assailant. The formal name for this procedure is the *cognitive interview* and its purpose is to provide an initial starting point or 'pre-face' that can be suitably modified to yield the final composite image. It has been suggested that the cognitive interview is a limiting factor in the overall effectiveness of composites [60]. The problem can be considered as comprising two issues: the witness' capacity to verbally describe a face and the ability of an operator to interpret these descriptions reliably. A study by Christie and Ellis [18] proposed that the difficulty is in translating the verbal information provided by the witness into an image, and not in the witness' aptitude for providing an accurate account of facial appearance. To test this hypothesis, participants were asked to construct a PhotoFIT of a target face and also provide a verbal description of the same face. Judges were then instructed to identify a subject from an array of faces based on the participant's verbal descriptions and composite images. The results of the experiment indicated that the verbal descriptions were significantly more accurate than the PhotoFIT composites. Therefore the flaw in the composite procedure was not rooted in the witness' verbal ability but in some other aspect of the system, possibly direct interference between the visualized image and the PhotoFIT composite itself.

A similar experiment was performed by Brace [9] to determine whether composites generated using E-FIT were also susceptible to the misinterpretation of the witness' verbal descriptions. However, the objective of this study was subtly different from Christie and Ellis [18] with the emphasis on how well composites of familiar faces are recognised. Generally, it is anticipated that a composite image will be recognised by someone who is familiar with the suspect, hence famous target faces were used. Composite images of famous faces were constructed from a description or directly by the E-FIT operator based on their own memory of the face(s). E-FIT images that were generated by the operator alone were significantly more likely to be recognised than composites generated from information provided by someone else. The results of this study are in agreement with Christie, confirming that the quality of the composite is limited by the translation of a verbal description.

1.3 Emerging composite technologies

Brunelli and Mich [13] constructed a developmental system named Spotit!, which partly addresses the limitations associated with a finite database of candidate features. In this approach, a principal components analysis is performed on each class of facial feature (eyes, noses, mouths etc), thereby extracting the mathematically salient information and providing a basis from which novel features can be constructed. The 'pre-face' image or starting point in this system is the mean face into which the facial features are set/blended. The appearance of each facial feature is controlled by seven sliders, where each slider corresponds to a principal component or mode of variation. The authors claim that their system provides a "virtually unlimited set of alternative features". This statement is slightly misleading because what the system actually provides is an infinite (within the limits of computational precision) number of combinations of a finite number of basis images. The range of composites that can be produced using this technique is limited by the finite size of sample used in the PCA. However, Brunelli and Mich [13] include a tool that allows the operator to manually distort the shape of a chosen feature. In this sense there is an unlimited set of *feature shapes* that can be achieved. One of the main weaknesses would appear to be that the sliders incorporated in the interface control changes in appearance defined on a mathematical premise, and as such have no specific perceptual meaning (e.g. 'a turned up nose'). Therefore, any prospective witness/operator will find it difficult to locate the optimum slider positions required for a good likeness to the target face.

An 'intelligent' search procedure is required to overcome the difficulty of selecting the most appropriate features from an almost unlimited sample. *Genetic algorithms* (GAs) [49] offer a conceptually pleasing solution to the search problem and are prevalent in emerging composite systems. Evolutionary techniques based on Darwinian theory [25] that simulate complex structures, textures, and motions for use in computer graphics and animation have previously been described [80]. More recently they have been used for the purpose of generating avatar faces. DiPaolo [27] describes such an algorithm, based on an aesthetic selection process in which faces are represented by genotypes comprising 25 parameters. The first recorded use of a GA for generating facial composite imagery was Caldwell and Johnston [16]. The GA implementation is initialized with a population of twenty faces which are constructed from individual facial features in a style reminiscent of earlier systems. Faces are displayed to the operator, who is required to assign a *fitness score* to each face depending on its similarity to the target. Parent faces are chosen from the initial population according to their associated fitness score and bred with each other using the principles of crossover and mutation. This process is described in more detail in the following chapter.

All of the composite systems described so far rely on databases or libraries of facial features. In section 1.2, the limitations of the feature based approach were highlighted [29]. Attempts have been made to incorporate information concerning the configuration of facial features into feature based systems such as E-FIT, and the effectiveness of these ad-hoc 'pseudo holistic' approaches have been examined [95]. However, a more elegant, and possibly more effective approach is to model facial variation as a whole. Hancock [43] describes a developmental system that utilizes both global PCA face models and a GA. This design allows composite images to be created by adjusting global/holistic properties of facial appearance, in a way that is not too demanding of the witness. Unlike previous systems this method is truly global, relying on whole face templates (the principal components) rather than a database of facial features. In this context the principal components are often referred to as *eigenfaces* [82], [93] with the first few components describing most of the variation exhibited in the human face (assuming an appropriate training sample). Hancock used two separate PCA models, one for face shape and another for pixel intensity values. Using two independent models overcomes problems associated with head pose and blurring which would otherwise degrade the composite images. PCA parameter values are not controlled by sliders as in Brunelli and Mich [13], instead the operator is presented with a selection of eighteen faces to which fitness ratings must be assigned on a scale of zero to ten. The genetic algorithm selects faces with a high rating (fitness proportionate selection) as parents. Parameters defining an offspring's appearance were selected at random from the parents (uniform crossover) and a mutation applied to some of the parameter values. This procedure was performed eighteen times to form a new *generation* of faces. Hancock's PCA model was built on a limited sample of twenty female faces. The system has been subsequently refined by Frowd [35] and is now known as EVO-FIT. Other systems based on evolutionary/PCA methods have been developed independently of Hancock, suggesting that this is a viable approach to producing composite imagery [83, 36, 92, 77].

1.4 Brief introduction to the EigenFIT system

The subject of this thesis is the technical design and implementation of a facial composite system for use in criminal investigations, provisionally named EigenFIT. Exploratory studies relating to the work in this thesis have previously been presented by Solomon et al [83] and by Gibson et al [36]. Unlike traditional feature based methods, the approach described uses both local and global facial characteristics and allows a witness to produce plausible, photo-realistic face images in an intuitive way. EigenFIT offers two modes of operation termed *EasyFIT* and *ExpertFIT*. The simplicity of the EasyFIT mode makes it suitable for use by the witness with the minimum amount of assistance from the expert operator, whereas ExpertFIT comprises a suite of advanced tools aimed at a trained operator. Conceptually, EigenFIT is constructed from three main components,

- A generative face model
- A search procedure
- A user interface

The main focus of this thesis is the technical development of the generative face model and the user interface and how, in conjunction with the search algorithm, they fuse to form a complete composite system. Figure 1.8 provides an overview of the basic components of the system and the necessary interaction between them required to produce a composite image. A brief literature review of work relating to the generative face model and search algorithm is provided below,

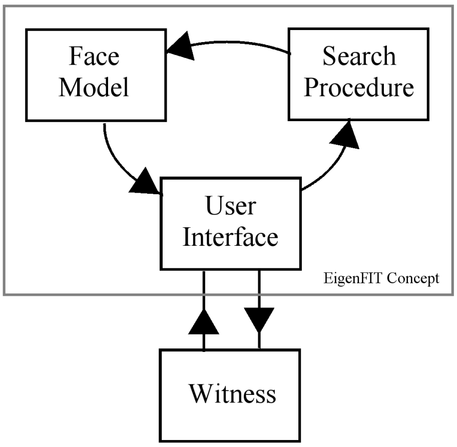


Figure 1.8: A schematic diagram representing the interactions between the main EigenFIT components and the witness.

The **generative face model** (described in detail in chapters 2 and 3) incorporates information extracted from a sample of real face images using *principal components analysis* (PCA) [54]. The first recorded use of principal components for modelling facial variation was Sirovich and Kirby [82], who demonstrated that it could provide a highly efficient representation of a human face as a linear superposition of global principal components or eigenfaces. This seminal paper precipitated such a significant amount of research that the PCA technique is now a standard paradigm in face recognition and both 2-D and 3-D face modelling research. The Sirovich and Kirby method can be used to encode exact likenesses of the faces contained in the original training example. However, the capacity of the method for encoding approximate likenesses to faces that lie outside the original training sample is what makes the eigenface technique particularly useful in computer vision applications. With the rigid image registration process that was employed in their original work, a perfect alignment of facial features was not possible. The misalignment caused ghosting artefacts that were visible in the approximation images of out-of-sample faces. To avoid such artefacts, a better correspondence between facial features was required.

Craw and Cameron [23] solved the correspondence problem by warping the face images to a standard shape prior to performing the PCA. A further development was provided by Cootes et al [20] et al who utilized this shape normalization technique, forming a combined shape texture representation that can be applied to the human face. Cootes refers to this procedure as constructing an *appearance model.* The use of appearance models in automatic object detection and recognition has been well documented, especially in the form of *active appearance models* [20, 64]. Much less explored is the possibility of using PCA in facial synthesis, which is the application described in later chapters.

The **search procedure** employed in the EigenFIT system is an asexual *evolutionary algorithm* (EA). Genetic algorithms (GA) can be regarded as a specific variation on the EA theme. There has been widespread interest in using genetic and evolutionary algorithms to solve optimization problems. In many situations the more traditional calculus and enumerative strategies can be difficult to implement. EAs often offer a simpler solution to optimization and search problems and in some instances are more likely to find the global solution. Other applications for these techniques exist. For example, Fogel et al [31] introduced evolutionary programming for creating artificial intelligence. Evolutionary algorithms (specifically GAs) were made widely known due to pivotal work by, Holland [50] and Rechenberg [75] in the 1970's although their origins can be traced back to two decades earlier [10, 33, 8]. However, the full potential of genetic and evolutionary algorithms was not realized until the 1980's when advances in computer hardware made the proposition of using them viable. Since then evolutionary algorithms have been used various problems in the fields of both computer vision [47] and computer graphics [80].

Mathematical details relating to the generative face model and evolutionary search procedure developed for this thesis are covered in detail in the following chapter.

Chapter 2

Mathematical foundations

The facial composite system described in this thesis is significantly different in design and operation from the previously described composite systems. The aim of this chapter is to provide the theoretical background that underpins the algorithms employed in the system. The chapter begins with an historical introduction to *principal components analysis* (PCA), and the procedure for deriving principal components from sample data. Application of the PCA technique to image data is then discussed, and the necessary preprocessing required to extract shape and texture information from the image data is presented. The construction of a combined PCA shape-texture model (*appearance model*), from which new examples of a constrained object class can be synthesised, is described and accompanied by an illustrative example. Using the appearance model, any object within the modelled object class can be approximated by a vector of numbers, known in this context as parameters. The facial composite system described in Chapters 3 and 4 employs an *evolutionary algorithm* (EA) to determine the appearance model parameters from which a target face can be synthesized. In this chapter, an overview of evolutionary algorithms is provided, outlining the differences between four of the most widely used algorithms and the necessary details required for the implementation of an EA.

2.1 Introduction to principal components analysis (PCA)

In 1901, Karl Pearson gave a geometric account of the statistical technique that is known today as *principal components analysis*. Pearson's initial ideas [70] were developed further by Hotelling, who provided an explanation of the same technique in terms of a variance maximizing transformation. In his 1933 paper "Analysis of a Complex of Statistical Variables with Principal Components" [51], Hotelling describes the method of principal components and how it relates to factor analysis which was already established at this time. Although there are similarities between the two techniques he preferred to use the word 'component' instead of 'factor' (a term favoured by psychologists) to avoid confusion with the mathematical meaning of the word. It was at this time that the phrase 'principal components' was first introduced. Hotelling derived his principal components from population statistics, using Lagrange multipliers and differentiation to solve a variance maximizing, optimization problem. Principal components analysis is sometimes referred to as the *Hotelling transform.* The term *Karhunen-Loeve transform* [32] has also been used in the context of principal components, although some variations on the *Karhunen-Loeve expansion* method, presented in the pattern recognition literature, differ from PCA (see Webb [98]). A similar derivation to Hotelling's method is presented below.

2.1.1 Derivation of PCA for sample data

Let $\mathbf{x}$ be a random vector of m random variables $\{X_1, X_2, \ldots, X_m\}$ and $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ be n observations of $\mathbf{x}$. If the projection of the k^{th} observation onto the unit vector $\mathbf{u}$ is defined as $z_k = \mathbf{u}^T\mathbf{x}_k$, then the sample *variance* of all such projections for the n observations can be written as,

$$var\left[z\right] = \frac{1}{n-1}\sum_i \left(z_i - \bar{z}\right)^2 = \frac{1}{n-1}\sum_i \left(\mathbf{u}^T\mathbf{x}_i - \frac{1}{n}\sum_j \mathbf{u}^T\mathbf{x}_j\right)^2 \tag{2.1}$$

where $\bar{z} = \frac{1}{n}\sum_j z_j$. The second term in parentheses on the r.h.s of equation 2.1 expands and simplifies as follows,

$$\begin{aligned}\frac{1}{n}\sum_j \mathbf{u}^T\mathbf{x}_j &= \frac{1}{n}\left[\mathbf{u}^T\mathbf{x}_1 + \mathbf{u}^T\mathbf{x}_2 + \mathbf{u}^T\mathbf{x}_3 \ldots \mathbf{u}^T\mathbf{x}_n\right] \qquad (2.2)\\ &= \frac{1}{n}\mathbf{u}^T\left[\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3 \ldots \mathbf{x}_n\right] = \mathbf{u}^T\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i = \mathbf{u}^T\bar{\mathbf{x}}\end{aligned}$$

substituting $\mathbf{u}^T\bar{\mathbf{x}}$ for $\frac{1}{n}\sum_j \mathbf{u}^T\mathbf{x}_j$ into the r.h.s of 2.1 and simplifying gives,

$$\frac{1}{n-1}\sum_i \left(\mathbf{u}^T\left(\mathbf{x}_i - \bar{\mathbf{x}}\right)\right)^2 = \frac{1}{n-1}\sum_{i=1}^{n}\mathbf{u}^T\left(\mathbf{x}_i - \bar{\mathbf{x}}\right)\left(\mathbf{x}_i - \bar{\mathbf{x}}\right)^T\mathbf{u} \tag{2.3}$$

since $\mathbf{u}^T(\mathbf{x}_i - \bar{\mathbf{x}})$ is a scalar value equal to $(\mathbf{x}_i - \bar{\mathbf{x}})^T\mathbf{u}$. For convenience, the above can be expressed in matrix form. We begin by expanding the summation,

$$\begin{aligned}
&\frac{1}{n-1}\Big[\mathbf{u}^T(\mathbf{x}_1-\bar{\mathbf{x}})(\mathbf{x}_1-\bar{\mathbf{x}})^T\mathbf{u}+\mathbf{u}^T(\mathbf{x}_2-\bar{\mathbf{x}})(\mathbf{x}_2-\bar{\mathbf{x}})^T\mathbf{u}+\dots \\
&\qquad +\mathbf{u}^T(\mathbf{x}_1-\bar{\mathbf{x}})(\mathbf{x}_n-\bar{\mathbf{x}})^T\mathbf{u}\Big] \\
&=\frac{1}{n-1}\mathbf{u}^T\Big[(\mathbf{x}_1-\bar{\mathbf{x}})(\mathbf{x}_1-\bar{\mathbf{x}})^T+(\mathbf{x}_2-\bar{\mathbf{x}})(\mathbf{x}_2-\bar{\mathbf{x}})^T+\dots \\
&\qquad +(\mathbf{x}_1-\bar{\mathbf{x}})(\mathbf{x}_n-\bar{\mathbf{x}})^T\Big]\mathbf{u} \\
&=\frac{1}{n-1}\mathbf{u}^T\begin{bmatrix}\uparrow & & \uparrow\\ (\mathbf{x}_1-\bar{\mathbf{x}}) & \dots & (\mathbf{x}_n-\bar{\mathbf{x}})\\ \downarrow & & \downarrow\end{bmatrix}\begin{bmatrix}\leftarrow & (\mathbf{x}_1-\bar{\mathbf{x}})^T & \rightarrow\\ & \vdots & \\ \leftarrow & (\mathbf{x}_n-\bar{\mathbf{x}})^T & \rightarrow\end{bmatrix}\mathbf{u} \\
&=\frac{1}{n-1}\mathbf{u}^T\mathbf{X}\mathbf{X}^T\mathbf{u}=\mathbf{u}^T\mathbf{S}\mathbf{u}\ \ \textit{with}\ \ \mathbf{S}=\frac{1}{n-1}\mathbf{X}\mathbf{X}^T
\end{aligned} \tag{2.4}$$

Thus in the equation above the columns of the matrix $\mathbf{X}$ are the n observation vectors, in mean deviation form $(\mathbf{x}_i - \bar{\mathbf{x}})$. The aim here is to find the unit vector[1] $\mathbf{u}$ that *maximizes the quadratic form* $\mathbf{u}^T\mathbf{S}\mathbf{u}$, subject to the constraint that $\mathbf{u}^T\mathbf{u} = 1$. This is equivalent to determining the vector $\mathbf{u}$ that maximizes $var[z]$. If these conditions are met, then $\mathbf{u}$ is referred to as the first *principal component* of the observation matrix $\mathbf{X}$ and is denoted by $\mathbf{u}_1$. The standard approach to solving an optimization problem of this kind is to use Lagrange's method of undetermined multipliers. The cost function is defined as,

$$Q = \mathbf{u}_1^T\mathbf{S}\mathbf{u}_1 - \lambda_1\left(\mathbf{u}_1^T\mathbf{u}_1 - 1\right) \tag{2.5}$$

Differentiating Q w.r.t. $\mathbf{u}_1$ gives,

$$\mathbf{S}\mathbf{u}_1 - \lambda_1\mathbf{u}_1 = (\mathbf{S} - \lambda_1\mathbf{I}_m)\,\mathbf{u}_1 = 0 \tag{2.6}$$

Multiplying equation 2.6 by $\mathbf{u}_1^T$ and applying the constraint $\mathbf{u}_1^T\mathbf{u}_1 = 1$,

$$\mathbf{u}_1^T\mathbf{S}\mathbf{u}_1 = \mathbf{u}_1^T\lambda_1\mathbf{u}_1 = \lambda_1\mathbf{u}_1^T\mathbf{u}_1 = \lambda_1 \tag{2.7}$$

Equation 2.6 represents an eigenvalue problem. Since the aim is to seek

[1] choosing $\mathbf{u}$ to be a unit vector simplifies the analysis but it is not an essential requirement for calculating the PCs. The only requirement is that $\mathbf{u}_1$ not be the null vector.

the vector $\mathbf{u}_1$ that maximizes the variance of $\mathbf{u}_1^T\mathbf{S}\mathbf{u}_1$, λ_1 must be the largest eigenvalue of $\mathbf{S}$ and $\mathbf{u}_1$ the corresponding eigenvector[2] The second principal component, $\mathbf{u}_2$, is derived in a similar way with the additional constraint that $\mathbf{u}_2^T\mathbf{u}_1 = 0$ which means that the first and second components are *orthogonal* and statistically uncorrelated.

Hence a new Lagrange cost function can be formed as follows,

$$Q = \mathbf{u}_2^T\mathbf{S}\mathbf{u}_2 - \lambda_2\left(\mathbf{u}_2^T\mathbf{u}_2 - 1\right) - \phi\mathbf{u}_2^T\mathbf{u}_1 \tag{2.8}$$

Differentiating Q w.r.t $\mathbf{u}_2$ gives

$$2\mathbf{S}\mathbf{u}_2 - 2\lambda_2\mathbf{u}_2 - \phi\mathbf{u}_1 = 0 \tag{2.9}$$

multiplying this equation by $\mathbf{u}_1^T$

$$\mathbf{u}_1^T\mathbf{S}\mathbf{u}_2 - \lambda_2\mathbf{u}_1^T\mathbf{u}_2 - \frac{\phi}{2}\mathbf{u}_1^T\mathbf{u}_1 = 0 \tag{2.10}$$

also, using equation 2.7 and the fact that $\mathbf{S}$ is a symmetric matrix requires that,

$$\mathbf{u}_1^T\mathbf{S}\mathbf{u}_2 = \left[\mathbf{S}\mathbf{u}_1\right]^T\mathbf{u}_2 = \left[\lambda_1\mathbf{u}_1\right]^T\mathbf{u}_2 = \lambda_1\mathbf{u}_1^T\mathbf{u}_2 = 0 \tag{2.11}$$

Therefore ϕ must be equal to zero since $\mathbf{u}_1^T\mathbf{u}_1 = 1$ and both $\mathbf{u}_1^T\mathbf{S}\mathbf{u}_2$ and $\mathbf{u}_1^T\mathbf{u}_2$ are equal to zero. Hence equation 2.10 reduces to $(\mathbf{S} - \lambda_2\mathbf{I}_p)\,\mathbf{u}_2 = 0$ with λ_2 being the second largest eigenvalue of $\mathbf{S}$, and $\mathbf{u}_2$ the second principal component of the dataset $\{\mathbf{x}_k\}$. In general, up to n principal components can be obtained by iterating this process with the condition that $\mathbf{u}_i^T\mathbf{u}_j = \delta_{ij}$, where δ_{ij} is kronecker's delta.

2.1.2 The singular value decomposition (SVD) and PCA

Although the calculation of principal components theoretically reduces to the solution of a standard eigenvector problem, the *singular value decomposition* (SVD) has become the mathematical tool of choice for principal components in

[2] It is worth noting that some texts (Jolliffe for instance). refer to the k^{th} principal component as the derived variable $\mathbf{u}_k^T\mathbf{x}$, and identify the elements of $\mathbf{u}_k$ as the loadings or coefficients. Here, as in [61] the eigenvector $\mathbf{u}_k$ will be named as the k^{th} principal component.

practical applications. Let $\mathbf{X}$ be a $p \times n$ matrix of observations in mean deviation form as described in section 2.1. The SVD of $\mathbf{X}$ can always be written as a decomposition of the form,

$$\mathbf{X} = \mathbf{U}\Lambda^{\frac{1}{2}}\mathbf{V}^T \tag{2.12}$$

The columns $\{\mathbf{u}_i\}$ of the matrix $\mathbf{U}$ form an orthonormal basis[3] for the column space of $\mathbf{X}$, whereas the orthonormal columns of $\mathbf{V}$ span the row space of $\mathbf{X}$. The vectors $\{\mathbf{u}_i\}$ are known as the *left singular vectors* of $\mathbf{X}$. The non zero elements on the diagonal of matrix $\mathbf{\Lambda}^{\frac{1}{2}}$ contain the corresponding singular values arranged in order of decreasing magnitude from top left to bottom right.

As shown in the previous section, in principal components analysis the objective is to find the eigenvectors and eigenvalues of the covariance matrix $\mathbf{S} = \frac{1}{n-1}\mathbf{X}\mathbf{X}^T$. Multiplying equation 2.12 by $\mathbf{X}^T$ from the right, and using the fact that $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices,

$$\mathbf{X}\mathbf{X}^T = \left(\mathbf{U}\Lambda^{\frac{1}{2}}\mathbf{V}^T\right)\left(\mathbf{U}\Lambda^{\frac{1}{2}}\mathbf{V}^T\right)^T = \left(\mathbf{U}\Lambda^{\frac{1}{2}}\mathbf{V}^T\right)\left(\mathbf{V}\Lambda^{\frac{1}{2}}\mathbf{U}^T\right) = \mathbf{U}\Lambda\mathbf{U}^T \tag{2.13}$$

Thus, calculation of the eigenvectors, $\{\mathbf{u}_i\}$,of $\mathbf{S}$ through equation 2.13 yields the principal components as the columns of the orthogonal matrix $\mathbf{U}$ and the corresponding eigenvalues on the diagonal of matrix Λ,

$$\Lambda = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_m \end{bmatrix} \tag{2.14}$$

In certain cases, however, $\mathbf{X}$ may contain many more rows than columns ($p >> n$). This is often the case when the observations are digital images and p is typically $40K$ or more. Calculation of the decomposition of $\mathbf{S}$ then becomes a prohibitive computational task. Fortunately the complexity of this computation can be reduced by performing a SVD on $\frac{1}{n-1}\mathbf{X}^T\mathbf{X}$ instead of decomposing $\frac{1}{n-1}\mathbf{X}\mathbf{X}^T$. By the same reasoning as above, multiplying equation 2.12 from the left by $\mathbf{X}^T$ gives,

[3] strictly speaking, a basis (whether for row or column space) is constructed from the first r columns corresponding to the number of non-zero singular values

$$\mathbf{X}^T\mathbf{X} = \left(\mathbf{U}\Lambda^{\frac{1}{2}}\mathbf{V}^T\right)^T \left(\mathbf{U}\Lambda^{\frac{1}{2}}\mathbf{V}^T\right) = \left(\mathbf{V}\Lambda^{\frac{1}{2}}\mathbf{U}^T\right) \left(\mathbf{U}\Lambda^{\frac{1}{2}}\mathbf{V}^T\right) = \mathbf{V}\Lambda\mathbf{V}^T \quad (2.15)$$

From equation 2.12 it can be seen that the orthonormal principal components, $\{\mathbf{u}_k\}$, that form the columns of the $p \times p$ matrix, $\mathbf{U}$, can then be obtained from the dimensionally smaller, $n \times n$ matrix $\mathbf{V}$ by rearranging equation 2.12 as follows,

$$\mathbf{U} = \mathbf{X}\mathbf{V}\Lambda^{-\frac{1}{2}} \quad (2.16)$$

Hence, when there are more variables than observations the principal components are determined by performing an SVD on $\mathbf{X}^T\mathbf{X}$ *to obtain* $\mathbf{V}$ *and then using equation 2.16.*

Care must be taken when inverting $\Lambda^{\frac{1}{2}}$ since some of the singular values are likely to approach zero which can lead to problems when this matrix is inverted. The way to avoid this problem is to identify the elements of $\Lambda^{\frac{1}{2}}$ that are vanishingly small and set the corresponding elements in $\Lambda^{-\frac{1}{2}}$ equal to zero. Alternatively, the dimensions of the matrices can be reduced with similar effect (see section 2.1.3).

PCA can be thought of as a procedure that rotates the coordinate frame in which the original data points are plotted (illustrated in figure 2.1). If the vector $\mathbf{x}_k$ contains the coordinates of the k^{th} data point in the original coordinate fame, then the let the vector $\mathbf{z}_k$ define the same point in the rotated frame in terms of a set of newly defined variables $\{Z_i\}$. In the rotated frame of reference, the spread of data points along the direction of the new variables is maximal.

The vectors, $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n$, can be obtained easily by multiplying equation 2.12 from the left by $\mathbf{U}^T$ (see equation 2.18).

$$\mathbf{Z} = \mathbf{U}^T\mathbf{X} = \Lambda^{\frac{1}{2}}\mathbf{V}^T \quad (2.17)$$

$$\mathbf{Z} = \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ \mathbf{z}_1 & \mathbf{z}_2 & \ldots & \mathbf{z}_n \\ \downarrow & \downarrow & & \downarrow \end{bmatrix}$$

The matrix product $\mathbf{U}^T\mathbf{X}$ represents the projection of the original data set (in mean deviation form) onto the orthonormal principal components.

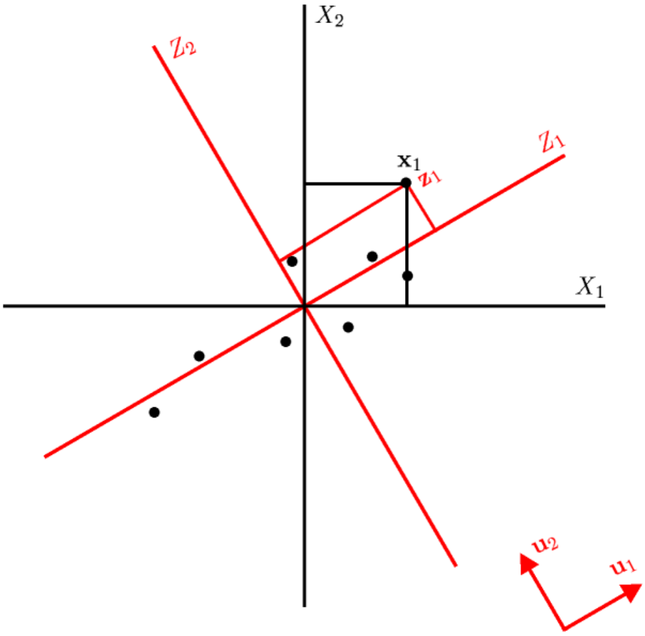


Figure 2.1: A geometrical interpretation of PCA. The principal components $\{\mathbf{u}_i\}$ define directions along which the spread of original data points is maximized. $\{X_i\}$ are the original variables and $\{Z_i\}$ is a new set of variables as defined by the principal components. If $\mathbf{x}_1$ represents a data point in terms of the original variables, then the same point is represented by a new vector $\mathbf{z}_1$ in terms of the new variables.

2.1.3 Compact data encoding using PCA

The PCA technique is particularly useful when attempting to construct a compact model of an pattern class (e.g. faces), in which the objects are represented by high dimensional, highly correlated, feature vectors. The primary aim of PCA is to achieve a reduction in the dimensionality of the data. If the vectors $\{\mathbf{x}_i\}$ are, to some extent, correlated with each other then some of the principal components will make a negligible contribution to the model and can be discarded. Formal methods exist for determining how many PCs to retain. However a rule of thumb that works well in most cases is to retain components that together describe a chosen percentage, T, of the variation present in the original data set, say 80% (see figure 2.3). Conveniently, the SVD returns the principal components in order of decreasing significance, allowing a threshold T to be set on the cumulative variance. The objective is to replace the total number of components, m, with a much smaller subset of t components.

$$T = 100 \frac{\sum_{k=1}^{t} \lambda_k}{\sum_{j=1}^{m} \lambda_j} \tag{2.18}$$

The original observation vectors can be reconstructed as follows,

$$\mathbf{x} = \bar{\mathbf{x}} + \sum_{i=1}^{m} \mathbf{u}_i z_i \tag{2.19}$$

where z_i determines the influence of the i^{th} principal component on the reconstructed observation. If all m principal components are used, equation 2.19 gives a perfect reconstruction of the original observation vector. However, if $t < m$ principal components are used, an approximate reconstruction is obtained as illustrated in figure 2.2 and described by the following equation,

$$\hat{\mathbf{x}} = \bar{\mathbf{x}} + \sum_{i=1}^{t} \mathbf{u}_i z_i \tag{2.20}$$

In practice there are often many more variables than observations (typically the case for image data). In such cases, the observation vectors lie in a n-dimensional subspace of $\mathbf{R}^m$ and the last $m - n$ components will not contribute to the reconstruction. Instances in which $m > n$ demand that $t \leq n$.

Subsequent sections in this chapter illustrate how the statistical method of

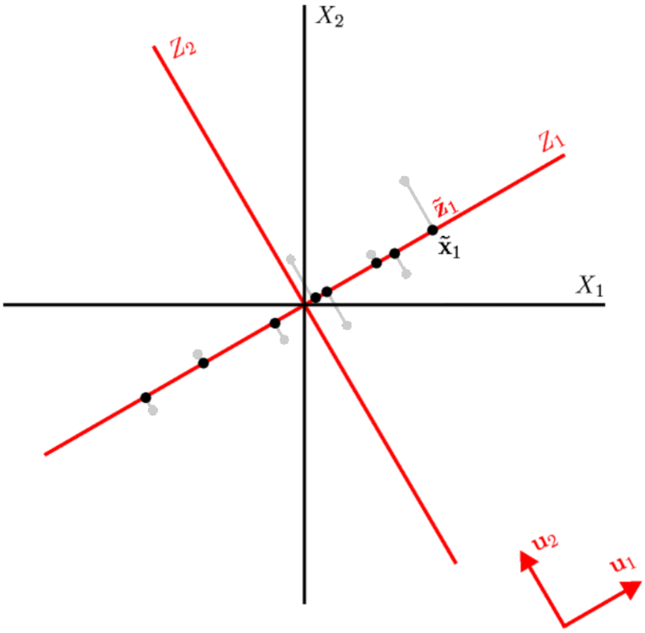


Figure 2.2: In the simple 2D example presented here, the original data (grey markers) has been approximated by a single new variable, Z_1 as defined by the first principal component $\mathbf{u}_1$. For each observation, the error due to approximation is given by the perpendicular distance to the Z_1 axis.

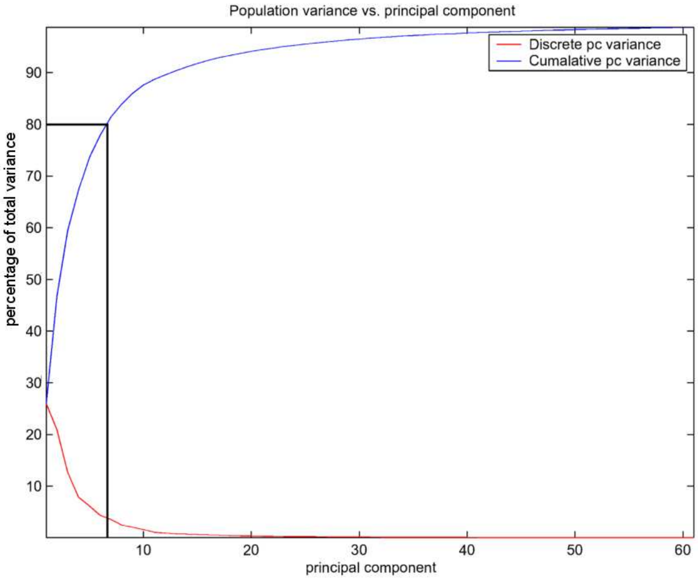


Figure 2.3: A plot of the variances associated with a typical principal components analysis. The cumulative variance plot indicates that 80% of the information contained in the original data set can be represented by the first seven PCs alone.

principal components analysis can be used to form compact representations of image data.

2.2 Modelling shape

Consider n objects from the same pattern class, contained in one or more digital images. Two properties that characterize objects are their shape and texture. In this section, a method for extracting the shape property from digital images, and a method for modelling shape variations of a specific pattern class, are described. If a sufficient number of representative sample objects are available, it is possible construct a model from which an approximation to any object from the population of all such objects can be synthesised.

The term shape is usually used to refer to an arrangement of points or lines that define the perimeter of an object. In this thesis, the word 'shape' will refer to a property of a configuration of points that is independent of scaling, rotation and translation. When a comparison is made between two or more shapes it will therefore be assumed that they have been subjected to a suitable alignment procedure that places them in the same frame of reference. To avoid confusion the term point set will be used in this section when referring to an unaligned shape.

The autumn leaves shown in figure 2.4 will be used as an example of objects belonging to the same pattern class. This pattern class will be used to illustrate how compact representations of shape, texture and appearance can be obtained using *principal components analysis*. Also, new examples of objects will be synthesized from a learned statistical model of this class of objects.

2.2.1 Landmarking: Obtaining shape from images

A landmark is a coordinate pair (or tuple for a 3d data set) which defines a specific position on an object or an image of an object. Collectively, a set of landmark points may be used to define the object's shape. When sets of landmarks are placed on two or more objects a *correspondence* is sought such that the order of labelling remains the same in each case. For instance, in the specific example provided in this section a suitable choice for the first landmark is the point at which the stem joins the leaf. Hence the first landmark would be placed at this point for each and every leaf in the sample. For each object the corresponding landmark data is stored in a $2m$ element vector as,

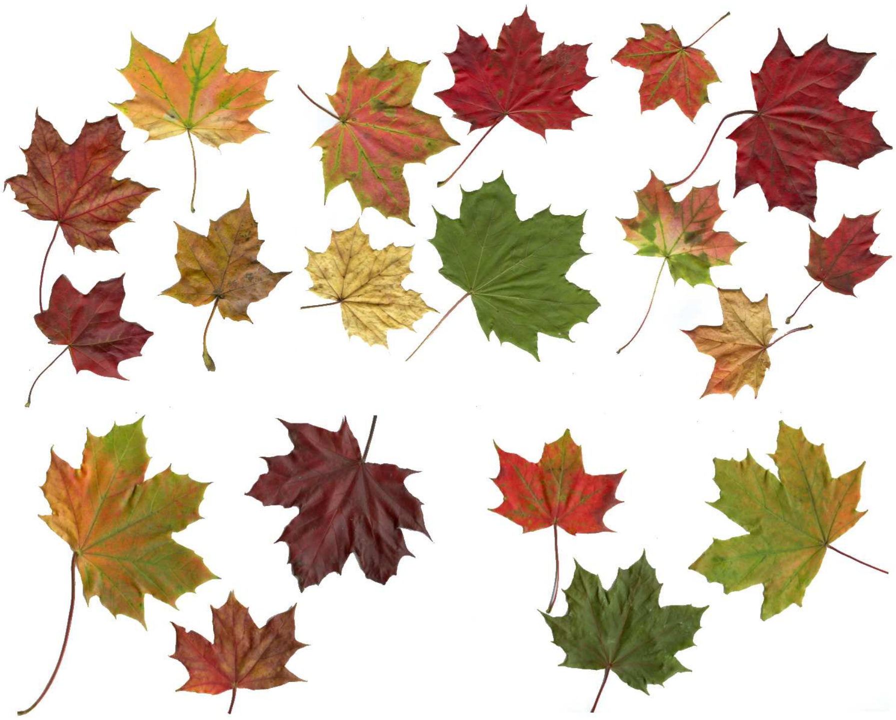


Figure 2.4: A sample of autumn leaves from a single species of tree. The leaves belong to a distinct pattern class. The intra-class variation in shape and colouration will used as an illustrative example of how to construct an appearance model.

$$\mathbf{x} = [x_1 \; x_2 \; \ldots \; x_m \; y_1 \; y_2 \; \ldots \; y_m]^T \tag{2.21}$$

Stegmann [85] (see also Dryden [28]) classifies landmarks into one of three categories according to the method by which they are placed on the object:

1. An *anatomical landmark* is a point assigned by an expert that corresponds between organisms in some biologically meaningful way. A suitable candidate for an anatomical landmark is a highly salient point that can be reliably located on all of the sample objects. These tend to occur on edges of the object, especially where there is a local maxima in the curvature of object's surface. Regions of distinctive colouration may also be candidates for anatomical landmarks.

2. *Mathematical landmarks* are points located on an object according to some mathematical or geometrical property. Various algorithms have previously been described for calculating positions of mathematical landmarks [57, 22].

3. *Pseudo-landmarks* are constructed points on an organism, located either around the outline or in between anatomical or mathematical landmarks. The positioning of such landmarks is dictated by the locations of the anatomical or mathematical landmarks.

In the autumn leaf example, 23 landmark points were used to represent the shape of each sample leaf, as illustrated in figure 2.5. The chosen positions of landmark points delineate the perimeter of the leaf and its basic vein structure.

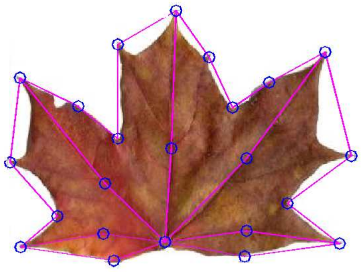


Figure 2.5: Landmark points used to delineate leaf-shape (blue circular markers). The magenta line segments are for the purpose of illustration, and as such do not contribute to the shape model.

2.2.2 Point set alignment

Translation, scaling and rotational effects are rarely of interest in morphological studies of an object class. Hence it is common practice to remove these unwanted factors by aligning point sets prior to constructing a shape model. One commonly used approach is to seek the shape transformations that optimally superimpose one shape upon another.

The Procrustes distance P_d^2 is a shape metric, providing a quantatitive measure of difference between two optimally aligned point sets $\mathbf{x}_1$ and $\mathbf{x}_2$. P_d^2 can be considered to be the square root of the sum of squared differences between the positions of the landmarks in $\mathbf{x}_1$ and $\mathbf{x}_2$.

$$P_d^2 = \sum_{j=1}^{m} \left[(x_{1j} - x_{2j})^2 + (y_{1j} - y_{2j})^2 \right] \tag{2.22}$$

The alignment procedure, or Procrustes superposition as it is known in this context, is achieved by applying similarity transformations to one point set to align it with the reference point set. The alignment procedure can be summarized as follows,

1. Compute the centroid of each point set.
2. Translate both point sets to the x, y origin by subtracting their respective centroids.
3. Re-scale each point set to have equal size.
4. Rotate one point set to align with the other.

The centroid of a point set is a two element vector containing the mean x-y values of the landmark positions,

$$[\bar{x}, \bar{y}] = \left[\frac{1}{m} \sum_{j=1}^{m} x_j, \frac{1}{m} \sum_{j=1}^{m} y_i \right] \tag{2.23}$$

In order to scale both shapes to a common size, a size metric $S(\mathbf{x})$ such as the Frobenius norm is required,

$$S(\mathbf{x}) = \sqrt{\sum_{j=1}^{m} \left[(x_j - \bar{x})^2 + (y_i - \bar{y})^2 \right]} \tag{2.24}$$

SVD can be used to determine the rotation matrix for aligning the point sets (see Bookstein [7]),

$$\mathbf{U}\Lambda\mathbf{V}^T = \mathbf{X}_1^T\mathbf{X}_2 \tag{2.25}$$

where

$$\mathbf{X}_1 = \begin{bmatrix} x_{1,1} & y_{1,1} \\ x_{1,2} & y_{1,2} \\ \vdots & \vdots \\ x_{1,n} & y_{1,n} \end{bmatrix}, \quad \mathbf{X}_2 = \begin{bmatrix} x_{2,1} & y_{2,1} \\ x_{2,2} & y_{2,2} \\ \vdots & \vdots \\ x_{2,n} & y_{2,n} \end{bmatrix}$$

The matrix **R** that rotates the first point set to the reference point set is equal to the product of the matrix containing the orthonormal *right singular vectors* **V** and the transpose of the matrix containing the *left singular vectors* **U**.

$$\mathbf{R} = \mathbf{V}\mathbf{U}^T \tag{2.26}$$

An alternative method [21] is to translate both point sets to the same position, scale them to have equal size and to determine a and b that minimize E as follows,

$$E = |\mathbf{X}_1\mathbf{R} - \mathbf{X}_2|\,, \quad \mathbf{R} = \begin{bmatrix} a & -b \\ b & a \end{bmatrix} \tag{2.27}$$

The key steps in the Procrustes alignment procedure are illustrated in figure 2.6.

An iterative procedure can be followed that determines the mean of the n sample shapes and aligns all training samples in the process.

1. Choose any point set as the first estimate of the mean shape.
2. Align all the remaining point sets to the mean shape using Procrustes alignment.
3. Re-calculate the estimate of the mean from the aligned shapes
4. If the mean estimate has changed return to step 2.

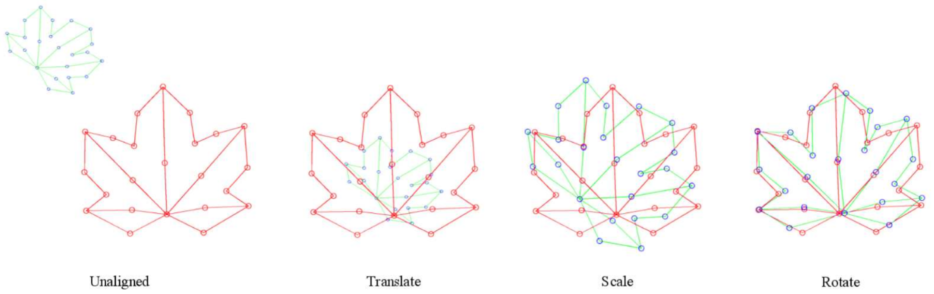


Figure 2.6: Procrustes alignment procedure. In this example, the shape of a sample leaf (blue and green) is aligned to a reference shape (red)

The mean shape is calculated using the *Procrustes mean* equation 2.28 which has the smallest summed squared Procrustes distance to all the configurations of a sample.

$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i \tag{2.28}$$

The process has converged when no further changes to the mean occur. This is often achieved in as few as two iterations. A *Procrustes scatter* formed by the alignment of all sample shapes to the Procrustes mean is shown in figure 2.7.

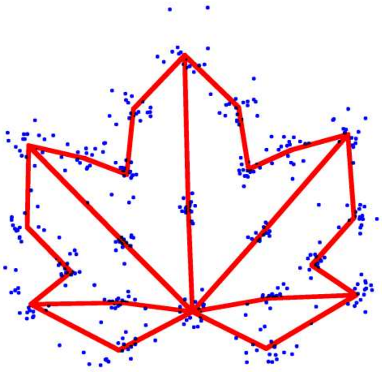


Figure 2.7: A Procrustes scatter depicting the sample leaf shapes (blue points), aligned to the Procrustes mean shape (red line).

2.2.3 Compact Shape Representation

Once the n sample shapes have been brought into the same frame of reference, using the alignment procedure described in the previous section, a compact representation can be obtained using PCA. The computational procedure for performing a PCA on the shape data is described by the following steps.

1. Subtract the mean shape from each of the n sample shapes. Let the i^{th} sample shape in mean deviation form be denoted by $d\mathbf{x}_i$

$$d\mathbf{x}_i = \mathbf{x}_i - \bar{\mathbf{x}} \tag{2.29}$$

2. Insert the sample shapes in mean deviation form into the columns of the observation matrix $\mathbf{X}$.

$$\mathbf{X} = \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ d\mathbf{x}_1 & d\mathbf{x}_2 & \dots & d\mathbf{x}_n \\ \downarrow & \downarrow & & \downarrow \end{bmatrix} \tag{2.30}$$

3. Form the $2m \times 2m$ sample covariance matrix $\mathbf{S}_s$ that describes the positional relationships between landmarks,

$$\mathbf{S}_s = \frac{1}{n-1} \sum_{i=1}^{n} d\mathbf{x}_i d\mathbf{x}_i^T = \frac{1}{n-1} \mathbf{X}\mathbf{X}^T \tag{2.31}$$

   where n is the number of sample objects.

4. If $n > 2m$ determine the eigenvectors and eigenvalues of $\mathbf{S}_s$, thereby obtaining the shape principal components directly.

$$\mathbf{S}_s \mathbf{p}_s^i = \lambda_s^i \mathbf{p}_s^i \tag{2.32}$$

   where the superscript i signifies the i^{th} principal component and $\left\{\mathbf{p}_s^i\right\}$ are orthonormal satisfying,

$$\left(\mathbf{p}_s^j\right)^T \mathbf{p}_s^k = \delta_{jk} \tag{2.33}$$

   (here the more intuitive symbol $\mathbf{p}$ is used to denote a principal component, whereas in the previous section in PCs were derived, $\mathbf{u}$ was used). For all $\mathbf{p}_s^i$, equation 2.32 can be written in matrix form as,

$$\mathbf{S}_s\mathbf{P}_s = \mathbf{P}_s\Lambda_s \tag{2.34}$$

with

$$\mathbf{P}_s = \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ \mathbf{p}_s^1 & \mathbf{p}_s^2 & \dots & \mathbf{p}_s^{2m} \\ \downarrow & \downarrow & & \downarrow \end{bmatrix}$$

Thus $\mathbf{P}_s$ is the matrix that diagonalizes the shape sample covariance matrix.

5. For the autumn leaves example, 23 landmarks were used ($m = 23$) to delineate the shapes of 19 leaves ($n = 19$). Hence $n < 2m$ and the shape covariance matrix given by equation 2.50 is *positive semi-definite* which means that it cannot be diagonalized (see Strang [87]). In this case the n eigenvectors $\left\{\mathbf{v}_s^i\right\}$ of the *positive definite* matrix $\frac{1}{n-1}\mathbf{X}^T\mathbf{X}$ are determined, and the principal components obtained by,

$$\mathbf{p}_s^i = \mathbf{X}\mathbf{v}_s^i \left(\lambda_s^i\right)^{-\frac{1}{2}} \tag{2.35}$$

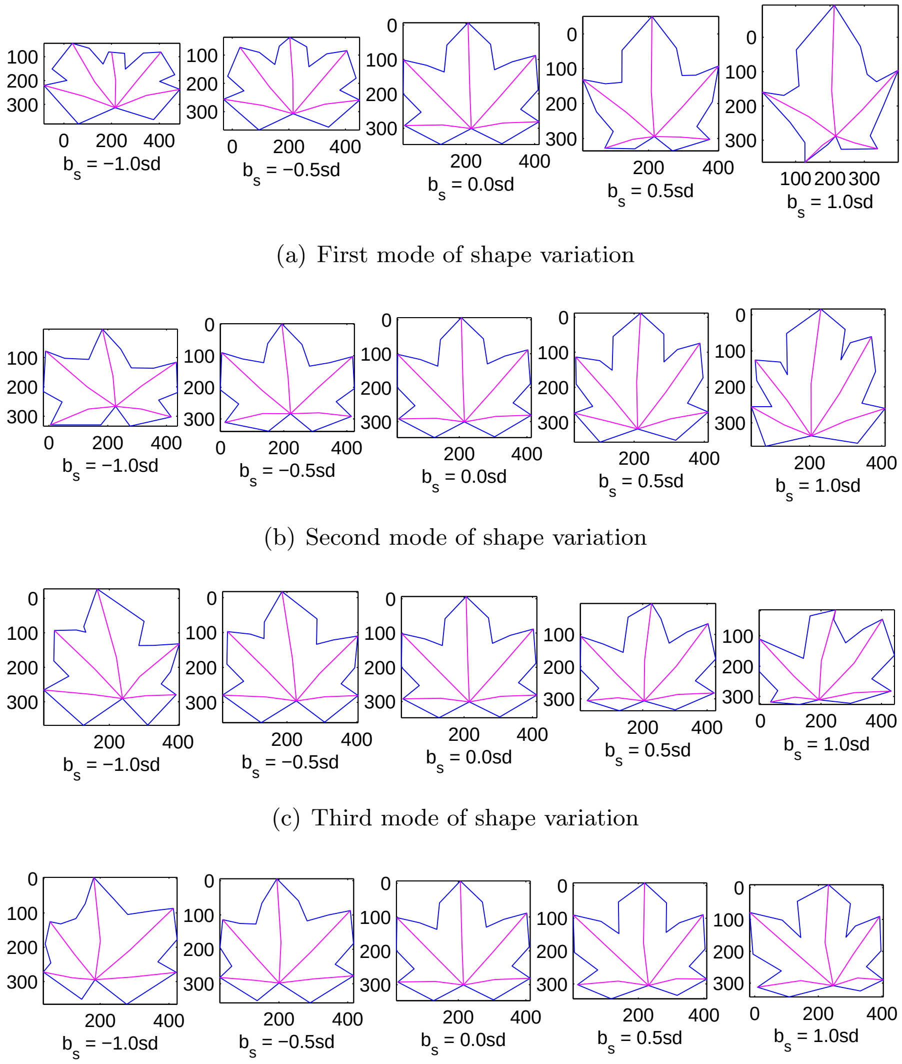
(see equations 2.15 and 2.16 from the previous section). In matrix form this is expressed as,

$$\mathbf{P}_s = \mathbf{X}\mathbf{V}_s\Lambda_s^{-\frac{1}{2}} \tag{2.36}$$

Each shape principal component records a unique and global shape deformation from the mean shape. Cootes et al [21] refer to these deformations as *modes of variation.* Visualizing these modes is helpful for interpreting the main ways in which the shape of objects from a specific class vary. The i^{th} mode can be visualized by adding a proportion of $\mathbf{p}_s^i$ to the mean shape in increments of 0.5σ to achieve deformations $\mathbf{x}_{mi}$ from the mean shape.

$$\mathbf{x}_{mi} = \bar{\mathbf{x}} + \alpha\mathbf{p}_s^i \tag{2.37}$$

In figure 2.8, the first three modes of variation ($\mathbf{x}_{m1}, \mathbf{x}_{m2}, \mathbf{x}_{m3}$) corresponding to the first, second and third principal components are displayed, where α lies in the range $-1\sigma \leq \alpha \leq +1\sigma$. It is assumed that the principal components have been arranged in order of decreasing variance such that $\lambda_s^1 > \lambda_s^2 > \lambda_s^3$, as is normal.



(a) First mode of shape variation

(b) Second mode of shape variation

(c) Third mode of shape variation

(d) Fourth mode of shape variation

Figure 2.8: First four modes of shape variation. The first two modes indicate variation in the basic form, whereas the third and fourth modes predominantly display left/right asymmetry.

Projecting one of the original sample shapes $d\mathbf{x}$ (in mean deviation form) onto each principal component, results in a non-lossy encoding, $\mathbf{b}_s$, of that shape.

$$\mathbf{b}_s = \sum_{j=1}^{2m} \left(\mathbf{p}_s^j\right)^T d\mathbf{x} \quad \textit{or} \quad \mathbf{b}_s^i = \mathbf{P}_s^T d\mathbf{x} \tag{2.38}$$

Since the principal components are ordered by significance, an approximate and compact representation of $\mathbf{x}$ can be obtained by using the first t_s components only,

$$\hat{\mathbf{b}}_s = \sum_{j=1}^{t_s} \left(\mathbf{p}_s^j\right)^T (\mathbf{x} - \bar{\mathbf{x}}) \tag{2.39}$$

where $\hat{\mathbf{b}}_s$ is a t_s element vector and typically $t_s << 2m$. An approximation, $\hat{\mathbf{x}}$, to the original sample shape can be reconstructed from $\hat{\mathbf{b}}_s$ as follows,

$$\hat{\mathbf{x}} = \sum_{j=1}^{t_s} \mathbf{p}_s^j \hat{b}_s^j + \bar{\mathbf{x}} \tag{2.40}$$

In the leaves example, 80% of the shape variance over the original sample is modelled by the first five principal components (figure 2.9). Therefore a reasonable value for t_s would be $t_s = 5$, allowing any leaf shape to be encoded by a vector of only five parameters.

2.3 Modelling texture

The previous section described how intra-class shape variation can be modelled using PCA. In this section, the intra-class texture characteristics are modelled. Specifically, the texture variation contained within the autumn leaves dataset. In the computer vision literature, the term *texture* is used when referring to a pattern of pixel intensity values. A model of texture variation for the chosen object class can be constructed by determining a set of principal components, using a method analogous to one outlined for the shape data. In order to model the texture independently of the shape, each sample image is first warped to the mean shape prior to constructing the model. Failure to perform this shape normalizing step will result in the presence of ghosting artefacts in the texture principal components.

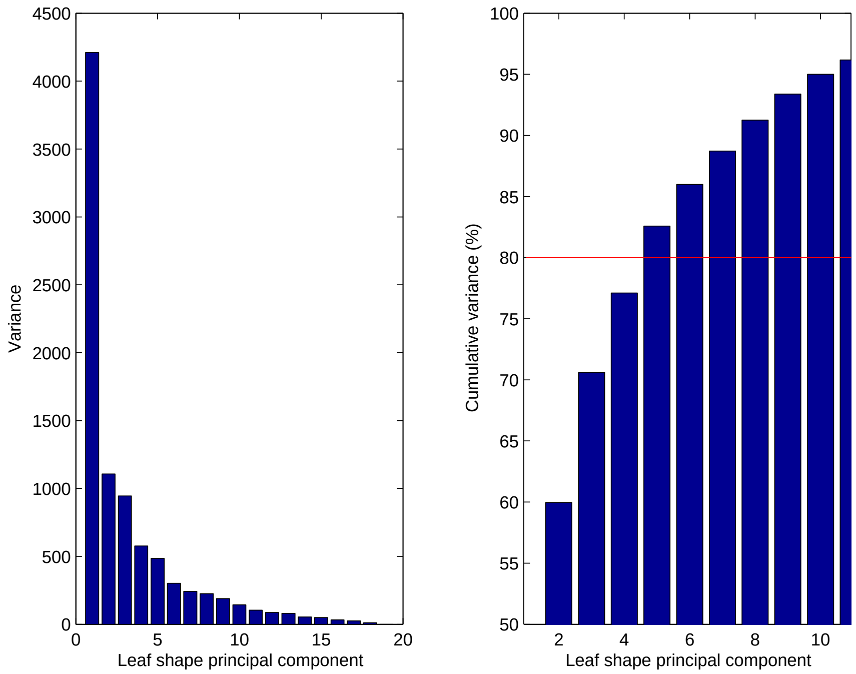


Figure 2.9: Variance associated with each *shape principal component* for the autumn leaves example (left). Cumulative variance (right) - 80% of the shape variation associated with the original dataset can be expressed by the first five principal components.

2.3.1 Obtaining pixel correspondences

When normalizing the image shape, an image warp is required that maps a set of control points, $\{x_i\}$, in image I to another set of control points, $\{x'_i\}$, in the output image, I'. There are various methods for effecting image warps some of which are more suited to certain applications than others. *Thin plate splines* [6] offer a smooth continuous warp but take a relatively long time to compute. For applications in which fast rendering is required a *piece-wise affine* warp is more appropriate.

Piece-wise affine

As the name suggests the *piece-wise* affine warp consists of individual local warps, F_1, $F_2, \ldots,$ F_{m-2}, that collectively produce a global geometric image transformation.

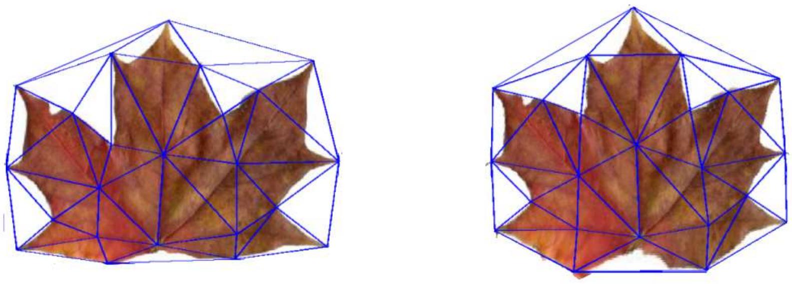


Figure 2.10: Original leaf image (left) and shape normalized leaf image (right). Corresponding Delaunay tessellations displayed on each image.

Local regions of the image can be defined by an irregular triangular tessellation in the image plane. Delaunay triangulation provides a method for the optimal partitioning of the convex hull of the control points into non-overlapping triangles. The Delaunay construction ensures that the circumcircle of each triangle does not contain any control points other than its own vertices. The warp is realized by constructing the corresponding tessellation in the output image according to the control points $\{\mathbf{x}'_i\}$ and then computing the transformation between corresponding triangles. If each shape consists of m landmark points, the number of triangles in the tessellation will be $m - 2$.

An affine transformation of R^2 is a map $F : R^2 \rightarrow R^2$ that preserves lines and parallelism. Unlike Euclidean transforms, the general affine transformation does not necessarily preserve length and angle. F can be expressed as a linear transformation, represented here by the matrix $\mathbf{A}$, and a translation, represented by the vector $\mathbf{t}$ as per equation 2.41

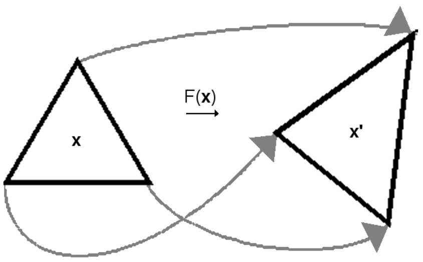


Figure 2.11: An affine transformation $F(\mathbf{x})$ maps the vertices $(\mathbf{x})$ of a triangle into the vertices $(\mathbf{x}')$ defining a new triangle. In practice, $F(\mathbf{x})$ is computed using the control points in $\mathbf{x}$ and $\mathbf{x}'$ and then applied to the pixel coordinates located inside the triangle.

$$F(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{t}, \;\; \mathbf{x} \in R^n \tag{2.41}$$

The translation component of F is not linear, and therefore can not be incorporated within $\mathbf{A}$. For practical purposes the form of F can be simplified using homogeneous coordinates in which a point in the set $\{\mathbf{x}_i\}$ is represented by the tuple $[x_i, y_i, 1]^T$ and the corresponding point in $\{\mathbf{x}'_i\}$ is represented by $[x'_i, y'_i, 1]^T$. Thus, equation 2.41 can be replaced by,

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.42}$$

The set of all possible affine transformations in homogeneous format represents a *group*, G and therefore satisfies the following axioms [2] expressed in matrix notation, where $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ are 3×3 matrices representing transformations.

1. *Closure*: $\mathbf{AB} \in G$
2. *Associativity*: $(\mathbf{AB})\,\mathbf{C} = \mathbf{A}\,(\mathbf{BC})\,, \;\forall\; \mathbf{A}, \mathbf{B}, \mathbf{C} \in G$
3. *Identity*: There exists an identity matrix $\mathbf{I} \in G$, such that $\mathbf{IA} = \mathbf{AI} = \mathbf{A}, \;\forall\; \mathbf{A} \in G$
4. *Inverse*: Each matrix $\mathbf{A} \in G$ has an inverse matrix $\mathbf{A}^{-1} \in G$, such that $\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$

These axioms state that two or more affine transformations can be multiplied together to form a compound transformation that is also affine, and that any transformation that belongs to a group can be inverted. Special transformations within the affine group are translation (only in the homogeneous form), **T**, rotation, **R** and scaling, **S** (equation 2.43). In computer graphics applications it is often convenient to compose an affine transformation from these basic transformations.

$$\{\mathbf{T}\left(t_x, t_y\right), \mathbf{R}, \mathbf{S}\left(s_x, s_y\right) | s_x, s_y \neq 0\} \tag{2.43}$$

An affine transformation can be found that can transform a triangle into any other specified triangle. This follows logically from equation 2.42 by inserting two more position vectors in homogeneous form. In equation 2.45, the vertices $[x_1\ y_1\ 1]^T, [x_2\ y_2\ 1]^T, [x_3\ y_3\ 1]^T$ of the source triangle are mapped to the vertices $[x_1'\ y_1'\ 1]^T, [x_2'\ y_2'\ 1]^T, [x_3'\ y_3'\ 1]^T$ defining the destination triangle.

$$\begin{bmatrix} x_1' & x_2' & x_3' \\ y_1' & y_2' & y_3' \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{bmatrix} \tag{2.44}$$

Multiplying out the left hand side of equation 2.45 yields six equations in six unknowns. The coefficients $a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23}$ can be determined by matrix inversion as follows,

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x_1' & x_2' & x_3' \\ y_1' & y_2' & y_3' \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{bmatrix}^{-1} \tag{2.45}$$

The left hand side of equation 2.45 provides the local transformation matrix which can be applied simultaneously to the interior coordinates of the source triangle and its vertices, thus achieving the local warp. Repeating the process for all triangles within the complex hull of the shape produces the warped output image.

Pixel Interpolation

The pixel mapping operations associated with image warping are often thought of, and implemented as, mappings from locations in the source image to locations in the output image. The problem with this forward mapping is that some of the pixel values in the output image are likely to remain undefined due to the fact that a one-to-one mapping between source pixels and output pixels does not exist. This issue is compounded by rounding errors in the computation of the destination coordinates. The combined effect results in "holes" in the output image where some pixels are not assigned a value. For this reason, it is often more convenient to compute the reverse-warp, whereby pixel coordinates are mapped from the output image into the source image. In matrix form, the reverse-warp can be expressed as the inverse of the matrix defined in equation 2.45. The reverse-warp method guarantees that the value of each and every pixel in the output image is defined. It does not however, guarantee that coordinates from the source are mapped to integer positions in the output image. In general, mapped coordinates will have non-integer values relating to a position in between pixels in the source image. In this case, an interpolation method is required to determine the value of the corresponding pixel in the output image. The simplest method is to round the mapped coordinates to integer values, a process known as *nearest neighbour* interpolation or *point sampling. Bilinear* interpolation and *cubic* interpolation offer more accurate results but take longer to compute than the nearest neighbour method. Once a corresponding pixel (or a weighted sum of pixels in the bilinear and cubic case) has been established it's value is sampled and assigned to its correct position in the output image.

Compact Texture Representation

Once the warping procedure has been applied to each of the n training images, the texture vectors $\{\mathbf{g}_i\}$ can be formed. For each shape normalized image, the pixel values are extracted in a column-wise fashion from the complex hull of the mean object shape as illustrated in figure 2.12. For RGB images, this process is repeated three times for each of the three colour planes and concatenated to produce the $3m$ element texture vector, where m is the number of pixels represented[4].

$$\mathbf{g} = [R_1 \; R_2 \; \ldots \; R_m \; G_1 \; G_2 \; \ldots \; G_m \; B_1 \; B_2 \; \ldots \; B_m] \tag{2.46}$$

Intensity values are often normalized at this stage to reduce the effects of varying illumination conditions. Covariances in the sample textures can be captured using a PCA model. The equations defining the texture follow the

[4] Note that here m is used to denote the number of pixels, whereas in the previous section it represented the number of landmarks. In both cases it is related to the number of variables

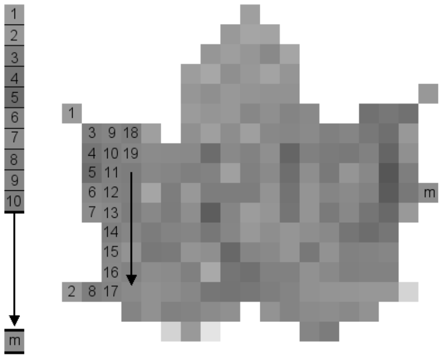


Figure 2.12: Pixels are extracted column-wise from the shape normalized image, thereby forming an observation vector.

same form as those described in section 2.2.3 with suitably amended notation.

1. Subtract the mean texture, $\bar{\mathbf{g}}$, from each of the n sample textures,

$$\bar{\mathbf{g}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{g}_i \tag{2.47}$$

   and let the i^{th} sample texture in standard deviation form be denoted by $d\mathbf{g}_i$

$$d\mathbf{g}_i = \mathbf{g}_i - \bar{\mathbf{g}} \tag{2.48}$$

2. Insert the sample textures in mean deviation form into the columns of matrix $\mathbf{G}$.

$$\mathbf{G} = \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ d\mathbf{g}_1 & d\mathbf{g}_2 & \dots & d\mathbf{g}_n \\ \downarrow & \downarrow & & \downarrow \end{bmatrix} \tag{2.49}$$

3. Digital images typically contain $10^4 < m < 10^7$ pixels, therefore, in most cases there are many more variables (RGB triplets) than observations (sample textures). Consequently, there will be at most n principal components. The $3m \times 3m$ covariance matrix $\mathbf{S}_g = \frac{1}{n}\mathbf{G}\mathbf{G}^T$ that describes the intensity relationships between pixels is positive semi-definite and can

not be diagonalized. In such cases the positive definite matrix $\tilde{\mathbf{S}}_g$ is constructed instead of $\mathbf{S}_g$,

$$\tilde{\mathbf{S}}_g = \frac{1}{n-1}\mathbf{G}^T\mathbf{G} \tag{2.50}$$

4. Determine the eigenvectors $\left\{\mathbf{v}_g^i\right\}$ and eigenvalues $\left\{\lambda_g^i\right\}$ of $\tilde{\mathbf{S}}_g$,

$$\tilde{\mathbf{S}}_g\mathbf{v}_g^i = \lambda_g^i\mathbf{v}_g^i \quad where \quad \left(\mathbf{v}_g^j\right)^T\mathbf{v}_g^k = \delta_{jk} \tag{2.51}$$

For all $\mathbf{v}_g^i$, equation 2.51 can be written in matrix form as,

$$\tilde{\mathbf{S}}_g\mathbf{V}_g = \mathbf{V}_g\Lambda_g \tag{2.52}$$

with

$$\mathbf{V}_g = \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ \mathbf{v}_g^1 & \mathbf{v}_g^2 & \dots & \mathbf{v}_g^n \\ \downarrow & \downarrow & & \downarrow \end{bmatrix}$$

5. The SVD (see equation 2.16) provides a relationship between $\mathbf{v}_g^i$ and $\mathbf{p}_g^i$ that allows the orthonormal principal components to be recovered.

$$\mathbf{P}_g = \mathbf{G}\mathbf{V}_g\Lambda_g^{-\frac{1}{2}} \tag{2.53}$$

and

$$\mathbf{P}_g = \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ \mathbf{p}_g^1 & \mathbf{p}_g^2 & \dots & \mathbf{p}_g^n \\ \downarrow & \downarrow & & \downarrow \end{bmatrix} \tag{2.54}$$

Each texture principal component records a unique and global deviation from the mean texture. The i^{th} mode can be visualized by adding a proportion of $\mathbf{p}_g^i$ to the mean texture in increments of 0.5σ to achieve variations in colouring and shading $\mathbf{g}_{mi}$ with respect to the mean.

$$\mathbf{g}_{mi} = \bar{\mathbf{g}} + \alpha \mathbf{p}_g^i \tag{2.55}$$

In figure 2.13, the first three modes of variation $(\mathbf{g}_{m1}, \mathbf{g}_{m2}, \mathbf{g}_{m3})$ corresponding to the first second and third principal components are displayed, where α lies in the range $-1\sigma \leq \alpha \leq +1\sigma$. It is assumed that the principal components have been arranged in order of decreasing variance such that $\lambda_g^1 > \lambda_g^2 > \lambda_g^3$, as is normal.

Projecting one of the original sample textures onto each principal component results in a vector of parameters $\mathbf{b}_g$.

$$\mathbf{b}_g = \sum_{j=1}^{3m} \left(\mathbf{p}_g^j\right)^T d\mathbf{g} \quad or \quad \mathbf{b}_g = \mathbf{P}_g^T d\mathbf{g} \tag{2.56}$$

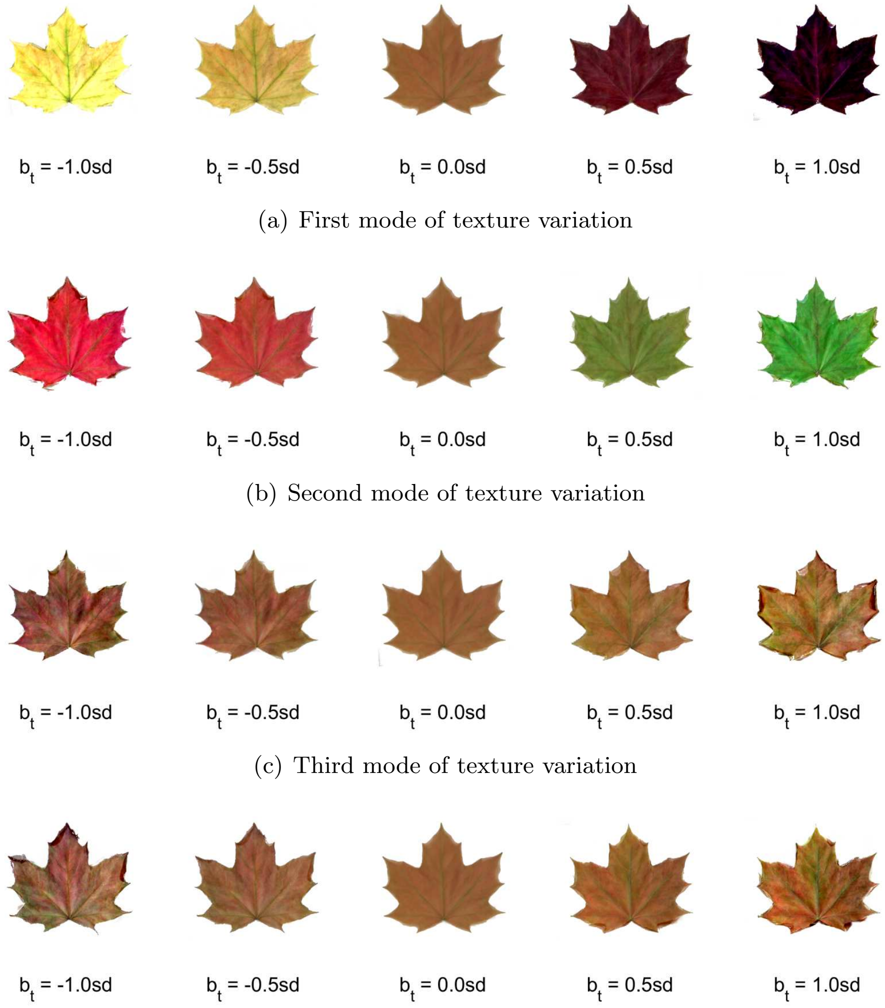
Since the principal components are ordered by significance, an approximate and compact encoding, $\mathbf{b}_g$, of $d\mathbf{g}$ can be obtained by using the first t_g components only,

$$\hat{\mathbf{b}}_g = \sum_{j=1}^{t_g} \left(\mathbf{p}_g^j\right)^T (\mathbf{g} - \bar{\mathbf{g}}) \tag{2.57}$$

where $\hat{\mathbf{b}}_g$ is a t_g element vector and typically $t_g << 2m$. An approximation, $\hat{\mathbf{g}}$, to the original sample texture can be reconstructed from $\hat{\mathbf{b}}_g$ as follows,

$$\hat{\mathbf{g}} = \sum_{j=1}^{t_g} \mathbf{p}_g^j \hat{b}_g^j + \bar{\mathbf{g}} \tag{2.58}$$

In the leaves example, 80% of the texture variance over the original sample is modelled by the first ten principal components (figure 2.14). Therefore a reasonable value for t_g would be $t_g = 10$, allowing any leaf texture to be encoded by a vector of only ten parameters.



(a) First mode of texture variation

(b) Second mode of texture variation

(c) Third mode of texture variation

(d) Fourth mode of texture variation

Figure 2.13: First four modes of texture variation. The first two modes predominantly indicate solid colour variation, whereas the third and fourth modes display mottling and surface shading.

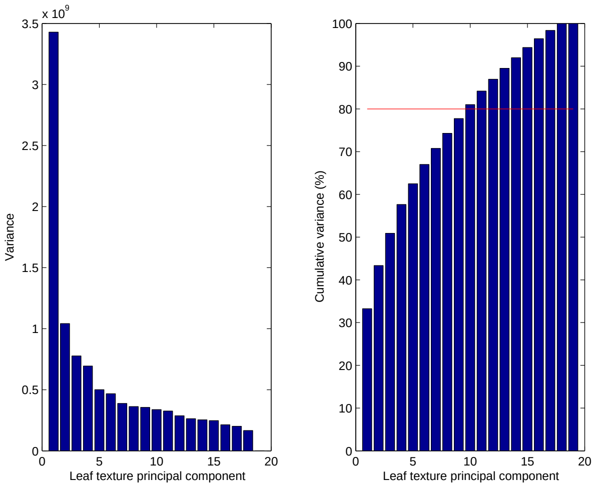


Figure 2.14: Variance associated with each *texture principal component* for the autumn leaves example (left). Cumulative variance (right) - 80% of the texture variation associated with the original dataset can be expressed by the first ten principal components.

2.4 Appearance model

Although the separate shape and texture models are sufficient for producing new instances of shapes and textures [43], a more compact model can be obtained by combining both shape and texture aspects in a single shape-texture representation. The combined model is often referred to as an *appearance model* [20] (this term is adopted here) though Baker et al use a slightly different terminology [64]. The appearance model captures correlations that exist between shape and texture. Its construction can be justified on the basis of two points. Firstly, it prevents implausible shape-texture combinations occuring when new instances are generated. Secondly, it allows a more compact representation of the pattern class. An appearance model is constructed using an additional PCA, in which the concatenated shape parameters ($\mathbf{b}_s$) and texture parameters ($\mathbf{b}_t$) are treated as observations.

2.4.1 Combining shape and texture

A method for constructing compact shape-texture or appearance representation can be summarized in the following steps.

1. Determine a scalar weighting value, w that scales the shape parameters such that equal significance is assigned to shape and texture (A typical

value for w can be inferred by comparing figure 2.9 and figure 2.14),

$$w = \left(\frac{\sum_{i=1}^{t_g} \left(\lambda_g^i\right)^{\frac{1}{2}}}{\sum_{i=1}^{t_s} \left(\lambda_s^i\right)^{\frac{1}{2}}} \right) \tag{2.59}$$

Other methods for determining w may be adopted in cases where the shape is considered to be of greater or lesser importance than the texture.

2. For each object/observation form a concatenated vector of weighted shape parameters and texture parameters,

$$\mathbf{b} = \begin{bmatrix} w\mathbf{b}_s \\ \mathbf{b}_g \end{bmatrix} = \begin{bmatrix} w\mathbf{P}_s^T\left(\mathbf{x} - \bar{\mathbf{x}}\right) \\ \mathbf{P}_g^T\left(\mathbf{g} - \bar{\mathbf{g}}\right) \end{bmatrix} \tag{2.60}$$

Each of the n vectors in the set $\{\mathbf{b}_i\}$ can be considered as an observation in a further PCA. $\mathbf{b}$ will already be in mean deviation form, a direct result of shape and texture observation being in mean deviation form.

3. Insert the n observations into a matrix, $\mathbf{B}$,

$$\mathbf{B} = \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ \mathbf{b}_1 & \mathbf{b}_2 & \dots & \mathbf{b}_n \\ \downarrow & \downarrow & & \downarrow \end{bmatrix} \tag{2.61}$$

4. Form the covariance matrix that describes how shape and texture parameters vary with respect to each other,

$$\mathbf{S}_a = \frac{1}{n-1}\mathbf{B}\mathbf{B}^T \tag{2.62}$$

5. For the leaves example there are more variables than observations and $\mathbf{S}_a$ will have at most n non-zero eigenvalues. The principal components are once again obtained using the relationship between the left singular vectors of $\mathbf{B}$ and the right singular vectors of $\mathbf{B}$, described by equations 2.15 & 2.16 in section 2.1.

$$\mathbf{Q} = \mathbf{B}\mathbf{V}_a\Lambda_a^{-\frac{1}{2}} \tag{2.63}$$

and

$$\mathbf{Q} = \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ \mathbf{q}_1 & \mathbf{q}_2 & \dots & \mathbf{q}_n \\ \downarrow & \downarrow & & \downarrow \end{bmatrix} \tag{2.64}$$

where $\mathbf{q}$ is a principal component of appearance, matrix Λ_a is a diagonal matrix containing the variances associated with the n components and $\mathbf{V}$ is a matrix containing the right singular vectors of $\mathbf{B}$

The principal components of appearance offer a compact shape-texture representation in terms of a newly derived vector of parameters, $\mathbf{c}$.

$$\mathbf{c} = \mathbf{Q}^T\mathbf{b} \tag{2.65}$$

Each element of $\mathbf{c}$ is an *appearance parameter* that embodies a global shape-texture characteristic of the pattern class. The first four modes of appearance variation for the autumn leaves pattern class are illustrated in figure 2.15.

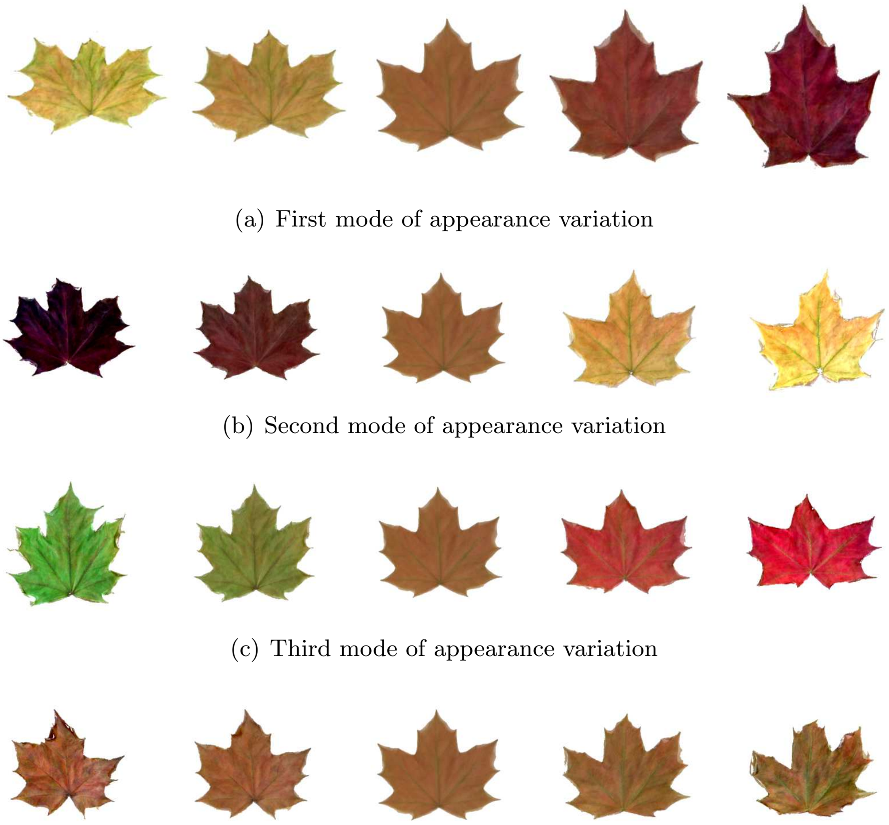
The PCs $\{\mathbf{q}_i\}$ are arranged in order of decreasing significance, hence an approximation to $\mathbf{b}$ can be obtained by forming a linear combination of the first t_a principal components. Figure 2.16 indicates that the first six components are sufficient for retaining 80% of the variance from the shape and texture models. Since $t_a < (t_s + t_g)$, the appearance model is a more compact representation than is afforded by the separate shape and texture PCA encodings.

From $\mathbf{c}$ an approximate object image can be reconstructed using the steps outlined below,

1. Form an approximation to $\mathbf{b}$ using the first t_a appearance principal components,

$$\hat{\mathbf{b}} = \sum_{i=1}^{t_a} \mathbf{q}_i c_i \tag{2.66}$$

where $\mathbf{q}_i$ is the i^{th} column of the matrix $\mathbf{Q}$ in equation 2.64



(a) First mode of appearance variation

(b) Second mode of appearance variation

(c) Third mode of appearance variation

(d) Fourth mode of appearance variation

Figure 2.15: First four modes of appearance variation. The first three modes indicate strong variations in colour and form. The fourth mode displays mottling and surface shading and asymmetry in shape.

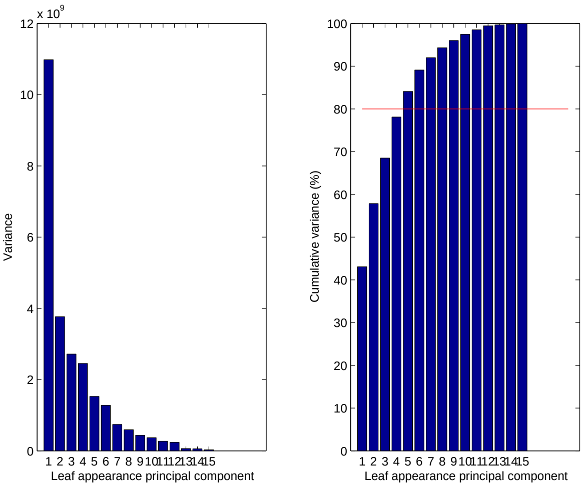


Figure 2.16: Variance associated with each *Appearance principal component* for the autumn leaves example (left). Cumulative variance (right) - 80% of the appearance variation associated with the original dataset can be expressed by the first six principal components.

2. Decouple the shape and texture parameters from **b**

$$\mathbf{b} = \begin{bmatrix} w\hat{\mathbf{b}}_s \\ \hat{\mathbf{b}}_g \end{bmatrix} \tag{2.67}$$

3. Reconstruct the approximate shape vector $\hat{\mathbf{x}}$,

$$\hat{\mathbf{x}} = \sum_{i=1}^{t_s} \mathbf{p}_s^i \hat{b}_s^i + \bar{\mathbf{x}} \tag{2.68}$$

   where $\mathbf{p}_s^i$ is a column vector containing the i^{th} shape principal component and $\hat{b}_s^i$ is a scalar representing the i^{th} shape parameter.

4. Reconstruct the approximate texture vector $\hat{\mathbf{g}}$,

$$\hat{\mathbf{g}} = \sum_{i=1}^{t_g} \mathbf{p}_g^i \hat{b}_g^i + \bar{\mathbf{g}} \tag{2.69}$$

5. The pixel intensities in the texture vector, $\hat{\mathbf{g}}$, are inserted into a 2D ar-

ray, by reversing the procedure outlined in figure 2.12. Thus, a shape-normalized texture-map is formed.

6. Warp the shape normalized texture from the mean object shape to the approximated shape $\hat{\mathbf{x}}$, thereby producing the final leaf image.

If the maximum number of PCs are used for shape texture and appearance in the reconstruction process then a perfect reconstruction of an in-sample object image can be achieved.

2.4.2 Generating new plausible examples

The appearance model can be used to synthesize new plausible examples from the chosen pattern class. For each new example, this is achieved by selecting an appropriate vector of model parameters, $\mathbf{c}$, from which the an image can be reconstructed. Appropriate parameter values are determined by fitting a statistical model $p(\mathbf{c})$ to the data points corresponding to the original training sample. For the leaf example the probability density function, $p(\mathbf{c})$, is approximately a standard multivariate normal distribution (see figure 2.17),

$$p(\mathbf{c}) = N(\mathbf{c}; 0, \Lambda) = (2\pi)^{-\frac{n}{2}} |\Lambda|^{-\frac{1}{2}} exp\left\{-\frac{1}{2}\mathbf{c}^T\Lambda^{-1}\mathbf{c}\right\} \tag{2.70}$$

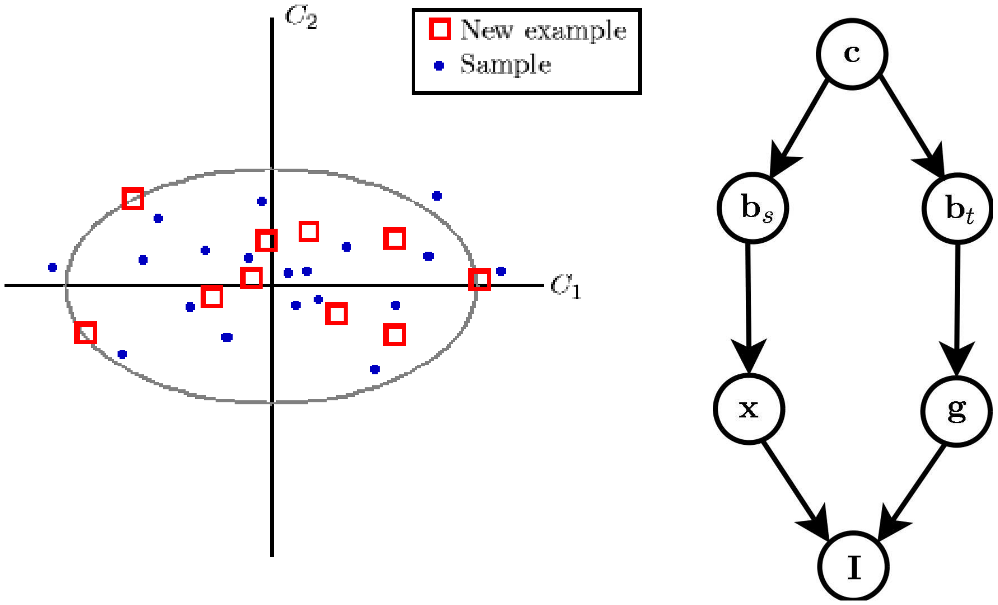
where Λ is a diagonal matrix of variances, in which the element $\lambda_k = \Lambda_{k,k}$ is equal to the variance of the k^{th} parameter variable. For random instances of new objects, parameters can be obtained using a pseudo random number generator (examples provided in figure 2.18).

Once the appropriate parameter values have been selected, the corresponding object image can be reconstructed in the usual manner according to equations 2.67-2.69 (see also figure 2.17).

2.5 Introduction to evolutionary algorithms

The facial composite system described in this thesis constructs a likeness to a target face by employing an evolutionary algorithm to optimize a set of appearance model parameters. This section provides the necessary background material on evolutionary optimization procedures.

Evolutionary algorithm is a generic term referring to an optimization procedure that mimics biological evolution. Any population-based optimization algorithm that uses mechanisms such as reproduction, mutation, recombination



(a) Data points follow a multivariate normal distribution. The grey ellipse signifies two standard deviations from the mean object. process

(b) A schematic diagram indicating the steps required to reconstruct an image, $\mathbf{I}$, from its appearance model representation $\mathbf{c}$. The shape model parameters $\mathbf{b}_s$ and texture model parameters $\mathbf{b}_t$ are recovered from $\mathbf{c}$ which can be used to obtain shape $\mathbf{x}$ and texture vectors $\mathbf{g}$ respectively. The elements of $\mathbf{g}$ are inserted into a 2d array, thus forming the shape normalized texture, which is warped to the shape defined by $\mathbf{x}$ thereby producing the reconstructed image.

Figure 2.17: New examples of faces can be synthesized by selecting parameters $\{c_i\}$ from a multivariate normal distribution and performing the image reconstruction procedure outlined by sub-figure (b). Sub-figure (a) indicates the process of generating model parameters, corresponding to new plausible faces, from the distribution defined by the original training data.

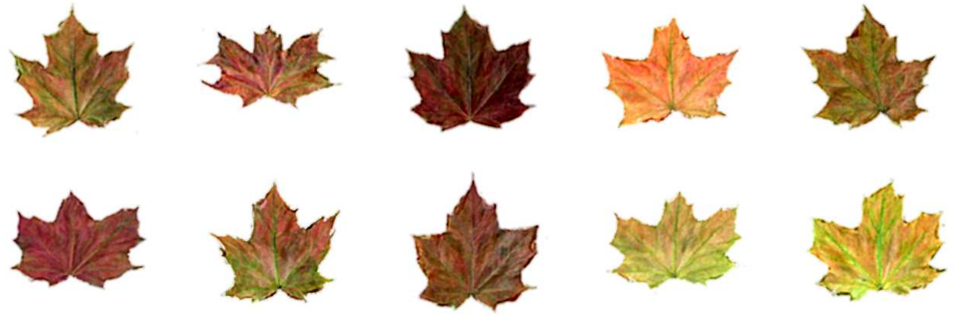


Figure 2.18: New examples synthesized from the autumn leaves appearance model.

(see genetic operators), natural selection can be categorized as an evolutionary algorithm. Candidate solutions to the optimization problem play the role of individuals in a population, and a fitness function determines the environment inhabited by the candidate solutions. Evolution of the population then takes place ayfter the repeated application of the evolutionary operators. The exact sequence of operators varies between the different forms of EA. Some of the more notable forms are *genetic algorithms*, *evolutionary strategies*, *evolutionary programming* and *genetic programming*.

2.5.1 Classification of evolutionary algorithms

Genetic algorithms

Genetic algorithms (GA) are one class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, natural selection, and crossover (also known as recombination). GAs are used to determine approximate solutions to optimization and search problems and are particularly useful when classical optimization methods cannot be applied.

The standard GA operates on a population of bit-strings, although real numbers (Gray-code) can be accommodated by some variants of GA. Each bit-string represents a coded candidate solution, and is commonly known as a genotype. A decoded candidate solution is referred to as a phenotype, the real world object corresponding to the genetic material contained in the genotype. A single candidate solution is also known as an individual. The evolution starts from a population of completely random individuals and proceeds in steps or generations. In each generation, the individuals are assigned a fitness score as defined by a fitness function or objective function. Individuals are selected for breeding on a stochastic basis in relation to their fitness score. The offspring of the breeding process are inserted into a new generation, and the process continues until a satisfactory solution has been achieved.

Evolutionary strategies

Evolution strategies (ES) primarily use real-vector coding, and not binary coding, as is commonly used in GAs. As with evolutionary algorithms in general, the evolutionary operators of mutation, crossover, and environmental selection are used.

The first ES variants, developed in Germany by Rechenberg [75] and Schwefel [78] for engineering optimization problems, used one parent rather than an initial population of individuals. From this parent, λ offspring were constructed and all $1 + \lambda$ solutions placed in competition. This process is normally denoted by $(1 + \lambda) - ES$ where $\lambda \geq 1$ and ES indicates that the optimization process is an evolutionary strategy. Contemporary versions usually employ a population $(\mu +, \lambda) - ES$. In general, $(\mu, \lambda) - ES$ selection outperforms $(\mu + \lambda) - ES$ selection, since it allows for temporal deterioration of the population's best solution, and therefore may overcome local optima.

In an ES, mutation is performed by adding a gaussian distributed random value simultaneously to each decision variable (in the context of the appearance model, a decision variable is an encoding of an appearance parameter). A strategy parameter is assigned to each decision variable that controls the mutation strength (ie. the standard deviation of this distribution). The strategy parameters are adaptive and, in general, change at each generation. Usually, in an ES both decision variables and strategy parameters are optimized.

Evolutionary programming

Evolutionary programming (EP) is similar to ES, although it was developed separately by Fogel [31] for use in artificial intelligence.

There is no precise definition of EP, and it is sometimes difficult to differentiate between EP and ES, although in a standard implementation mutation is the only evolutionary operator employed in EP. The basic EP-cycle is similar to a strictly mutation-based $(\mu + \mu) - ES$, though a stochastic selection scheme is used instead of deterministic selection. The mutation strength is also adapted differently. Contrary to ES, the mutation strength is a function of the parents fitness and each parent produces one offspring.

Genetic programming

The aim of genetic programming (GP) is not to optimize a vector of variables, but rather to determine a computer program that performs a predefined task in an optimal way. In the GP context, a population comprises computer programs, each of which is constructed from sets of functions and terminals. Functions may be arithmetic, mathematical, boolean, loop operators, domain-specific functions etc. The set of terminals consists of variables and constants. The fitness of each program is assigned according to its average performance based on a set of test problems. As with GAs, GP employs a crossover operator that interchanges segments of code to produce offspring. GP is computationally expensive, and its use has only recently become practical with advances in computer hardware technology. Quantum computing, electronic design, game playing, sorting and searching are some of the areas to which GP has been applied.

2.5.2 Encoding methods

The set of all possible solutions to an optimization problem defines a *solution space*, which is specific to the problem. An accurate representation of the solution space is a necessary requirement for building a robust evolutionary algorithm. The solution space is characterised by the encoding method used. Many different encodings have been proposed, some of which are better suited to certain problems than others. In this section, a brief introduction to the most widely used encoding schemes is presented.

Binary code

Binary encoding was introduced by Holland [50], and is the encoding method of choice for GAs, in which each solution is encoded as a unique string of 0s and 1s. Let $B(x, l)$ represent a function that encodes the real value x into its binary representation of length l. For example, if $x = 1$ then $B(x, 4) = 0001$. The parameter l determines the accuracy of the encoding. A large value of l will enable accurate encoding, but will also reduce the algorithm's performance. In practice, a compromise is sought between speed of convergence and accuracy.

Integer numbers can be easily encoded in binary form without loss of accuracy. Real numbers can also be encoded, albeit with limited accuracy. For example, let x be a real valued variable defined over the range,

$$-1 \leq x \leq 2 \quad x \in R$$

An acceptable coding precision must be decided upon, in this case one decimal place will be considered sufficient. Hence the domain of x will be segmented into 30 intervals of 0.1. To encode to this level of precision, five bits are required (since $\sum_{i=0}^{4} = 31$). Let $\mathbf{D}$ be a decoding function that maps binary numbers to real numbers.

$$\begin{aligned} \mathbf{D}\,(00000) &= -1 \qquad (\textit{lower bound } I_{min}) \\ \mathbf{D}\,(11111) &= 2 \qquad\;\; (\textit{upper bound } I_{min}) \end{aligned}$$

All other bit-strings ($b_4\ b_3\ b_2\ b_1\ b_0$) are mapped to a value contained in the interval $[I_{min}, I_{max}]$. First, the strings are converted from base 2 to base 10,

$$x' = \sum_{i=0}^{4} b_i 2^i$$

Then the decoded real number is given by,

$$x = I_{min} + \frac{I_{max} - I_{min}}{2^5 - 1} x'$$

Discrete codings such as these can lead to problems. For instance, the above coding scheme maps both binary strings, 10000 and 01111 to $x = .5$.

Gray code

A limitation of the binary encoding method is that swapping the value of a single decision variable from 1 to 0, or vice-versa, may cause a large change in the decoded variable. The binary representations for the integer number 7 and 8 is, $B(7,4) = 0111$ and $B(8,4) = 1000$ respectively. Thus, for consecutive integer numbers, we need to flip all of the binary elements. This is not consistent with the idea that small changes to the genotype should result in small changes in the phenotype. Gray code is an encoding method based on $1s$ and $0s$ that overcomes this problem. This encoding procedure bears the name of Frank Gray, who patented the use of this coding for shaft encoders in 1953 [39]. The main characteristic of a Gray code is that adjacent integer numbers differ only by one bit. A comparison between the binary and Gray code representations of integers $0 - 15$ is provided in table 2.1

Integer	Binary Code	Gray Code	Integer	Binary Code	Gray Code
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000

Table 2.1: Comparison between binary and Gray code representations of integer values.

Real value coding

Real valued representations have become widely used in evolutionary optimization problems. As the name suggests, in this encoding method each decision variable is represented by a real value. This method is not susceptible to the same coding errors as binary coding. A detailed comparison between binary and real encoding can be found in Janikow [53]. For some applications it is advantageous to transform (e.g. logarithmic mapping) the real variables before encoding them.

2.5.3 Evolutionary operators

By definition, all evolutionary algorithms employ evolutionary operators. These are summarized as follows,

Natural selection

After deciding the encoding method, the second decision to make is how to perform the selection. Factors to take into consideration are how many individuals from the population will be selected for reproduction, how many offspring each individual will produce and how these offspring will be fed into the algorithm. Some of the most frequently used selection rules are outlined below.

Fitness-proportional selection

This is the selection method used by Holland in his original genetic algorithm [50]. In this selection method, the probability for an individual to be selected for reproduction is proportional to its fitness. The probability is calculated by dividing the fitness of the individual by the total fitness of the population. There are many different ways of implementing fitness-proportionatal selection. A widely used method is called ”Roulette Wheel” selection. This process simulates the spinning of a roulette wheel and the random selection of one of its slots, where each slot corresponds to the fitness of an individual in the population. The greater the member’s fitness, the greater the slot size and the probability of the member being selected.

Elitism

An *elitist model* is any selection method in which the best solution so far is copied into the next generation. Hence the fittest solution is always retained by the algorithm. Elitism was originally proposed by DeJong [56] and variations on the elitist model are employed in many applications. Care must be taken when implementing an elitism selection method to avoid premature convergence of the population, resulting in the search becoming trapped in a local optimum.

Rank selection

In rank-based fitness assignment, the individuals comprising the population are sorted according to the values returned by an objective function. The fitness assigned to each individual depends only on its position in the rank [49] and not on the actual value returned by the objective function. Unlike fitness-proportionate selection, rank selection is robust to premature convergence.

Tournament selection

In tournament selection a random sub-sample of individuals from the population is chosen. The fittest individual from the sub-sample is selected for breeding. This procedure is repeated until the required number of parents have been selected. Each selected parent has an equal chance of contributing genetic material to the offspring comprising the next generation. The parameter for tournament selection is the subsample size, or tournament size as it is know in this context, and can range from 2 to the number of individuals in the population.

2.5.4 Crossover

Crossover, also known as recombination, is the main evolutionary operator used in genetic algorithms (see Holland [50]). The basic idea behind this operator is the combination of well defined small pieces of genetic code, also called building blocks. Crossover combines building blocks extracted from fit individuals, with the aim of constructing offspring that exhibit a greater fitness than their parents. There are many different ways in which recombination can be achieved. The most important methods are single point crossover, multiple point crossover and uniform crossover. Each of these three methods is described below.

Single point crossover

This crossover method, selects at random an identical position within the genetic string of each parent, which signifies a cutting point in the genotype. Each cut yields two building blocks per parent. The building blocks from the parents are then interchanged to produce the offspring. Single point crossover is illustrated in figure 2.19

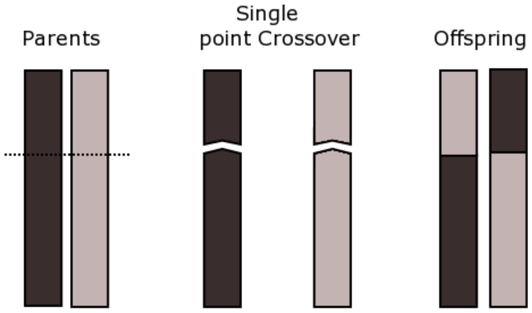


Figure 2.19: Single Point Crossover

Multi-point crossover

For multi-point crossover [84], $m > 1$ crossover positions are chosen at random and sorted in ascending order. Then, the building blocks between successive crossover points are exchanged between the two parents to produce two new offspring. The building block between the first variable and the first crossover point is not exchanged between individuals. Figure 2.20 illustrates this process. Multi-point crossover encourages the exploration of the search space, rather than favouring the convergence to highly fit individuals early in the search, thus making the search more robust than single point crossover.

Uniform crossover

Uniform crossover [88] differs from the single-point and multi-point crossover schemes. Each offspring is created by copying individual decision variables from one parent to the other as defined by a crossover mask. The crossover mask

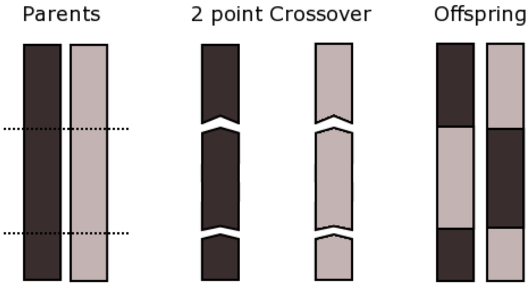


Figure 2.20: Multi-point crossover with number of crossover points $m = 2$.

itself consists of a string containing the same number of bits as the parent bit-strings. A new mask of randomly selected bit values is created for every crossover process. A one in the mask indicates that the corresponding bit in the first parent is to be interchanged with the bit located at the same position in the second parent. It is possible to assign different probabilities of occurence to zeros and ones in the crossover mask, thereby influencing the disruptive effect of uniform crossover.

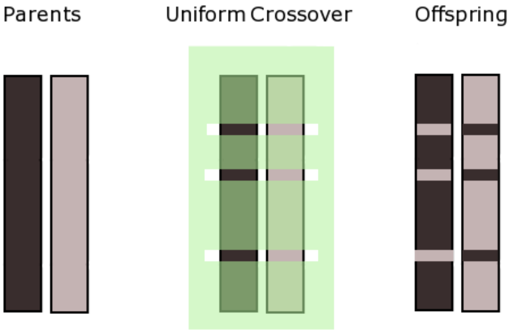


Figure 2.21: Uniform crossover, in which individual bits are interchanged between parents to yield new offspring.

2.5.5 Mutation

Although crossover is the main evolutionary operator employed in genetic algorithms, it plays little or no role in evolution strategies and evolutionary programming in which mutation is the dominant operator. The main difference between mutation and crossover is that while crossover cannot create new information, mutation can. The affect of the mutation operator is most easily illustrated by example. Consider two parents, x_1 and x_2, encoded as binary bit-strings, 0101 and 0010 respectively. By inspection of the bit strings, it is clear that the crossover operator can not alter the value of the first bit which will always remain equal to 0. This can result in the exclusion of potentially optimal solutions. The mutation operator overcomes this problem by introducing a probability that the value of the first bit, or indeed any other bit, will flip from 0 to 1 or vise-versa.

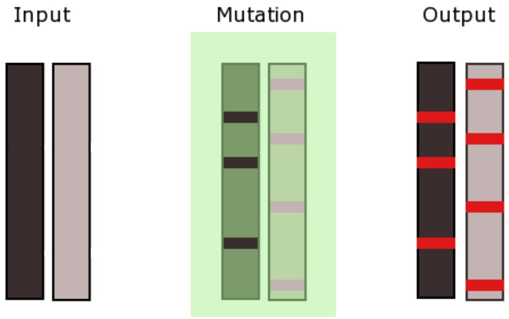


Figure 2.22: Genetic mutation.

In general, the mutation operator will randomly sample positions within the genetic string and flip the values of the bits at the sampled positions. All the bits have an equal probability of being selected for mutation.

Figure 2.23 gives a graphical representation of a simple evolutionary algorithm, indicating the order in which the evolutionary operators are applied. For this illustrative example a genetic algorithm was chosen because it employs all of the main evolutionary operators.

2.5.6 Parameter tuning

Fine tuning the parameters that control the evolutionary algorithms, such as probability of mutation, number of crossover points, probability or rate for crossover operation and population size, can have a significant effect on the performance of the algorithm. Finding the best values for a given problem is difficult because these parameters are not independent from each other.

Early studies attempted to determine the parameters that resulted in a universally optimal algorithm. Dejong [56] performed an analytical study of the parameters relating to genetic algorithms and found that a population size of around 50 to 100 with a 60% chance of a single point crossover occuring, and a low probability of mutation of 0.001 per bit provided the best combination of parameters.

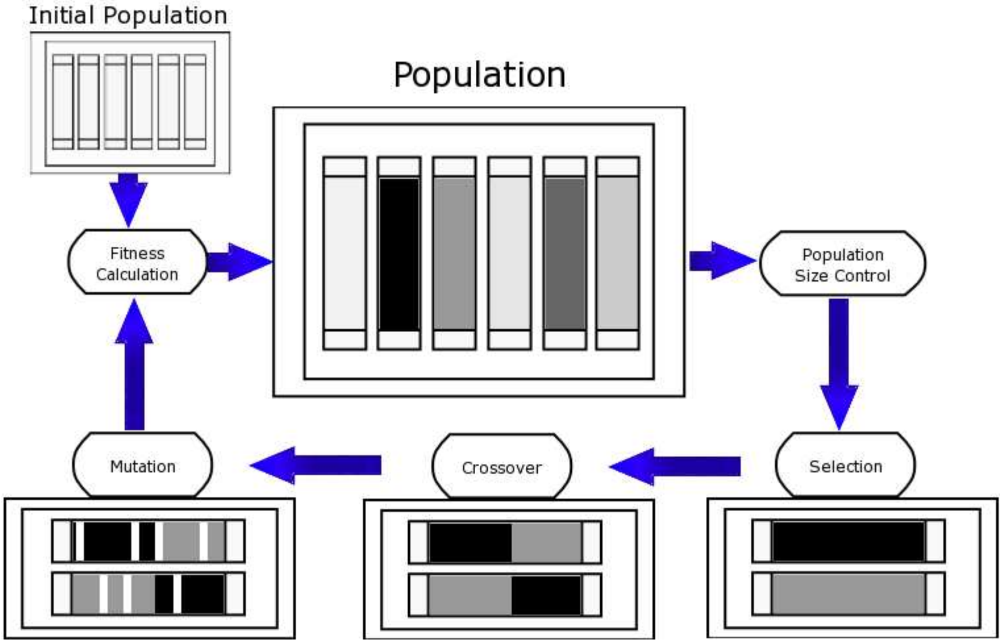


Figure 2.23: A schematic diagram illustrating the order in which operators are applied in a typical genetic algorithm. Black indicates the fittest individual and white, the least fit individual.

Grefenstette [41] introduced the idea of using a genetic algorithm to determine the parameter values of a second genetic algorithm. Grefenstette found that for the same problem studied by Dejong and that led to Dejong's estimated values, a population size of 30, with a crossover rate of 0.95, and a mutation rate of 0.01 with an elitist selection would outperform Dejong's algorithm.

Later studies indicated that the optimal parameters were problem specific, with the optimal parameters being dictated by the specific task. Moreover, the possibility of using different values within the same problem was investigated. This lead to the implementation of dynamic parameters and self-adapting parameters that changed depending on the current state of the evolutionary algorithm [26].

2.6 Summary

This chapter provided the mathematical background relevant to the facial composite system described in this thesis.

An historical and mathematical account of the statistical method of principal components analysis was followed by the mathematical procedure for constructing an appearance model (AM). The AM enabled objects belonging to chosen pattern class to be represented by a compact vector of parameters. By modelling the distribution of these parameters, it was shown that new examples of objects could be synthesized. In the following chapter the AM is revisited, where the details specific to modelling the human face are explained. This model of facial appearance, provides a mechanism for synthesizing new examples of faces, and is therefore an essential component in the facial composite system.

The problem of constructing a likeness to a suspect's face can be expressed as an optimization problem in which an appropriate set of appearance model parameters is sought. The objective function for this problem is unknown and an optimal set of parameters must be determined according to the witness' assessment of facial likeness. Evolutionary algorithms (EA) provide a flexible tool for optimization problems, to which there is no analytical solution. The concept of evolutionary search procedures was also introduced. The basic evolutionary operators; mutation crossover and selection were outlined and different methods for encoding parameters were discussed. The following chapter includes the design of an EA for the specific task of obtaining a facial likeness.

Chapter 3

EigenFIT - design and core implementation

In this chapter, the core technical design and implementation of a novel procedure for generating facial composites is described. Computer software based on this procedure has been developed, which will be referred to hereafter as EigenFIT (Eigen Facial Identification Technique). The chapter begins with the motivation for developing the EigenFIT composite system, followed by an outline of the system's functionality and how it may be used to generate a composite image. Subsequent sections describe the system's construction, starting with the data capture process required for building the generative appearance model. This is followed by an account of the appearance model construction and an evolutionary search algorithm designed to enable an operator to achieve convergence to a target face. A general automated method for applying hairstyles to the generated faces is presented. This is followed by a preliminary study into a potentially superior method for applying hairstyles. The last section outlines a technique for overriding the evolutionary process in which the witness is able to lock the shape of individual facial features. A step by step example illustrating the production of a facial composite using the EigenFIT system is provided in Appendix D.

3.1 Motivation

Commercially available composite systems to date have relied on assembling individual facial parts in order to *compose* a target face. This approach to constructing facial composites conflicts with a significant body of psychological literature which has shown that we are better at recognising faces as a whole rather than as a sum of their individual parts or features [95, 89, 73]. Our in-ability to recognise faces on a feature-by-feature basis is due to the importance of the relative positions of the individual features within the face. Many of the established systems also rely strongly on a preliminary cognitive interview in which the witness is required to provide a verbal description of the face. Recalling descriptions of faces from memory and assigning semantic labels to these descriptions is a difficult task. Conversely, a much easier task is to recognise the identity of a subject when presented with an image of their face. Taking the psychological issues into consideration, it is reasonable to assume that a composite system that incorporates whole face stimuli and does not entail a lengthy verbalization processes, may yield superior results.

3.2 System overview - EasyFIT mode

The EigenFIT system has been carefully designed in an attempt to address the problems associated with current commercial systems. The basic elements in this process can be described in general terms as follows,

- Initially, a witness is presented with an array of randomly generated faces to which he or she must respond.
- The witness is required to make a subjective judgement of facial likeness each image in the array with respect to the suspect. There are in principle a number of different ways in which this can be implemented (e.g. ranking or assigning full scale ratings). However, in the preferred method, the witness is required to select a *single face* from the sample as the best likeness to the target face. Faces that do not resemble the suspect in any way can be removed, thereby allowing the witness to concentrate on the remaining faces in the array.
- Based on the selected face at this step, a new array of faces is produced and displayed to the witness.
- The witness responds to the new faces in the same manner as the previous array of images. This process is iterated until a satisfactory likeness to the target is produced.

Implementation of a composite system based on this protocol requires a method for *creating plausible faces* and the means to *propagate facial characteristics through successive iterations.* These requirements are met by the three main elements comprising the EigenFIT system. These are,

1. A generative face model that is capable of producing photo-realistic whole face stimuli.
2. A procedure for evolving a facial likeness.
3. A graphical user interface that allows the operator to interact with the system in an intuitive manner.

The generative model used in this work was based upon an *appearance model* (AM), a technique that has received a lot of interest within the computer vision community since its inception [20]. An AM of the human face was constructed and the natural variation exhibited by a sample of real faces was modelled by estimating the underlying probability density function relating to the associated appearance model parameters. This enabled *new plausible* examples of *whole faces* to be synthesized (see section 3.4.5), thereby overcoming the problems inherent in the feature based approach to composite construction.

A likeness to a suspect's face can be achieved by determining an appropriate set of appearance model parameter values. These optimal parameters can not be obtained analytically, dictating an alternative iterative optimization method. In this work an interactive *evolutionary algorithm* was employed to determine the optimal appearance model parameters, and hence a likeness to the suspect's face.

The evolutionary algorithm was driven by the witness' response to consecutive arrays of facial stimuli via an intuitive *graphical user interface*. The interface was designed to utilize the mind's capacity for recognising faces and to negate the problems associated with verbalizing facial appearance and the misinterpretation of verbal descriptions. Figure 3.1 shows how the main components of the system and the witness interact in the composite construction process.

To accelerate the composite process, an initial starting point for the evolutionary algorithm was obtained by establishing the sex and ethnicity of the suspect via graphical cues. Thus, in the EigenFIT system the emphasis has shifted from the cognitive interview towards a *graphical* interviewing procedure that does not require the witness to verbalize facial descriptions recalled from memory.

EigenFIT has been developed using the MATLAB programming language. Due to MATLAB's limitation in speed and graphics handling, a commercial beta version was later coded in C++ using Borland Builder. EigenFIT has been successfully tested on Windows XP and Windows 2000 operating systems. It runs satisfactorily on any computer (Pentium II 400MHz or better) with only modest processing capacity.

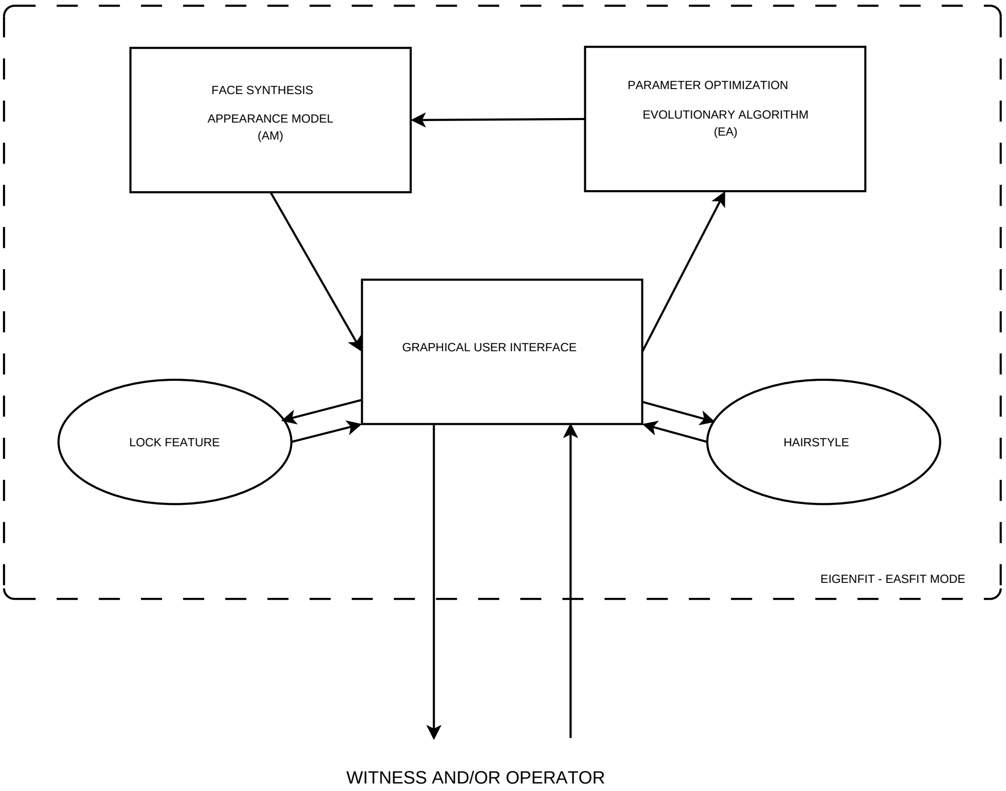


Figure 3.1: EigenFIT (EasyFIT): A conceptual diagram indicating the interaction between the three main components; the *appearance model*, *evolutionary algorithm* and *graphical user interface*. Additionally, functionality is provided by a *hairstyle* tool and a *lock feature* tool that enables the operator to intervene in the global evolutionary process.

3.3 Design of graphical user interface - EasyFIT

The interface design was motivated by a need for *cognitive simplicity*, employing an intuitive graphical approach to facial composite construction. The key idea underpinning this approach is that the witness is guided through the process of generating a facial composite with the aid of graphical cues that do not rely on explicit verbalization. The interface operates in two modes; *expertFIT*, which is outlined in chapter 4 and *easyFIT* which is described in this chapter. The EasyFIT mode embodies the core implementation of EigenFIT. In this form, computer generated *virtual faces* are presented to the witness in a three by three configuration. A summary of the layout and functionality of the EasyFIT

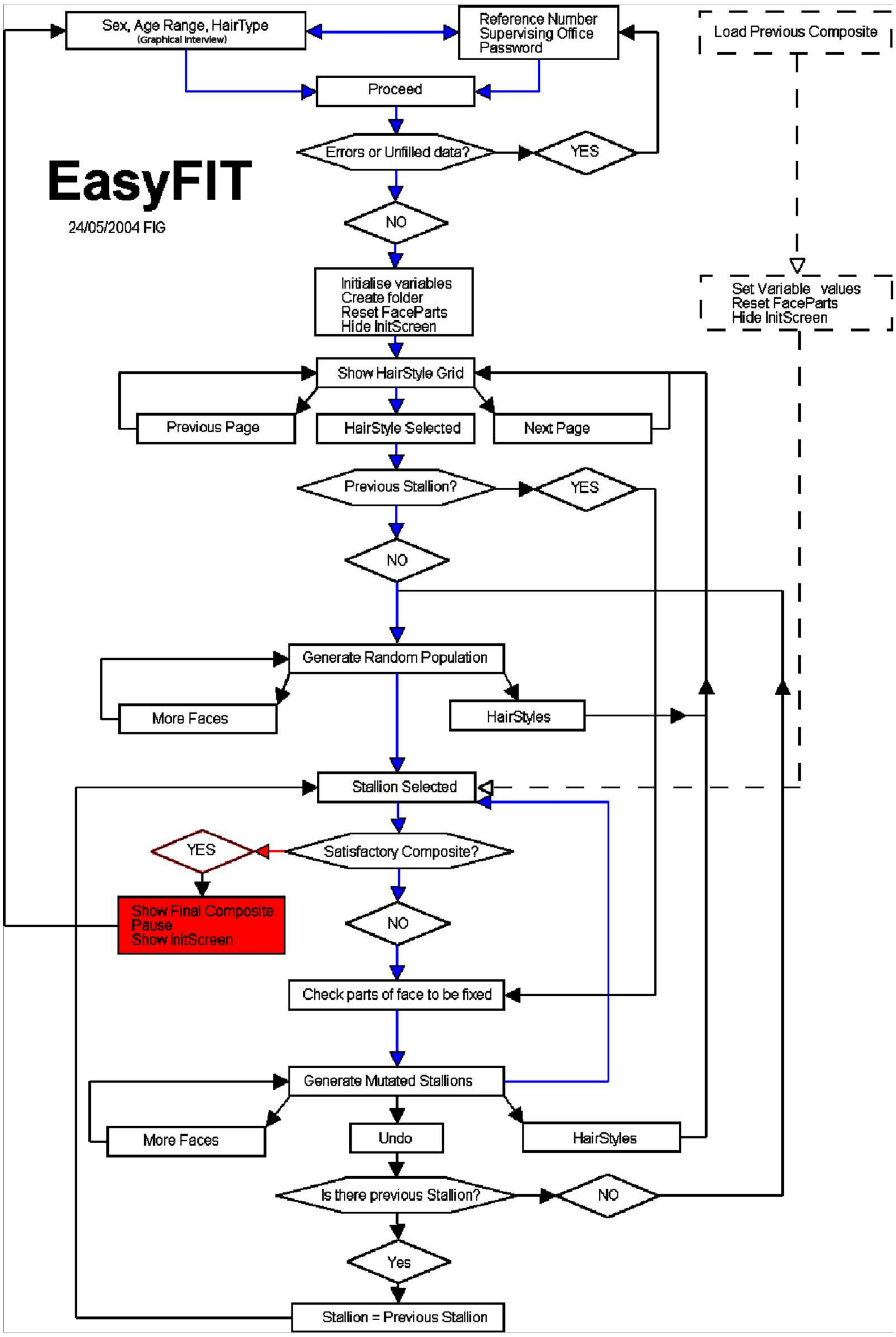


Figure 3.2: Flow diagram indicating the procedure for generating a composite image using the EigenFIT software in EasyFIT mode.

interface, shown in figure 3.3, is provided below,

- **Backdrop for generation:** Area of interface in which the computer generated face stimuli are displayed. The witness is required to select one of nine faces (the faces are collectively known as a generation) that are displayed on the backdrop. The selection process invokes a new generation of faces. Repetition of the selection process for subsequent generations allows a likeness to the target face to be evolved.

- **Remove face:** This icon allows the witness to *hide* faces that are significantly different from the target, thus allowing them to concentrate on the remaining visible faces. Toggling the icon switches the face between its visible and hidden state.

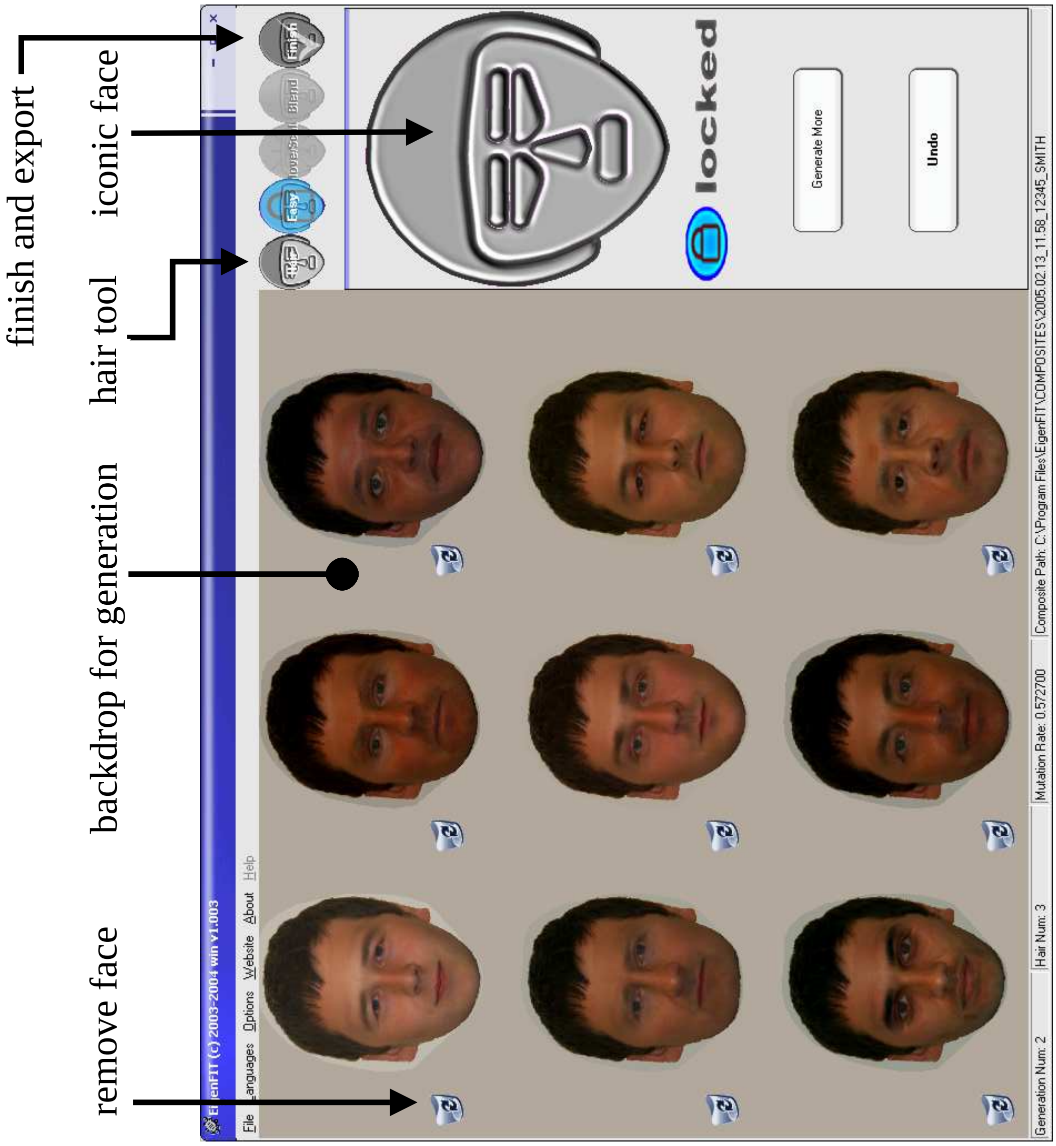
- **Hair tool:** Displays the hair selection tool, allowing the witness to choose an appropriate hairstyle. The hairstyles are displayed in order of preference according to a semantic filter. This approach to selecting a hairstyle is similar to the method used in commercially available systems such as E-FIT [3].

- **Iconic face:** The iconic face comprises feature buttons corresponding to the eyes, eyebrows, mouth, nose and face shape. Once a feature button has been selected it turns blue to indicate that it has been locked. In subsequent generations the shape of the selected feature remains unchanged until the feature button is deselected. More than one feature button may be selected at once. The iconic face is an essential component of the lock feature tool (described in the later part of this chapter) and the feature manipulation tool (covered in Chapter 4).

- **Finish and export:** Ends the composite process with the option to export the image to a graphics package for post processing. The optional post processing step allows distinctive markings such as scars and tattoos to be drawn onto the composite image.

- **'Generate More' button:** Produces a new generation with more facial variation than the previous generation. Repeatedly selecting the 'Generate More' button increases the amount of facial variation further.

- **'Undo' button:** Ignores the last face, selected by the operator, thereby returning to the previous generation.



Generation Num: 2
Hair Num: 3
Mutation Rate: 0.572700
Composite Path: C:\Program Files\EigenFIT\COMPOSITES\2005.02.13_11.58_12345_SMITH

Figure 3.3: Screen dump of the EigenFIT graphical user interface - EasyFIT mode.

3.4 Generative model of facial appearance

Active appearance models have previously been used for computer pattern recognition applications [21, 64] in which a face patch is synthesized and morphed to match a target face using an *automated* iterative fitting algorithm.

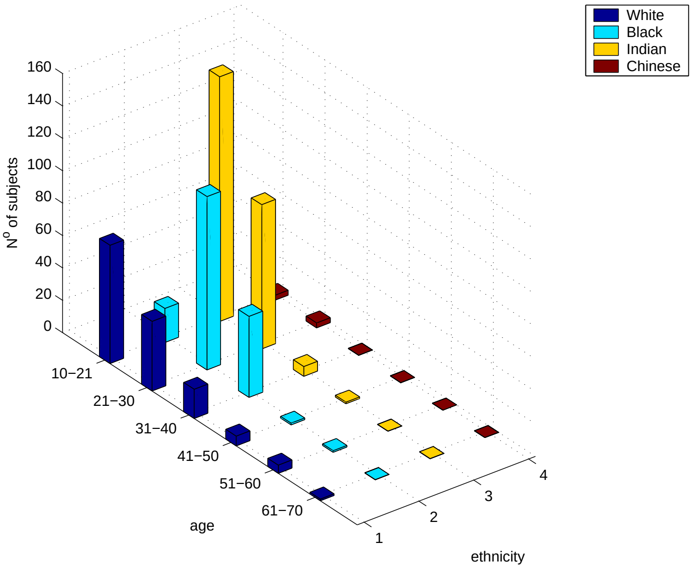
Conversely, in facial composite applications, the fitting procedure is guided by the *witness'* response to face stimuli. Hence, we are only concerned with the capability of the model for synthesizing new examples of face images. Accordingly, the word 'active' has been dropped and the term *appearance model* is used when referring to the PCA based method for synthesizing new examples of faces. In Chapter 2, the mathematical procedure for constructing an appearance model was described in detail. In this section, the specific steps required to construct an appearance model of the human face are presented. By modelling the probability density function of model parameters, the AM can be used to synthesize new examples of faces.

3.4.1 Facial demographics

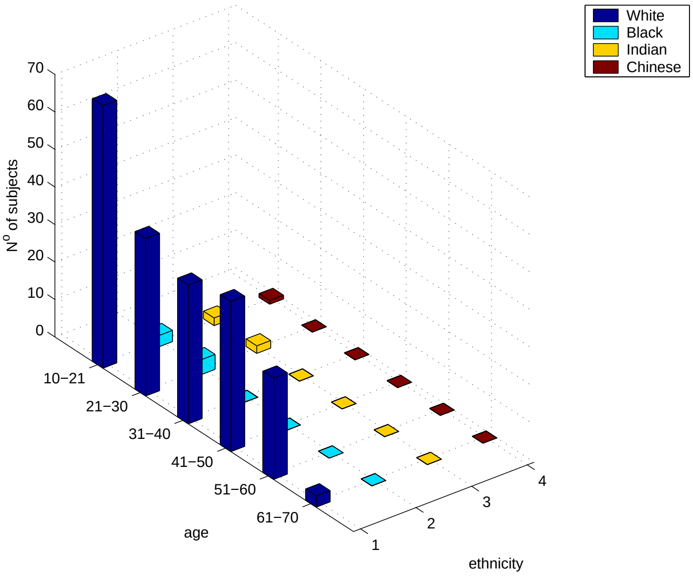
The objective of the appearance model is to model the full range of natural variation that occurs in both face shape and face texture. In order to achieve this aim, a comprehensive sample of training images is required. A sample of 823 face images was used to construct the AM. The demographics of this sample are provided in the bar charts of figure 3.4.1. White males, black males and Indian males were adequately sampled whereas Chinese males were not (see figure 3.4.1a). The number of samples of White females was sufficient for representing facial variation within the White female population. Other female ethnic groups were not well represented (see figure 3.4.1b).

3.4.2 Image capture protocol

Subjects were required to sit in a chair and face directly towards a six mega pixel digital camera. Special care was taken to normalize the pose. Where necessary a subject was asked to rotate their head slightly, so that their face was 'square on' to the camera lens. Variations in head pose result in shape modes which exhibit undesirable rotations. These can be difficult to remove from the shape model [48]. The camera was mounted on a tripod and positioned at a distance of 1.5 metres from the subject, with the subject's face occupying as much of the image as was feasible. Variability in the height of subjects was accommodated by using the vertical adjustment on the tripod. Images were captured in a laboratory without windows, preventing variations in lighting conditions due to



(a) male



(b) female

Figure 3.4: Demographics of AM training data comprising 823 sample faces. For clarity, the demographics of the male sample has been plotted separately from the demographics of the females.

daylight entering the room. Subjects were illuminated by a single fluorescent strip light and the camera flash. A table of camera settings is provided in appendix A.

3.4.3 Shape delineation tool

Although automated methods have been developed for positioning landmarks [19, 57], manually placed landmarks can, in many cases, offer a higher degree of accuracy. However, there is a considerable labour burden associated with the task of manually placing landmark points. The shape model used in this work demands that 190 landmark points be placed on each of 823 sample images. To make this process less tedious software was written to simplify this task, allowing the shape of image objects to delineated easily and relatively quickly. This *shape delineation tool* included a simple user interface that could be operated by an untrained user (see figure 3.5).

The shape delineation tool (or landmarking tool), based on a least squares polynomial fitting procedure, was coded. One or more polynomial line segment(s) were used to represent the boundary of each facial feature. The shape of each polynomial line segment was determined by a set of control points $\{x, y\}$. The aim is to obtain the n degree polynomial in x that best fits y in the least squares sense. This provides a smooth curve, $\hat{y}(x)$, that can be manipulated by hand to follow the contour of a given feature. The problem can be written in terms of the Vandermonde's matrix, $\mathbf{V}$, as,

$$\mathbf{Vp} \cong \mathbf{y} = \hat{\mathbf{y}} \tag{3.1}$$

or in tableau form as,

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ \vdots & \vdots & \vdots & \vdots & \\ 1 & x_3 & x_1^3 & \dots & x_3^n \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ \vdots \\ p_{n+1} \end{bmatrix} \cong \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n+1} \end{bmatrix} \tag{3.2}$$

The elements of $\mathbf{V}$ are powers of $x_i^{(j-1)}$ and the coefficients that form $\mathbf{p}$ are to be determined by least squares methods. Vector $\mathbf{y}$ contains the y coordinates of the control points. The properties of each curve segment were set according to an empirical measure of their suitability to a specified feature. Some features

Feature	Orientation	Order of polynomial	N^o of interpolated points	Save intersecting interpolated points
Upper edge of left eyebrow	horizontal	4	8	yes
Lower edge of left eyebrow	horizontal	4	8	no
Upper edge of right eyebrow	horizontal	4	8	yes
Lower edge of right eyebrow	horizontal	4	8	no
Upper edge of left eye	horizontal	4	8	yes
Lower edge of left eye	horizontal	4	8	no
Upper edge of right eye	horizontal	4	8	yes
Lower edge of right eye	horizontal	4	8	no
Left upper edge of top lip	horizontal	4	8	yes
Right upper edge of top lip	horizontal	4	8	yes
Lower edge of top lip	horizontal	6	8	no
Upper edge of bottom lip	horizontal	6	8	no
Lower edge of bottom lip	horizontal	4	8	no
Chin	horizontal	4	24	yes
Left side of face	vertical	4	16	no
Forehead	horizontal	5	24	yes
Right side of face	vertical	4	16	no
Left side of nose	vertical	5	8	no
Right side of nose	vertical	5	8	no
Base of nose	horizontal	2	8	no
There is no curve associated with this point	none	4	8	no

Table 3.1: The shape delineation tool allowed landmarks to be placed relatively quickly and easily. This was achieved by manipulating a set of polynomial curve sections to fit the boundaries of the main facial features and the perimeter of the head. The properties of these polynomial curve sections are presented in this table. Where a base landmark controls more than one curve (for instance, at the corner of the mouth), care must be taken not to duplicate landmarks (see last column in the above table).

exhibit more curvature than others, and therefore must be modelled using a higher order of polynomial. Predominantly horizontal features, such as the mouth, are best delineated using a function of the form $y\left(x\right)$ whereas for vertical features the role of x, y coordinates should be interchanged. This was easily achieved by setting the elements of $\mathbf{V}$ to $y_i^{(j-1)}$ and making the x coordinate the response variable in the least squares problem.

The two end points of each line segment are classified as anatomical landmarks because they were positioned at salient points on the feature boundary (e.g. at each corner of the eye). Pseudo landmarks for each feature were obtained by sampling the coordinates of equidistant points points along the interpolated curve. The number of landmarks and polynomial order for a given curve section were determined by its typical length and intricacy of shape. Table 3.1 lists the properties of the curve sections used.

Figure 3.5 depicts the interface of the shape delineation tool, which has been annotated to provide an overview of its functionality. Control points can be positioned by using the left mouse button to 'click and drag'. A detailed explanation of how to use this interface is provided in appendix B.

A sequence of images illustrating the typical steps required to landmark the mouth region is given in figure 3.6. The initial positions of the landmarks, for the whole face configuration, are set by placing three points; two points at the outer corners of the eyes and a point at the base of the nose. A transformation of the mean point configuration (representing the mean face shape) based on the position of these three points is used to obtain an approximate position for the whole point configuration. The curve sections can then be fitted to their corresponding features by manipulating the control points (represented by dots in the figure). In some instances, the landmarking procedure can be aided by initially translating a whole feature shape using the *lock* control as shown in figure 3.6 a-b.

3.4.4 Appearance model construction

The method for constructing an appearance model was described in detail in Chapter 2. Here, the specific details concerning the construction of an appearance model of the *human face* using the shape and texture data are presented.

Point model

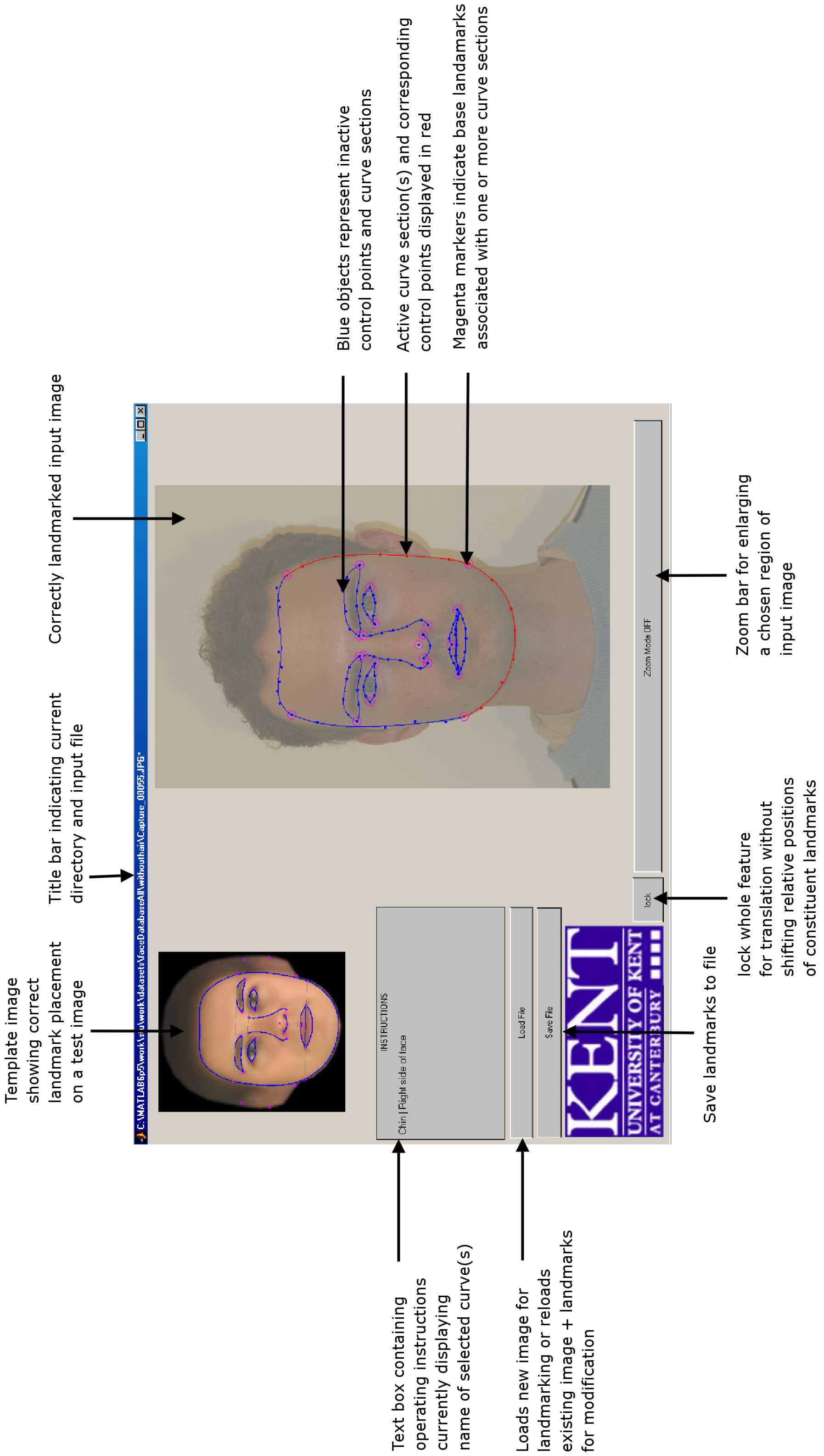
A point model was chosen that delineated the main facial features and the perimeter of the face. Accordingly, the face morphology was represented using 190 landmark points as illustrated in figure 3.7.

Each of the 823 training faces were landmarked according to the specified point model, thus 823 point sets were obtained. The point sets were aligned using the Procrustes method (see section 2.2.2), allowing variations in face shape to be analysed independently of scale, position and rotation.

A compact shape representation was obtained by performing a principal components analysis on the face shapes according to the procedure described in Chapter 2, by equations 2.29-2.36. The shape of any *out-of-sample face* can be approximated by a linear combination of the shape principal components as,

$$\hat{\mathbf{x}} = \bar{\mathbf{x}} + \sum_{j}^{t_s} \mathbf{p}_s^j \hat{b}_s^j \tag{3.3}$$

where $\bar{\mathbf{x}}$ is the mean face shape, $\mathbf{p}_s^j$ is the j^{th} principal component and $\hat{b}_s^j$



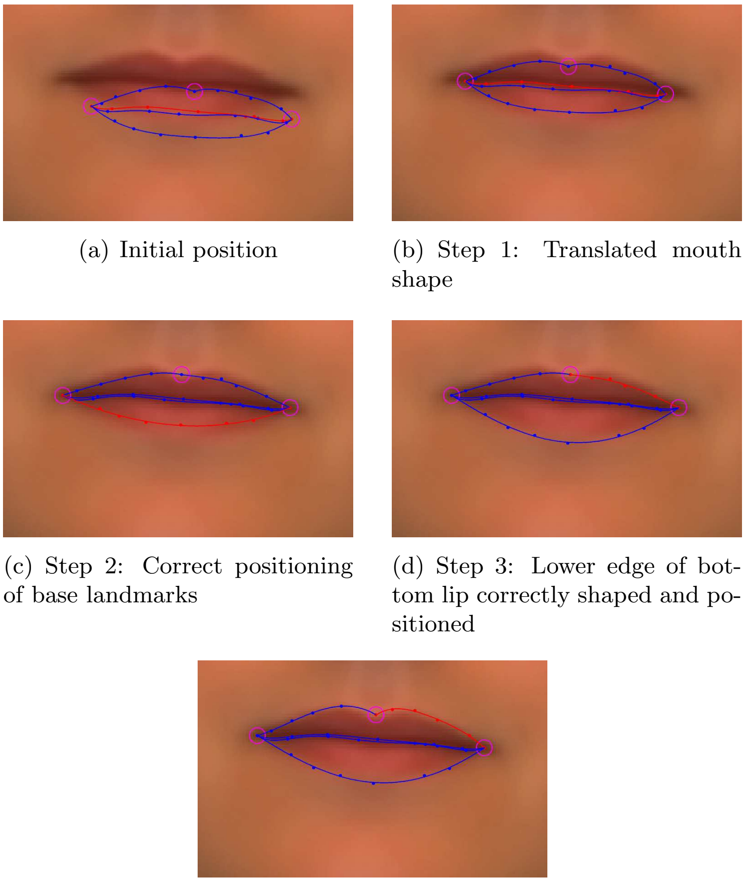
C:\MATLAB6p5\work\stu\work\datasets\faceDatabaseAll\withouthair\Capture_00055.JPG*

INSTRUCTIONS

Chin | Right side of face

Template image showing correct landmark placement on a test image
Title bar indicating current directory and input file
Correctly landmarked input image
Blue objects represent inactive control points and curve sections
Active curve section(s) and corresponding control points displayed in red
Magenta markers indicate base landmarks associated with one or more curve sections
Zoom bar for enlarging a chosen region of input image
lock whole feature for translation without shifting relative positions of constituent landmarks
Save landmarks to file
Loads new image for landmarking or reloads existing image + landmarks for modification
Text box containing operating instructions currently displaying name of selected curve(s)
Load File
Save File
lock
Zoom Mode OFF
KENT UNIVERSITY OF KENT AT CANTERBURY

Figure 3.5: User interface for shape delineation tool. The anatomical landmarks are labelled with magenta circles. The polynomial curve segments and their associated control points are plotted in blue, except for the curve that is currently being manipulated, which is plotted in red. Note: The contrast of the input image in has been reduced to make the landmarks more visible.



(a) Initial position

(b) Step 1: Translated mouth shape

(c) Step 2: Correct positioning of base landmarks

(d) Step 3: Lower edge of bottom lip correctly shaped and positioned

(e) Step 4: Correctly landmarked mouth

Figure 3.6: Typical steps taken to correctly landmark the mouth region of a sample face. The above images were extracted from the shape delineation tool interface and enlarged to illustrate the process more clearly. The left mouse button was used to 'click and drag' the control points such that the curve sections were located on the boundary of the feature.

is a shape parameter that dictates the influence of the j^{th} component on the generated face shape $\hat{\mathbf{x}}$.

Table 3.2 illustrates the first three modes of shape variation with respect to the mean face shape. The first mode appears to capture variation in forehead height, and to a lesser extent, chin length. However, care must be taken when interpreting the variation represented by this mode, since what appears to be variability in the forehead height is partially attributable to disparities in hairline position. This is undesirable because the aim is to model the facial features and the perimeter of the head, which is unrelated to the hairline. A solution is to weight [55] the relevant landmark points so that they are less significant

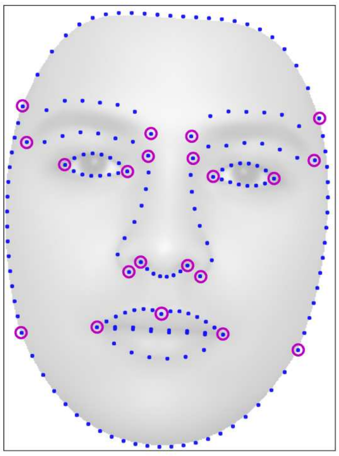


Figure 3.7: Face shape point model. Magenta circular markers represent anatomical landmarks. Blue markers represent interpolated landmarks which follow the feature boundaries.

in the construction of the shape model than the landmarks pertaining to the facial features and remainder of the head shape. The second and third modes are more indicative of true face shape variation.

Texture model

A texture model was constructed that described the variations in pixel values over the sample images (see Chapter 2, section 2.3). Initially, each of the 823 training face images were warped to the mean face shape, such that a correspondence between 'like' pixels was obtained (i.e. an exact non-rigid alignment was obtained such that each facial feature was positioned at the same image coordinates in every one of the warped sample images). The pixel intensity values were extracted column-wise from these shape normalized images, to form 823 texture vectors.

A compact texture representation was obtained by performing a principal components analysis on the texture vectors as described in Chapter 2, by equations 2.47-2.54. The texture of any *out-of-sample face* can be approximated by a linear combination of the texture principal components,

$$\hat{\mathbf{g}} = \bar{\mathbf{g}} + \sum_{j}^{t_g} \mathbf{p}_g^j \hat{b}_g^j \tag{3.4}$$

where $\bar{\mathbf{g}}$ is the mean face texture, $\mathbf{p}_g^j$ is the j^{th} principal component and $\hat{b}_g^j$ is a texture parameter that dictates the influence of the j^{th} component on the generated face texture $\hat{\mathbf{g}}$.

The shape principal components can easily be calculated using vector and matrix operations. Conversely, the texture principal components can not be calculated in this way. The dimensionality of the texture data (typically $10^4 - 10^7$ elements per vector) prevents all of the sample textures being loaded into memory at any instance. Instead, the n[1] texture principal components, $\left\{\mathbf{p}_g^j\right\}$, must be constructed iteratively using a nested for loop, as shown in algorithm 1.

Algorithm 1 Computationally viable method for generating principal components from image data

```
$\mathbf{p}_g^j = \mathbf{0}$ {initialize principal component as a $3m$ by 1 null vector}
for $i = 1$ to $n$ do
  load $d\mathbf{g}_i$ {load first observation}
  $\mathbf{p}_g^j = d\mathbf{g}_i \mathbf{v}_g^j(i) + \mathbf{p}_g^j$ {add contribution from the $i^{th}$ observation}
end for
$\mathbf{p}_g^j = \mathbf{p}_g^j \lambda_j^{-\frac{1}{2}} (n-1)^{-\frac{1}{2}}$ {normalize principal component}
write $\mathbf{p}_g^j$ to file
```

Table 3.3 illustrates the first three modes of texture variation with respect to the mean face texture. The first mode predominantly captures variation in skin pigmentation associated with *ethnicity*. The second and third modes appear to represent aspects of texture variation due to *race and gender*.

Combined appearance model

An appearance model was constructed that simultaneously captured both shape and texture aspects of human faces using the procedure previously described in Chapter 2, by equations 2.59-2.64. A compact vector representation, $\mathbf{c} = [c_i, c_2, \ldots, c_n]^T$, of the appearance of an out-of-sample face can be expressed in terms of its shape and texture as follows,

$$\mathbf{c} = \mathbf{Q}^T \begin{bmatrix} w\mathbf{b}_s \\ \mathbf{b}_g \end{bmatrix} \equiv \mathbf{Q}^T \begin{bmatrix} w\mathbf{P}_s^T d\mathbf{x} \\ \mathbf{P}_g^T d\mathbf{g} \end{bmatrix} \tag{3.5}$$

where the columns of $\mathbf{Q}$, $\mathbf{P}_s$ and $\mathbf{P}_g$ are the appearance, shape and texture principal components respectively. $d\mathbf{g}$ is the face texture vector in mean deviation form and $d\mathbf{x}$ is the face shape vector in mean deviation form. The vectors

[1] The dimensionality of the data usually dictates that there will be at most n principal components with corresponding eigenvalues that are non-zero, where n is the number of sampled faces.

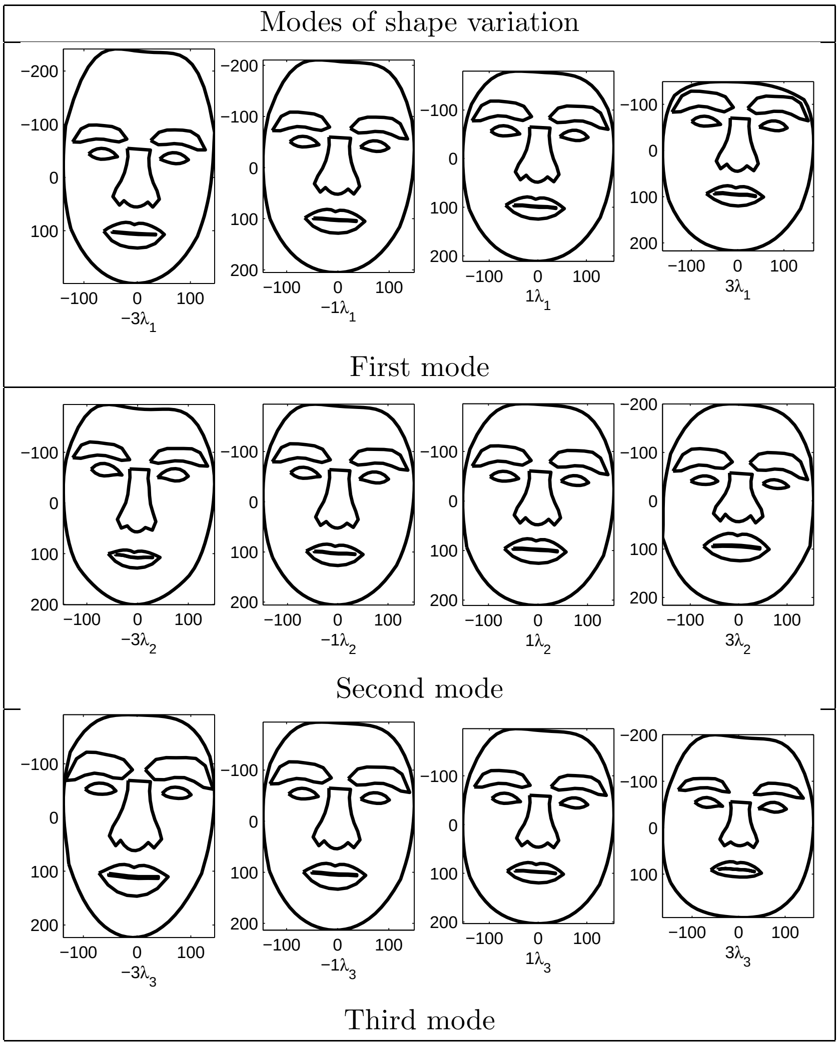


Table 3.2: Shape modes capturing the natural shape variation in the dataset. The j^{th} mode is illustrated by adding a proportion of the j^{th} shape principal component to the mean face shape - $\left(\bar{\mathbf{x}} - \mathbf{p}_s^j 3\sqrt{\lambda_j}\right) < \mathbf{x} < \left(\bar{\mathbf{x}} + \mathbf{p}_s^j 3\sqrt{\lambda_j}\right)$, where λ_j is the variance associated with the j^{th} mode of variation.

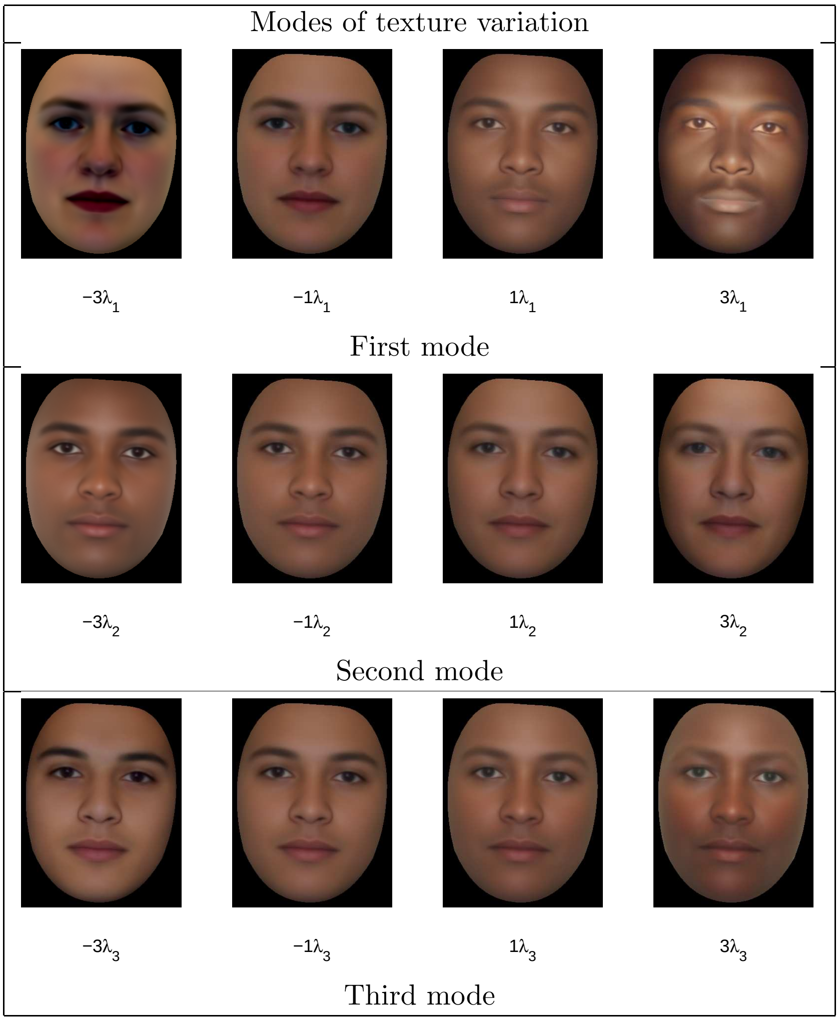


Table 3.3: Texture modes capturing the natural colour variation in the dataset. The j^{th} mode is illustrated by adding a proportion of the j^{th} texture principal component to the mean face texture - $\left(\bar{\mathbf{g}} - \mathbf{p}_g^j 3\sqrt{\lambda_j}\right) < \mathbf{g} < \left(\bar{\mathbf{g}} + \mathbf{p}_g^j 3\sqrt{\lambda_j}\right)$, where λ_j is the variance associated with the j^{th} mode of variation.

$\mathbf{b}_s$ and $\mathbf{b}_g$ are the shape and texture parameter vectors respectively and w is a scalar that determines the relative significance of shape and texture.

Conversely, new plausible examples of faces can be synthesized by sampling the parameters $\{\mathbf{c}_i\}$ from a multivariate distribution and then manipulating equation 3.5 to obtain the corresponding face shape and face texture.

The first three modes of appearance variation are illustrated in table 3.4. These modes correspond to the most dominant of the t_a axes defining a *parameter space*. The first mode indicates a change in race. It exhibits masculine attributes due to the Black, Indian and White male demographic groups represented in the training sample. Only White females are adequately represented in the training sample (see section 3.4.1), hence there is no equivalent mode for female faces. The second and third modes represent variation in sex, race and overall face shape.

3.4.5 Generating new examples of faces

The provision for synthesizing new examples of faces is an essential component in the EigenFIT system. To generate *plausible* faces, the probability density function (PDF) of appearance parameter values must be estimated. Knowing this PDF is important for two reasons. Firstly, it allows new random faces to be synthesized, thereby forming a starting point for the evolutionary search procedure. Secondly, it enables random plausible variations of a chosen face to be produced; the process on which the evolutionary search procedure is based.

Consider the n training samples. These can be represented collectively as a point cloud in the parameter space, in which each sample is defined by a unique point. The point cloud is modelled using a single multivariate normal (abbreviated to SMN elsewhere in this thesis) probability density function,

$$N\left(\mathbf{c};\mathbf{0},\Lambda\right) = \left(2\pi\right)^{-\frac{n}{2}} \left|\Lambda\right|^{-\frac{1}{2}} exp\left\{-\frac{1}{2}\left(\mathbf{c}^T\Lambda^{-1}\mathbf{c}\right)\right\} \tag{3.6}$$

where $\mathbf{c}$ is a vector of appearance model parameters, $\left[c_1\ c_2\ \ldots\ c_n\right]^T$, Λ is a diagonal matrix containing the variances associated with each mode of appearance and $\mathbf{0}$ is the mean vector indicating that the distribution of parameter values is centred about the origin of the parameter space. An approximate likeness to a chosen face that is *not* represented in the training sample, can be synthesized by selecting the appropriate set of parameters $\{c_i\}$ (the subject of section 3.6) and following the process for generating an image from model appearance parameters (mathematical details were presented at the end of sec-

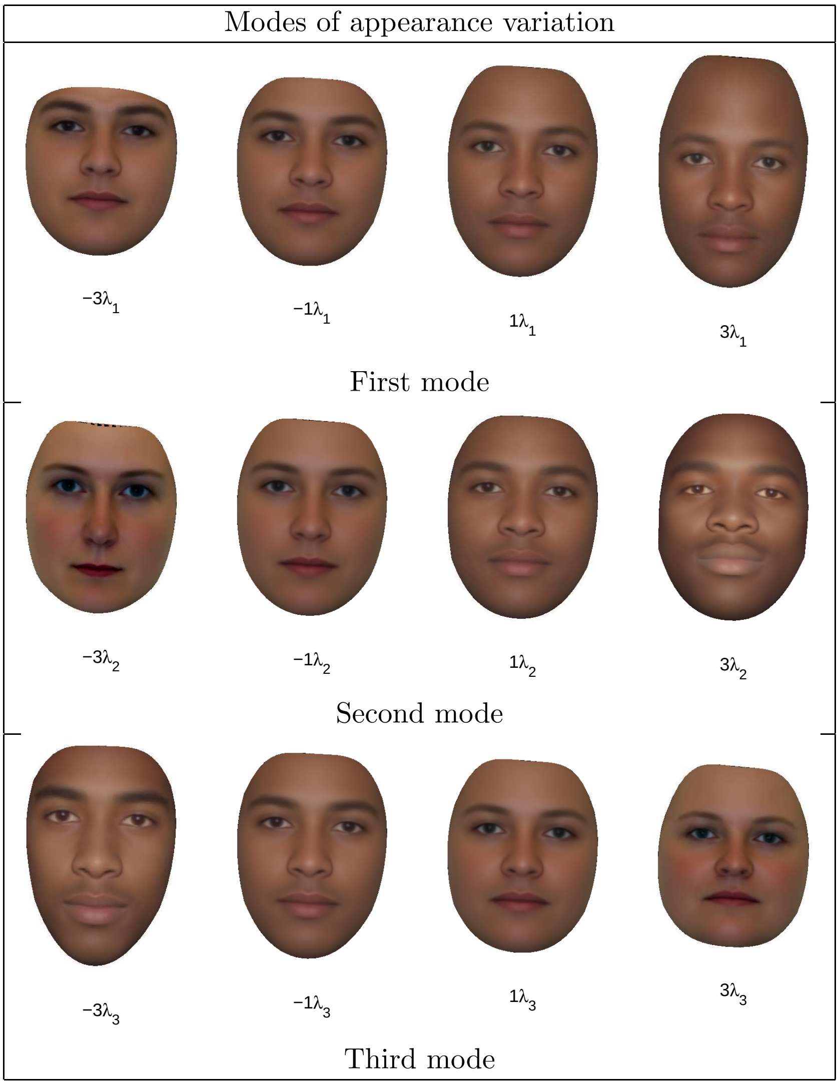


Table 3.4: Appearance modes representing the natural shape, skin pigmentation and shading variation in the dataset. The j^{th} mode is illustrated by adding a proportion of the j^{th} appearance principal component to the mean face appearance - $\left(\bar{\mathbf{c}} - \mathbf{q}_j 3\sqrt{\lambda_j}\right) < \mathbf{c}_j < \left(\bar{\mathbf{c}} + \mathbf{q}_j 3\sqrt{\lambda_j}\right)$

tion 2.4.1 in Chapter 2). Examples of random faces can be synthesized by sampling for parameters $\{c_i\}$ from $N(\mathbf{c}; \mathbf{0}, \Lambda)$. Computationally, this is easily achieved by scaling standard normal variables $[z_1\ z_2\ \ldots\ z_n]^T$ obtained from a pseudo-random number generator (PRNG),

$$\mathbf{c} = \Lambda^{\frac{1}{2}}\mathbf{z} \quad or \quad \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} \lambda_1^{\frac{1}{2}} & 0 & 0 & 0 \\ 0 & \lambda_2^{\frac{1}{2}} & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \lambda_n^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} \tag{3.7}$$

where $\lambda_1^{\frac{1}{2}}, \lambda_2^{\frac{1}{2}}, \lambda_3^{\frac{1}{2}}$ etc are the standard deviations associated with the $1^{st}, 2^{nd}$ and 3^{rd} parameters. The SMN distribution, $N(\mathbf{c}; \mathbf{0}, \Lambda)$, describes a model that embodies *all* of the training faces and is therefore independent of their associated demographic classification.

Demographic sub-sample models

Although the SMN model generally produces plausible faces, instances of unrealistic faces may also occur. For example, faces that simultaneously exhibit both male and female characteristics. Furthermore, the SMN model is of limited use for facial composite applications because it not does easily allow prior demographic knowledge to be incorporated in the initial population (see section 3.6). A superior approach is to model the individual demographic groups as *separate multivariate normal* distributions, (e.g. $N_{BM}(\mathbf{c}; \mu_{BM}, \Sigma_{BM})$ for black males), within the parameter space as depicted in figure 3.8.

Hence, faces that exhibit characteristics associated with a chosen demographic group can be generated using the appropriate sub-sample model. The evolutionary search procedure requires an initial pseudo-random population of faces which must be drawn from a chosen sub-sample model. Let $\mathbf{z}$ be a vector of independent random variables sampled from a multivariate standard normal distribution. A vector of transformed variables was sought such that,

$$\mathbf{c} = \mathbf{A}\mathbf{z} + \mu \tag{3.8}$$

where μ is a translation vector, representing the mean face of the demographic sub-sample and $\mathbf{A}$ is a matrix to be determined. The requirement that $\mathbf{A}$ be of full row rank implies that the dimension of $\mathbf{c}$ can be no greater than

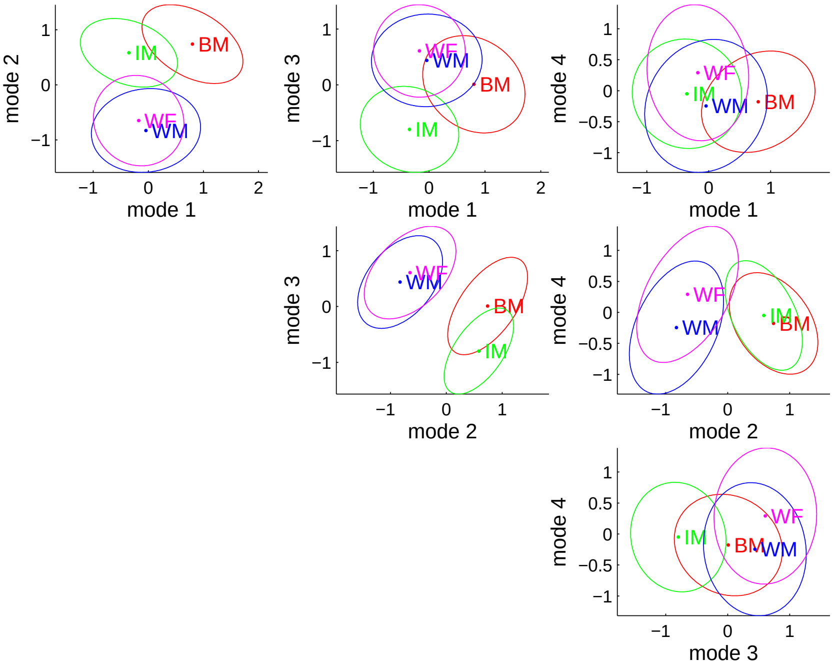


Figure 3.8: Figure indicating the distributions for black males (BM), Indian males (IM), white males (WM) and white females (WF) over the first three dimensions of the parameter space. Ellipses represent 2σ contour lines of the sub-sample models.

the dimension of $\mathbf{z}$ and that none of the variables within $\mathbf{Az}$ is expressible as a linear combination of the others. The covariance matrix of $\mathbf{c}$ is defined as,

$$\begin{aligned}
\Sigma_c &= \left\langle [\mathbf{c} - \langle \mathbf{c} \rangle] [\mathbf{c} - \langle \mathbf{c} \rangle]^T \right\rangle \\
\Sigma_c &= \left\langle [\mathbf{Az} + \mu - (\mathbf{A} \langle \mathbf{z} \rangle + \mu)] [\mathbf{Az} + \mu - (\mathbf{A} \langle \mathbf{z} \rangle + \mu)]^T \right\rangle \\
\Sigma_c &= \left\langle [\mathbf{A} (\mathbf{z} - \langle \mathbf{z} \rangle)] [\mathbf{A} (\mathbf{z} - \langle \mathbf{z} \rangle)]^T \right\rangle \\
\Sigma_c &= \mathbf{A} \left\langle [(\mathbf{z} - \langle \mathbf{z} \rangle)] [(\mathbf{z} - \langle \mathbf{z} \rangle)]^T \right\rangle \mathbf{A}^T \\
\Sigma_c &= \mathbf{A} \Sigma_z \mathbf{A}^T
\end{aligned} \tag{3.9}$$

Σ_z is diagonal for independent variables and equal to the identity matrix when $\{z_i\}$ are standard normal variables. Hence equation 3.9 simplifies as follows,

$$\Sigma_c = \mathbf{A} \Sigma_z \mathbf{A}^T = \mathbf{A} \mathbf{A}^T \tag{3.10}$$

for

$$z_i \sim N(0, 1)$$

The density function of $\mathbf{c}$ is found via the change-of-variable technique. This involves expressing $\mathbf{z}$ in terms of the inverse function $\mathbf{z}(\mathbf{c}) = \mathbf{A}^{-1}(\mathbf{c} - \mu)$. Using equation 3.10 the Jacobian of the transformation can be written as,

$$\left\| \frac{\partial \mathbf{z}}{\partial \mathbf{c}} \right\| = \left| \mathbf{A}^{-1} \right| = |\Sigma_c|^{-\frac{1}{2}} \tag{3.11}$$

Therefore, the resulting probability density function of the *multivariate normal* distribution of the transformed variables is,

$$N(\mathbf{c}; \mu, \Sigma_c) = (2\pi)^{-\frac{n}{2}} |\Sigma|^{-\frac{1}{2}} exp \left\{ -\frac{1}{2} (\mathbf{c} - \mu)^T \Sigma_c^{-1} (\mathbf{c} - \mu) \right\} \tag{3.12}$$

To transform the standard normal variables, and hence synthesize examples of faces from the chosen sub-sample model, the matrix $\mathbf{A}$ must be determined via a decomposition of the covariance matrix Σ_c (Σ_c is positive definite and symmetric). $\mathbf{A}$ is not unique and various decompositions can be found that satisfy $\mathbf{\Sigma}_c = \mathbf{A}\mathbf{A}^T$. A standard approach is to use a Cholesky decomposition in which the covariance matrix is decomposed into the product of a lower triangular matrix $\mathbf{L}$ and its transpose,

$$\mathbf{\Sigma} = \mathbf{L}\mathbf{L}^T \tag{3.13}$$

substituting **L** for **A** in equation 3.8 gives,

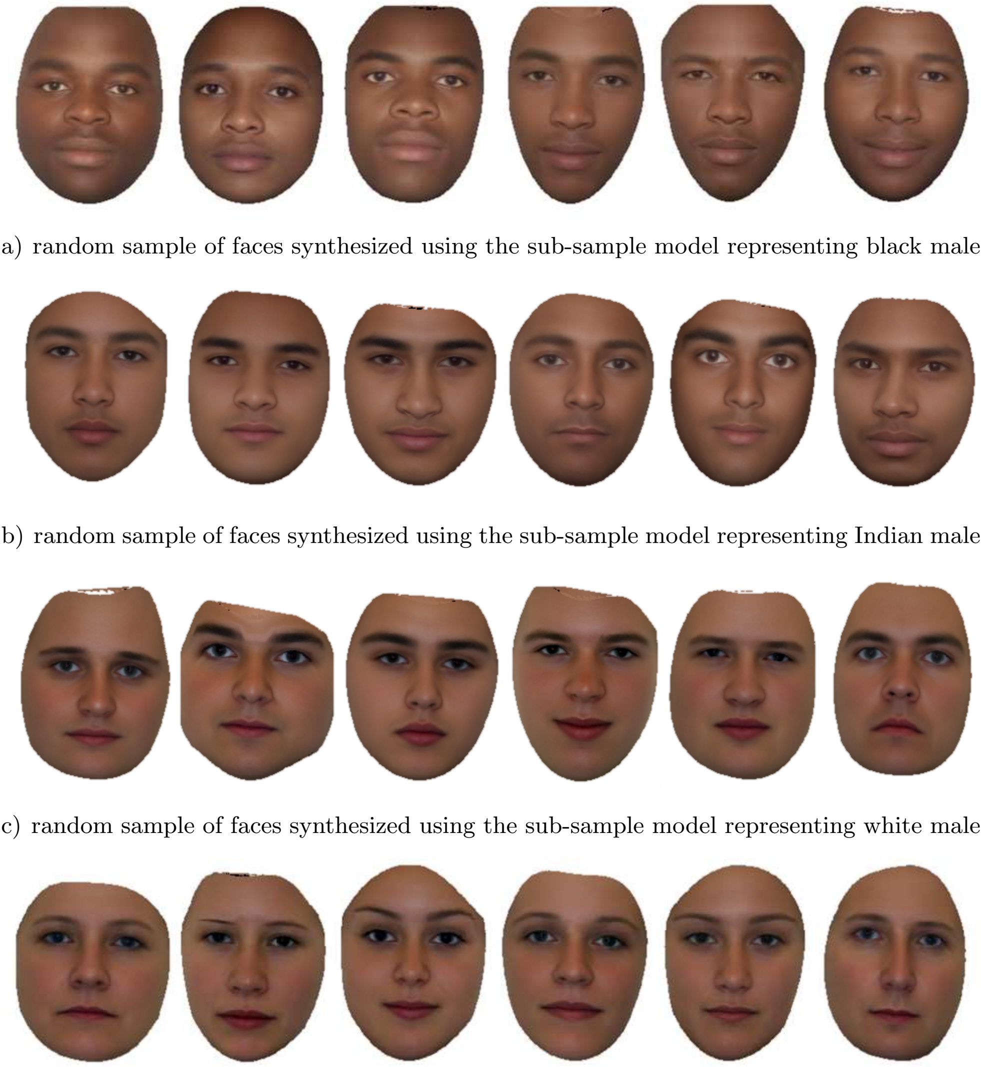
$$\mathbf{c} = \quad \mathbf{L}\mathbf{z} + \mu \tag{3.14}$$

where **L** can be constructed from the chosen demographic with $\mathbf{L} = \mathbf{L}_{BM}$ etc (in which the subscript refers to the demographic group). Equation 3.14 can be used to generate faces with the desired demographic properties by following the reconstruction process described in Chapter 2, by equations 2.66-2.69. Examples of faces synthesized from different demographic sub-sample models are presented in figure 3.9.

3.5 Evolving faces using an evolutionary algorithm

The appearance model described in section 3.4 provides the means for synthesizing plausible face stimuli (face images). An adequate approximation to any face can be obtained from an appropriate selection of 60 independent appearance model parameters, contained in the vector **c**. In principle, the parameter values could be determined using many different approaches. For instance, a naive approach would be to construct a user interface in which each parameter value is controlled by a slider, and the resulting composite face displayed to the witness. Another approach would be to implement a purely random search of the parameter space in which a large number of candidate faces are synthesized from which the best likeness to the target face is selected. Although, in theory, both of these methods are capable of achieving a likeness, neither method takes into account 'ease of use' or the time required to achieve a likeness. This section describes an efficient stochastic search procedure that enables the witness to determine an optimal vector of appearance parameters from which a likeness to the target face can be constructed. A more detailed account of the procedure described here is provided by Pallares-Bejarano [69].

It is crucial to recognize that the optimum search procedure for this task must be an algorithm that is a suitable compromise between *human usability* and *speed of convergence* (i.e. the required number of faces seen and rated by the user before a satisfactory composite is achieved). Evolutionary algorithms can be easily adapted to accommodate different types of input from a user,



(a) random sample of faces synthesized using the sub-sample model representing black males

(b) random sample of faces synthesized using the sub-sample model representing Indian males

(c) random sample of faces synthesized using the sub-sample model representing white males

(d) random sample of faces synthesized using the sub-sample model representing white females

Figure 3.9: Random faces synthesized from the appearance model parameters obtained using PRNG from separate multivariate normal distributions.

and hence are well suited to optimization problems of this type. Accordingly, three evolutionary approaches to conducting the parameter search have been explored, each of which required a different input from the user. The algorithm that offered the best compromise between speed of convergence and usability was employed in the EigenFIT facial composite system.

3.5.1 Fitness function and convergence criteria

In an evolutionary algorithm, a scalar valued fitness function, $f(\mathbf{c})$ quantifies the 'goodness' of a candidate solution, $\mathbf{c}$, to the optimization problem of interest. Each candidate solution is also referred to as a genotype (analogous to a chromosome of biological genetic code) which maps to a phenotype embodying the physical characteristics dictated by the chromosome.

In the context of facial composite applications, a genotype is a vector of appearance model parameters and a phenotype is a face image constructed from the genotype. In the process of evolving a face, the *witness* is required to assign fitness scores to phenotypes. Hence, the mathematical form of the fitness function $f(\mathbf{c})$ is unknown and exists only in the subconscious mind of the witness. The fitness function will differ according to the specific target face of interest and is also dependent on the witness' memory of the suspect. Furthermore, perceptual measures of similarity between faces differ slightly between individuals. Each witness will thus encode a face differently. For these reasons no analytical solution to $f(\mathbf{c})$ exists and a stochastic search procedure is required.

Virtual witness

The behavior of evolutionary algorithms can be strongly affected by a number of parameters such as probability of crossover and mutation, selection method and genotype length. For the application considered in this thesis, the most reliable method for establishing the best algorithm is to perform extensive trials involving human participants, performing the role of real witnesses. However, evaluations involving human participants are both time-consuming and costly. Hence the three evolutionary algorithms studied were evaluated using a *virtual witness* program in which the computer simulated the role of the human witness. The important difference between a virtual witness evaluation/trial and a human operator, is the use of a quantifiable distance metric between solutions. For a *virtual trial* each fitness score was assigned according the distance between the parameter vectors of candidate solution, $\mathbf{c}$ and the parameters $\mathbf{c}_s$ belonging to a target face. The distance metric used was the Mahalanobis distance.

$$f\left(\mathbf{c}\right)=\sqrt{\left(\mathbf{c}-\mathbf{c}_s\right)\Sigma_c\left(\mathbf{c}-\mathbf{c}_s\right)^T} \tag{3.15}$$

For each algorithm, an evaluation was conducted by selecting at random a target genotype and an initial population of random genotypes that constitute a starting point for the EA. For humans, the process of assigning fitness scores, pertaining to similarity between faces, is ambiguous resulting in inconsistent ratings. The virtual witness has been designed to simulate this ambiguity by adding a random perturbation, α to the fitness score (where α is in the order of a few percent of $f\left(\mathbf{c}\right)$, see Pallares-Bejarano for details). Hence, equation 3.15

$$\hat{f}\left(\mathbf{c}\right)=\sqrt{\left(\mathbf{c}-\mathbf{c}_s\right)\Sigma_c\left(\mathbf{c}-\mathbf{c}_s\right)^T}+\alpha \tag{3.16}$$

The aim of facial composite systems in general is to generate a likeness that is sufficient for recognition rather recovering the exact parameter values of the target. Hence, the evolutionary algorithm was judged to have converged when the Mahalanobis distance to the exact parameters values satisfied,

$$\hat{f}\left(\mathbf{c}\right)\leq 3 \tag{3.17}$$

3.5.2 Determining an appropriate algorithm

Initially, three different algorithms were designed. A quantitative measure of their respective performances was determined using the virtual witness, and a qualitative measure of their performances according to a human user was obtained. A brief description of these algorithms is provided below and their key properties are summarised in table 3.5.

Full scale rating algorithm (FSR)

In this approach an elitist genetic algorithm was employed, applying the operations of selection, crossover and mutation to the elite individual of the population (*the stallion*) and one other individual chosen according to the *fitness proportional rating* principle. At each iteration, the offspring produced were rated on a simple numerical scale of 0-10 by the user and virtual witness for their perceived similarity to the target face. To encourage consistency and avoid fitness scaling problems which might induce premature convergence to an incorrect solution, the current stallion (the best likeness generated so far)

	FSR	FTL	SMM
mutation	yes	yes	yes
crossover	yes	no	no
selection method	elitist	elitist	elitist
mutation rate	static	static	dynamic
population size	10-20 (static)	2 (static)	9 (static)
coding method	5 bit binary	real (double point)	real (double point)

Table 3.5: A Summary of the properties of the three exploratory evolutionary algorithms.

and its assigned score was made visible at all times to the user. The process of assigning numerical ratings to the faces in each generation, and replacing the stallion as appropriate, was continued until the convergence criteria had been satisfied (equation 3.17).

Follow the leader algorithm (FTL)

This strategy was the easiest algorithm for the human operator to use. At each step of the iterative process, a single new face was displayed alongside the current best likeness (*the stallion*) and the user was simply asked to select the fittest of the two faces. In this *non-elitist* strategy the current stallion was either retained or replaced by the other individual and used as the stallion in a new generation. The second individual in the new generation was obtained by breeding the stallion with a new individual. In this EA the recent evolutionary history was a factor in determining future offspring. For instance, the recent evolutionary history may suggest that the process is following a well defined direction (as opposed to a totally random path) through the search space. If so, a preference was made for this direction at subsequent iterations, accelerating the search process along a more efficient path through the appearance space and reducing the number of iterations required for convergence.

Select Multiply and Mutate Algorithm (SMM)

In a similar fashion to the FSR algorithm, this algorithm also employed an elitist strategy. However, in this case, an array of faces (typically nine faces per an array) were presented at each iteration, from which the user was required to *select* the best likeness to the target face. The selected face was then cloned (*multiplied*) a number of times and all but one was randomly mutated to produce a new generation that included the stallion. The SMM process was repeated for each of the following generations until a likeness was achieved.

3.5.3 Refined select multiply and mutate algorithm (SMM)

The exploratory study indicated that the SMM algorithm offered a better compromise between numerical speed-of-convergence and cognitive simplicity than the FSR and FTL algorithms. Further modifications to the SMM algorithm were made, thereby improving its performance. A dynamic mutation rate was introduced that provided a faster and more robust convergence to the target. If t denotes the number of generations that have occured since the start of the evolutionary process, then let $p\left(t\right)$ be the probability that an appearance parameter/decision variable of genotype in the current generation will mutate. The relationship between t and $p\left(t\right)$ for 60 appearance model parameters was determined by simulating the composite construction process many times using the virtual witness as a fitness scoring mechanism. Equation 3.18 provided a high mutation rate at the early stages of the evolutionary process, allowing the whole search space to be investigated.

$$p\left(t\right) = 0.100 + 0.417t^{-0.558} \tag{3.18}$$

As the number of generations increased the probability of mutation decreased allowing a local optimum in the search space, and hence a likeness to a target face, to be determined more easily. The basic *SMM* algorithm is depicted in the flow chart in figure 3.12.

3.6 SMM composite construction method

In section 3.2 the key steps for the operation of a general composite system based on a whole face, evolutionary approach were presented. Here a more detailed description is provided that is specific to the SMM algorithm outlined in the previous section. To avoid confusion the term *decision variable* is used when referring to an element of the genotype, and the term *appearance model parameter* is used in the context of the mapped variables from which faces are rendered.

1. The process is initialized by using a PRNG to obtain nine vectors each containing 60 double precision random numbers (decision variables) drawn from a standard normal distribution (see figure 3.11a)

$$\tilde{\mathbf{c}} = N\left(\mathbf{0}, \mathbf{I}\right) \tag{3.19}$$

Each of the nine vectors constitutes a single *genotype*, representing an encoded face. A decoded face image is termed a *phenotype* (see figure 3.10 for an example of phenotype and its corresponding genotype). Collectively, the nine genotypes are referred to as the *initial population.* The purpose of the initial population is to seed the evolutionary algorithm, thereby providing a starting point from which a likeness to a target face can be evolved.

2. A transformation is applied to each genotype vector. The transformation maps the standard, normal decision variables to appearance model parameters that follow the multivariate normal distribution relating to the chosen demographic group (see section 3.4.5)

$$\tilde{\mathbf{c}} \rightarrow \mathbf{c} = N\left(\bar{\mathbf{c}}, \Sigma_c\right) \tag{3.20}$$

3. From each of the appearance parameter vectors, a face image is constructed as described in section 2.4.1 by equations 2.66-2.69. Figure 3.11b illustrates nine phenotype faces images, rendered for display.

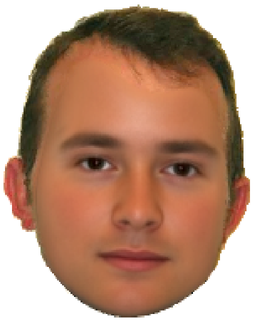
4. From the array of nine faces, the *witness* is required to select the *single* face that most closely resembles the suspect (see figure 3.11b). The selected face is the *fittest* phenotype, also referred to here as the *stallion.* It is the only face in the current generation from which genetic code is propagated into the next generation.

5. The genotype corresponding to the stallion is duplicated or *cloned* nine times (figure 3.11c), thereby, copying the genetic code of the selected face into a new *generation* of nine faces.

6. Eight of the cloned genotypes are *mutated* to produce variations on the selected stallion image. The remaining clone is left unaltered and is positioned randomly in the new array of nine faces. From these genotypes nine new phenotypes are constructed. Thus a new generation of faces is produced.

7. Steps are repeated until an acceptable likeness to the suspect's face is achieved.

In figure 3.12, the key steps in the SMM algorithm are outlined.



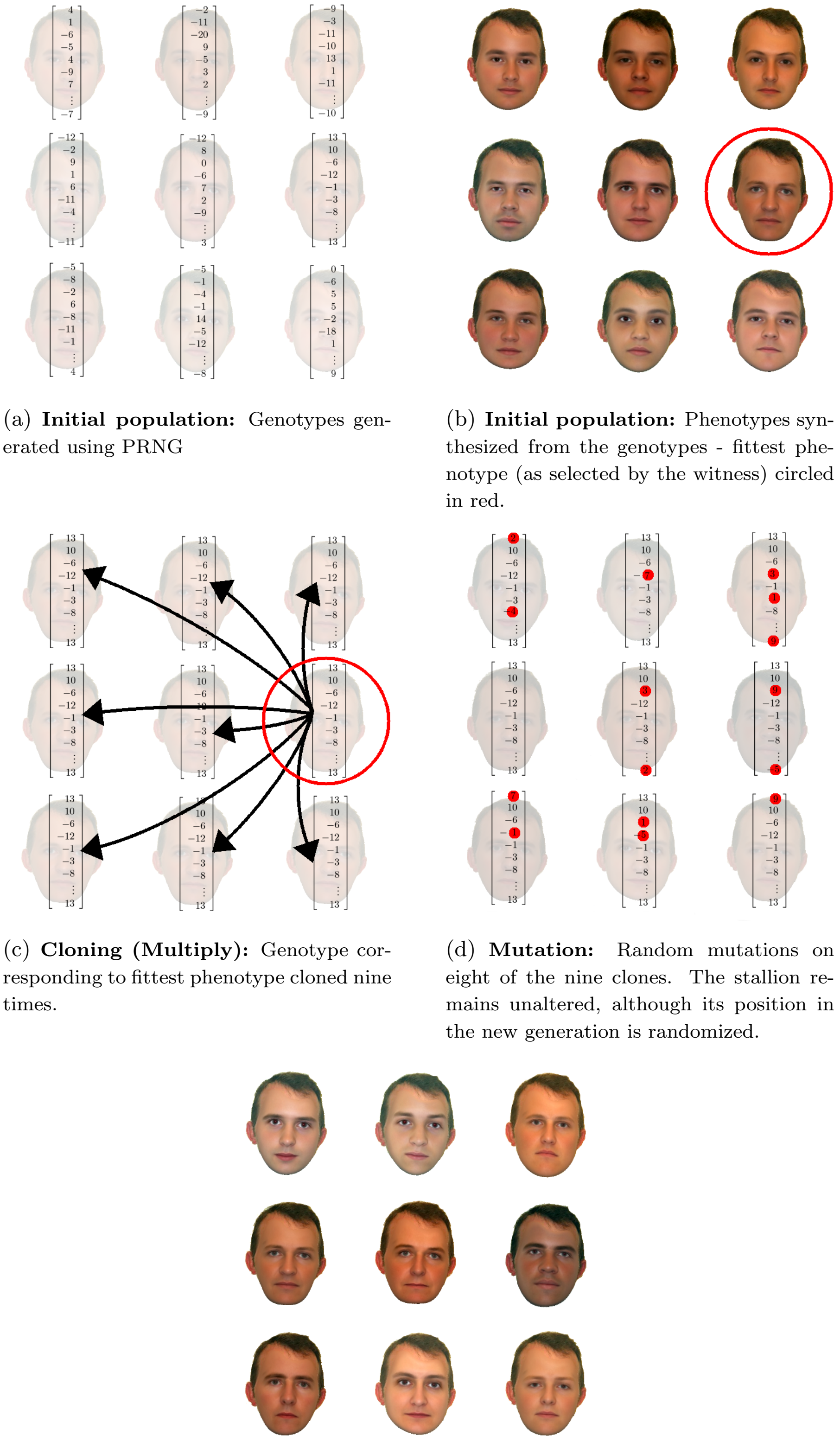
$$\begin{bmatrix} 4 \\ 1 \\ -6 \\ -5 \\ 4 \\ -9 \\ 7 \\ \vdots \\ -7 \end{bmatrix}$$

Figure 3.10: Phenotype face image and corresponding genotype.

3.7 Applying a hairstyle to a composite image

Many researchers use the term 'face recognition' in a very loose sense, referring not only to the face but also the hair and shoulders. However, automated face recognition systems that incorporate extraneous information such as hairstyles may be fundamentally flawed. Liao et al [63] performed experiments which indicated that the non-face regions of the head image, such as hair, dominated the face recognition process. A similar effect is observed when humans attempt to recognise faces, especially if the subject is unfamiliar to the observer. Given the importance of hair in recognition tasks, care must be taken to provide adequate means for applying hairstyles in any practicable facial composite system.

Unlike the face region, hair cannot be modeled using a PCA. The inherent randomness of hair, and lack of identifiable features, make it impossible to identify correspondences between different hairstyles. A better method for reliably capturing the shape and textural properties of hair is to simply duplicate the hairstyles of the subjects that constitute the training set. These hairstyles can then be placed over the composite face to achieve the desired result. Using the hair of the training subjects is guaranteed to give photo-realistic hairstyles that can be chosen independently of the face. The difficulty arises when attempting to blend a chosen hairstyle to the target face. Variations in hair length result in different regions of the face being obscured depending on the selected style. Hence, the position of the join between hairstyle and face is seldom in the same place from hairstyle to hairstyle. Disparities in skin pigmentation between the donor and target faces can also prove problematic when attempting to construct a seamless join. These issues make the task of applying a hairstyle to the composite face one of the most complex issues concerning composite construction using the PCA method. A simple blending procedure for applying hairstyle(s)



(a) **Initial population:** Genotypes generated using PRNG

(b) **Initial population:** Phenotypes synthesized from the genotypes - fittest phenotype (as selected by the witness) circled in red.

(c) **Cloning (Multiply):** Genotype corresponding to fittest phenotype cloned nine times.

(d) **Mutation:** Random mutations on eight of the nine clones. The stallion remains unaltered, although its position in the new generation is randomized.

(e) **New generation:** Based on mutated genetic material from selected face.

Figure 3.11: Schematic representation of processes that result in a generation of phenotype face images. The procedure is initialized by forming nine strings of random numbers (genotypes) using a pseudo-random number generator (PRNG). For each of the nine genotypes a corresponding phenotype face image is constructed. The witness is required to select (a subjective selection) the phenotype image that exhibits the closest likeness to the suspect. The genotype of the selected face is used as the genetic basis for a new generation of faces.

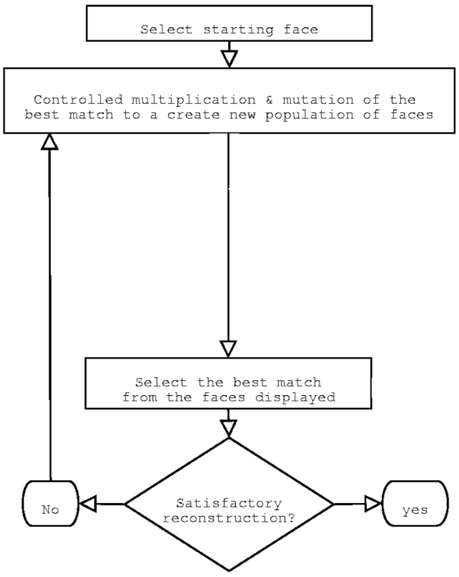

Select starting face
Controlled multiplication & mutation of the best match to a create new population of faces
Select the best match from the faces displayed
Satisfactory reconstruction?
No
yes

Figure 3.12: *SMM* Algorithm flowchart indicating the steps required for producing a composite using the evolutionary approach.

to a face images is described in the next section followed by a preliminary investigation into a more sophisticated multi-resolution approach.

3.7.1 Blending procedure

A method for applying a hairstyle to a face was employed in which the face *textures* and face *shapes* were blended, providing a computationally efficient and aesthetically pleasing solution to the problem. Hairstyles were taken from subjects comprising the training set. Thus the hairstyles were captured under the same environmental conditions as the face data, eliminating any potential issues due to inconsistent lighting. The term *donor* will be used when referring to a subject who provides the hairstyle and the word *target*[2] will be used to describe the face to which hair must be added. The key steps of the approach taken are outlined in the tree diagram presented in figure 3.13 and a description of the process is provided in algorithm 2.

Algorithm 2 Hairstyle mapping method

1. Perform a rigid shape alignment of the $2(m+k) \times 1$ *donor shape vector* $\mathbf{s}_d$ with the $2m \times 1$ *target shape vector* $\mathbf{s}_t$. Note that $\mathbf{s}_d$ contains k more

[2] Note that the word target is also used when referring to the face for which a likeness is required

coordinate pairs than $\mathbf{s}_t$ due to the additional landmarks required to delineate the perimeter of the hairstyle. The alignment transformation is calculated on the $2m$ *face coordinates only* but is applied to all $2\left(m+k\right)$ elements of the vector $\mathbf{s}_d$.

2. Pad $\mathbf{s}_t$ with zeros corresponding to hair landmarks in $\mathbf{s}_t$ such that the donor and target shape vectors contain the same number of elements.

$$\mathbf{s}_t \rightarrow \left[\frac{\mathbf{s}_t}{\mathbf{0}}\right] \tag{3.21}$$

3. Spline donor and target shapes,

$$\mathbf{s}_t' = \left[\frac{\mathbf{s}_t}{\mathbf{0}}\right] \circ \mathbf{m}_s + \mathbf{s}_d \circ \left[\mathbf{1} - \mathbf{m}_s\right] \tag{3.22}$$

where $\circ$ represents a point-wise multiplication and $\mathbf{m}_s$ is a $2\left(m+k\right) \times 1$ weighting vector with values in the range $[0, 1]$ such that,

$$\mathbf{m}_s\left(i\right) = \begin{cases} 0 & if \quad i \in L_{hair} \; and \; i \notin L_{face} \\ 0 < m_s(i) < 1 & if \quad i \in \; L_{face} \cap L_{hair} \\ 1 & if \quad i \in L_{face} \; and \; i \notin L_{hair} \end{cases} \tag{3.23}$$

With L_{hair} denoting the set of indices representing *hair landmarks* and L_{face} the set of indices representing *hair landmarks.* The symbol '$\circ$' represents a point-wise multiplication and $\mathbf{1}$ is a $2\left(m+k\right)$ element column vector containing ones.

4. Define a mask image I_M such that all pixels whose coordinates lie within the convex hull of $\left\{\mathbf{s}_t\left(i\right)' : i \in L_{face}\right\}$ are set to one, and all other pixels are equal to zero. Soften the transition between light and dark pixel values by applying a 15×15 averaging filter to I_M,

$$I_M'\left(x, y\right) = \sum_{i=-k/2}^{k/2} \sum_{i=-k/2}^{k/2} I_M\left(x+i, y+j\right) h\left(i+k/2, j+k/2\right) \tag{3.24}$$

where $k = 15$ and h is the convolution kernel. For convenience, the filtered mask image I_M will also be represented by the matrix $\mathbf{I}_M$. Elements of $\mathbf{I}_M$

that lie in the range $(0, 1)$ form a feathered edge around the face region, which is refered to by Burt and Adelson [15] as the *transition zone*.

5. Let $\mathbf{I}_d$ and $\mathbf{I}_t$ be the matrix representations of the donor and target textures respectively. To blend these images successfully a correspondence is needed between both images. This is achieved by *warping* to the splined/blended face shape, $\mathbf{s}'_t$. Let $\mathbf{I}'_d$ represent the warped donor image and $\mathbf{I}'_t$ represent the warped target image.

6. At this stage some form of photometric correction may be required so that the skin pigmentation of $\mathbf{I}'_t$ matches the skin pigmentation of $\mathbf{I}'_d$ in the transition zone. Treating each colour plane separately, the parameters s (sample standard deviation of pixel values) and $\bar{p}$ (sample mean of pixel values) of the Gaussian pdf of pixel intensities $\{p_i\}$ contained within face region is calculated. The photometric correction is implemented by transforming the pdf of the donor such that it has the same parameters as the target,

$$p_d\,(i)' = (p_d - \bar{p}_d)\,\frac{s_t}{s_d} + \bar{p}_t \tag{3.25}$$

7. The mask image $\mathbf{I}_M$ defines a weighted combination ($\mathbf{I}'_{splined}$) of the warped target and donor images,

$$\mathbf{I}'_{splined} = \mathbf{I}'_t \circ \mathbf{I}_M + \mathbf{I}'_d \circ [\mathbf{1}_{m,n} - \mathbf{I}_M] \tag{3.26}$$

3.7.2 Towards an improved blending procedure

The method outlined in section 3.7.1 for applying a hairstyle to a target face results in a join between image patches which is to some degree visible in the composite image. In most instances, the visibility of the join is reduced by photometric correction and the use of a smoothing transition zone. For examples in which the join remains prominent, a more sophisticated approach to the blending problem is required. One approach that has the potential to overcome the blending issue is to use a multi-resolution spline to join the image patches. Multi-resolution splines as described by Burt and Adelson [15] have previously been used for image mosaic applications. The following section explores the

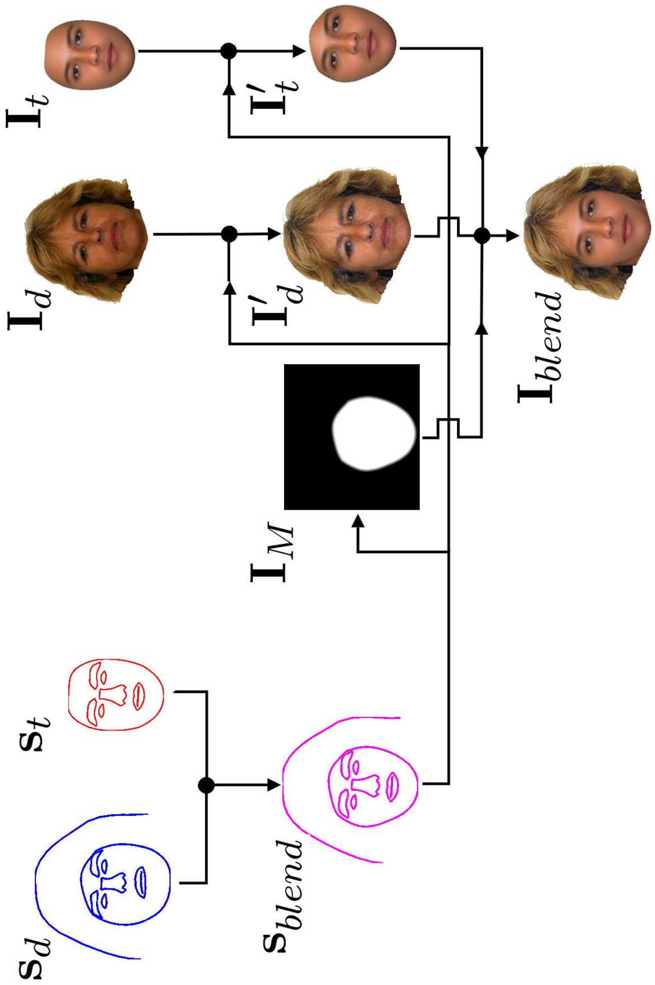


Figure 3.13: Steps required for applying a hairstyle to a composite face. The process involves: 1) Blending the donor shape $\mathbf{s}_d$ and the target face shape $\mathbf{s}_t$. 2) Warping the both the donor and target textures to blended face shape, $\mathbf{s}_{blen}$, thereby achieving pixel-wise correspondences. 3) Forming a weighted combination (a spline) of the warped images ($\mathbf{I}'_d$ and $\mathbf{I}'_t$) according to the pixel intensity values contained in the mask image, $\mathbf{I}_M$, to form the output image, $\mathbf{I}_{blend}$.

practicality of using a multi-resolution spline for mapping hairstyles. A brief mathematical explanation of the technique is given (for a full explanation the reader should refer to [15]). The results of some simple blending examples are provided and the suitability of the spline method for composites is discussed.

Acceptable results are obtained using the method outlined in section 3.7.1 when the transition zone is similar in size to the wavelengths present in the images. In practice, this criteria is seldom met since, in general, images contain a range of spatial frequencies. If the transition zone over which the mask image $\mathbf{I}_M$ changes from 0 to 1 is small compared to the image structures contained in $\mathbf{I}'_t$ and $\mathbf{I}'_d$, the boundary may still appear as a step in image intensity. Conversely, if the transition zone is large compared to the image structures, structures from both images may appear superimposed within the transition zone, producing undesirable double exposure type artefacts. A multiresolution spline provides a solution to the *transition zone/wavelength* mismatch problem by decomposing both $\mathbf{I}'_t$ and $\mathbf{I}'_d$ into bandpass images. Each pair of bandpass images can then be combined using a weighting image in which the transition zone is matched to the size of the image structures present. Algorithm 3 provides an overview of the procedure for blending the textures using a multi-resolution spline.

Algorithm 3 Multi-resolution method for image splines

1. **Gaussian pyramid construction:** Apply a Gaussian low pass [5x5] filter to the warped target image $\mathbf{I}'_t$, forming a smoothed version of the image. Sample every other pixel from the smoothed image, thereby reducing its area to a 1/4 of its original size. The filtering procedure and sampling procedure are collectively known as a *REDUCE* operation - see equation 3.27. Label the image that results from applying the reduce procedure to $\mathbf{I}'_t$ as G_1. Now perform a reduce operation on G_1, obtaining a new image G_2 which is a 1/4 the size of G_1 and 1/16 the size of $\mathbf{I}'_t$. By successive repetitions of the *REDUCE* process, a set of images that decrease both in size and in high frequency content is formed $\{G_0, G_1, \ldots, G_N\}$ (where $G_0 \equiv \mathbf{I}_t$). Stacking this set of images in order of decreasing size yields a tapering data structure known as a *Gaussian pyramid.*

$$G_l\left(i,j\right) = \sum \sum_{m,n=1}^{5} w\left(m,n\right) G_{l-1}\left(2i+m, 2j+n\right) \tag{3.27}$$

where $w\left(m,n\right)$ is the generating kernel and l indicates the level in the pyramid structure. In the same manner, Gaussian pyramids are con-

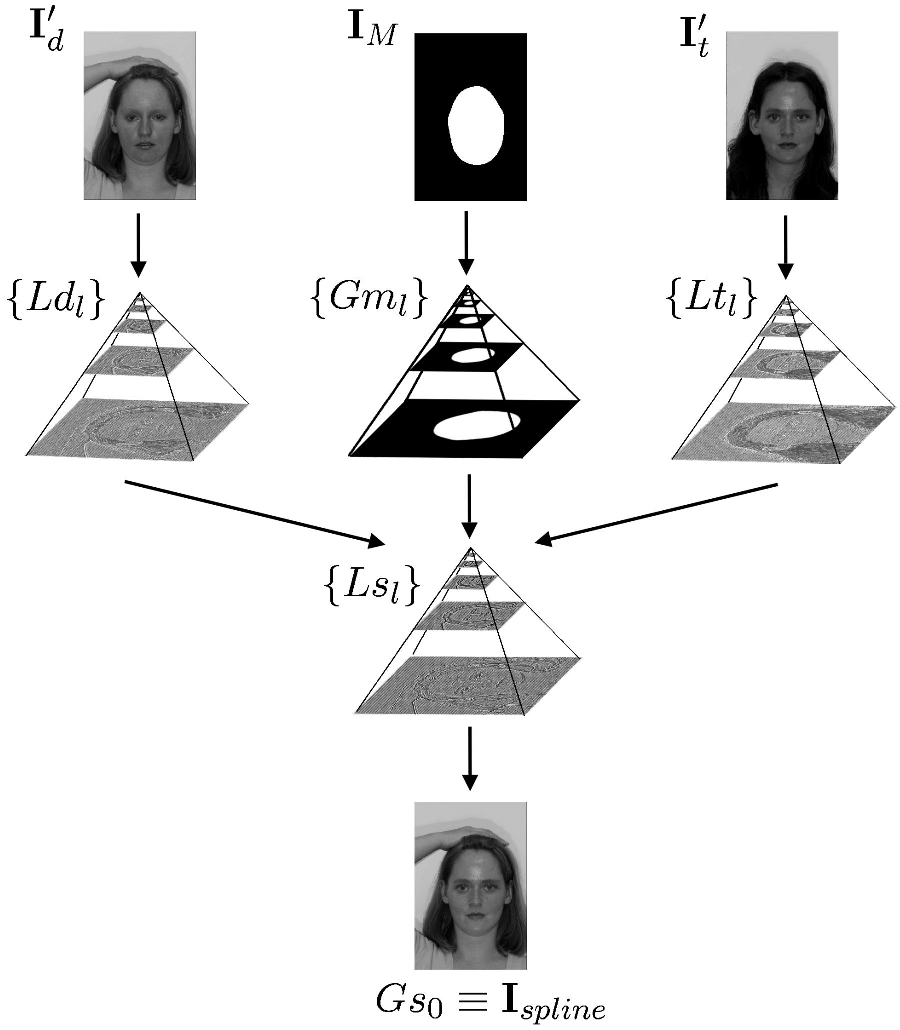


Figure 3.14: The multiresolution spline forms a weighted blend of images that preserves high spatial frequencies in the transition zone. Donor and target images are decomposed into bandpass pyramid structures $\{Ld_l\}$ and $\{Lt_l\}$ respectively). $\{Ls_l\}$ represents the Laplacian pyramid that is constructed by the weighted combination of $\{Ld_l\}$) and $\{Lt_l\}$, as specified by the Gaussian pyramid $\{Gm_l\}$.

structed for the image mask $\mathbf{I}_M$ and warped donor image $\mathbf{I}'_d$. Forming a Gaussian pyramid is equivalent to, but computationally more efficient than, generating a stack of images which are identical in size but smoothed with increasing large filter kernels.

2. **Laplacian pyramid construction:** For the donor and target Gaussian pyramids only, calculate the difference between consecutive images in the in the pyramid, thereby forming a pyramid of *bandpass images* $\{L_0, L_2, \ldots, L_N\}$ as illustrated in figure 3.14. The difference between two appropriate Gaussian filters approximates a Laplacian filter. Because these arrays differ in sample density, it is necessary to interpolate new samples between those of a given array (image) before it is subtracted from the next lowest array. Interpolation can be achieved by reversing the *REDUCE* process. Burt and Adelson refer to this as an *EXPAND* operation.

$$L_l = G_l - EXPAND\left(G_{l+1}\right) \tag{3.28}$$

The process of expanding G_l k times is written as,

$$G_{l,k}\left(i,j\right) = 4\sum\sum_{n,m=-2}^{2} w\left(m,n\right) G_{l,k-1}\left(\frac{2i+m}{2}, \frac{2j+n}{2}\right) \tag{3.29}$$

Only terms for which $\left(i-m\right)/2$ and $\left(j-n\right)/2$ are integers are included in this sum.

3. **Image spline and pyramid expansion:** Let Ls_l and Gs_l denote the splined l^{th} level bandpass detail image and low pass approximation image respectively. Starting at the top of the Laplacian pyramids, spline the l^{th} level bandpass image Ld_l with corresponding image Lt_l using the weighting mask Gm_l to form a blended bandpass image Ls_l (see equation 3.30).

$$Ls_l\left(i,j\right) = Gm_l\left(i,j\right) Lt_l\left(i,j\right) + \left(\mathbf{1} - Gm_l\left(i,j\right)\right) Ld_l\left(i,j\right) \tag{3.30}$$

Use the expand operation to resize the splined image, making it the same size as the $l^{th}+1$ level images. Spline the $l^{th}+1$ level images and add them to Ls_l. Repeat this process, working down the pyramid structure, eventually resulting in the output image I_s'. The whole recursive procedure can be expressed as nested expand operations,

$$\begin{aligned} Gs_0 = Ls_0 + EXPAND\,(Ls_1 + \\ EXPAND\,(\ldots Ls_{N-1} + EXPAND\,(Gs_N))) \end{aligned} \tag{3.31}$$

Because there is no higher level array to subtract from Gs_N, we define $Ls_N = Gs_N$.

To determine the suitability of the multi-resolution spline for applying hairstyles in the EigenFIT composite system it was compared to the simpler blending method described in section 3.7.1. Using the input images in table 3.6, the following four scenarios were considered,

1. A weighted sum of the donor image (a) and target image (b) was formed using the pixel weights in mask image (d) - note that in this case a low pass filter was first applied to image (d) to create a transition zone in which pixel intensities varied in the range $(0, 1)$.

2. The multi-resolution spline method was used to map the donor hairstyle in image (a) to the target face in the continuous image, (b).

3. The multi-resolution spline method was used to map the donor hairstyle in image (a) to the target face patch in image (c).

4. In the final experiment the face patch (c) was combined with the donor image (a) using a simple weighted sum.

The output images for each scenario are displayed in table 3.7. For the case in which two continuous images were to be blended the multi-resolution spline provided superior results. However, when a hairstyle was applied to a *face patch* (as is necessary for the EigenFIT system), the multi-resolution method failed to produce a satisfactory blend. The reason for this failure is due to the discontinuity in image intensity in the target image, making the multi-resolution spline method presented in this section unsuitable for the task of applying hairstyles in the EigenFIT composite system.

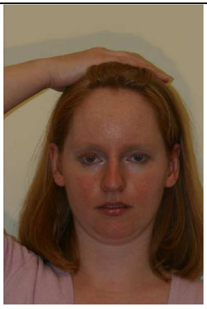
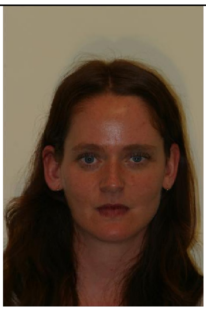
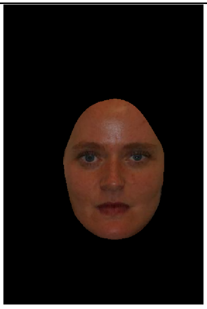
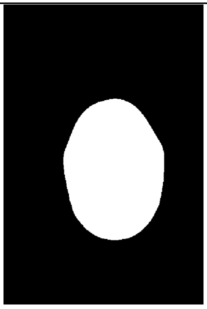
Input images			
			
a) Donor	b) Target	c) Target patch	d) Mask

Table 3.6: Input images for different hair mapping scenarios.

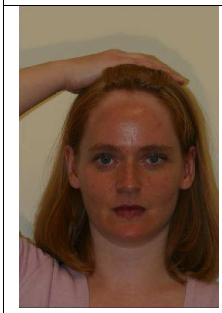
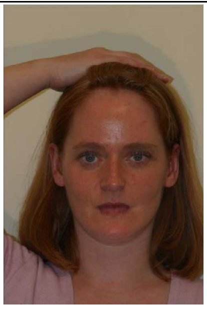
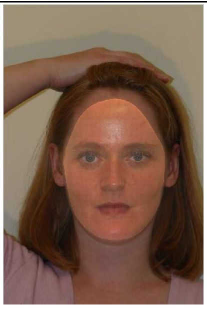
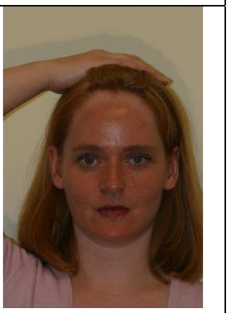
Blended output images			
			
1) Simple blend	2) Multi-resolution spline	3) Multi-resolution spline using target patch	4) Simple blend using target patch

Table 3.7: Donor hairstyle mapped to target face for the four different scenarios.

3.8 Overriding the evolutionary process

During the process of generating a facial composite, situations may arise in which it is advantageous to intervene in the evolutionary procedure. One such situation occurs when the evolutionary procedure has produced a face in which one or more features exhibit a good likeness to the target face, but the remaining features do not. The problem is that features in the composite face which are highly similar to the target may be degraded during subsequent generations at the expense of improving the 'whole face' likeness. This issue has been addressed by providing a *feature locking tool* that allows the shape of individual facial features of the stallion (best likeness so far) face to be fixed and propagated through future generations.

3.8.1 Locking facial features

Feature locking implementation

The EigenFIT interface has been designed to allow the operator to lock a facial feature by selecting the corresponding region of the schematic face image (see figure 3.3). Once the feature has been locked, it appears highlighted in the schematic image to inform the user that no further shape deformation of the selected feature will occur during subsequent generations. In terms of our facial composite system, we can imagine taking a snap shot of the stallion at certain instances in time and retaining the shape of one or more chosen features. Future generations can only introduce shape changes in the features that remain unlocked. The process is expressed in equation 3.32 by a vector addition comprising the current stallion shape $\mathbf{s}_t$ and a snap shot of a previous stallion $\mathbf{s}_{t_0}$, captured at time t_0. Here, we have used the term time to refer to a particular generation number, with $t > t_0$.

$$\mathbf{s}'_t = \mathbf{s}_t \left[\mathbf{I} - \mathbf{W}_f\right] + \mathbf{s}_{t_0}\mathbf{W}_f \tag{3.32}$$

Where $\mathbf{I}$ is the identity matrix and $\mathbf{W}_f$ is a diagonal matrix with elements equal to one or zero. $\mathbf{W}_f$ is referred to as the feature selector since it effectively extracts all of the coordinates from $\mathbf{s}_{t_0}$ corresponding to the fixed feature.

Locking multiple features

We can extend equation 3.32 to include multiple features, locked at different instances (generations).

$$\mathbf{s}'_t = \mathbf{s}_t \left[\mathbf{I} - \mathbf{W}_{f1} - \mathbf{W}_{f2} \ldots - \mathbf{W}_{fn}\right] + \mathbf{s}_{t_1}\mathbf{W}_{f1} + \mathbf{s}_{t_2}\mathbf{W}_{f2} \ldots + \mathbf{s}_{t_k}\mathbf{W}_{fn} \tag{3.33}$$

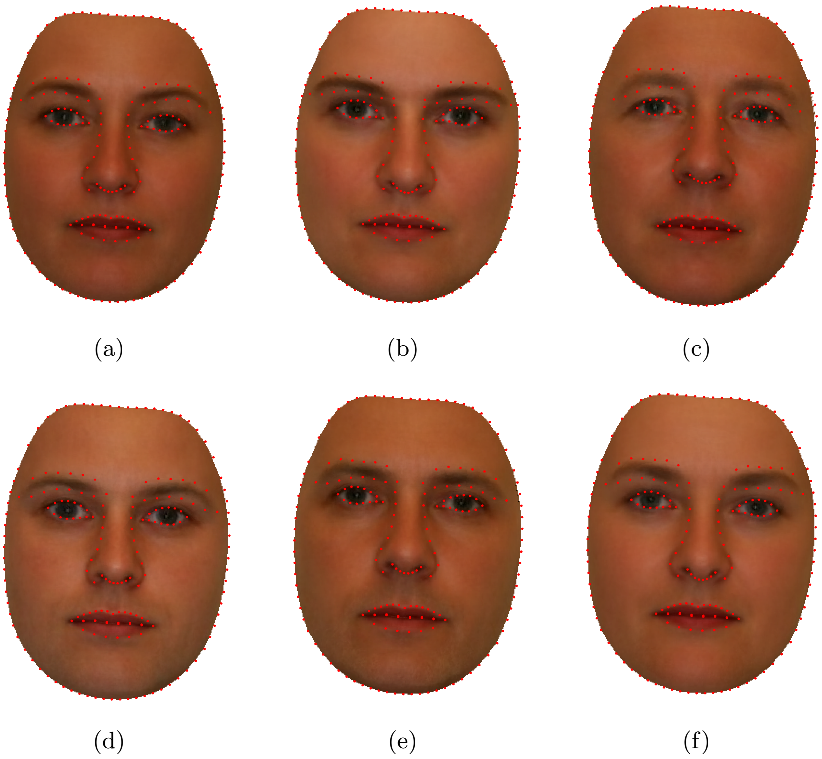
$\mathbf{W}_{f1}, \mathbf{W}_{f2} \ldots$ and $\mathbf{W}_{fn}$ are the feature selectors for the 1^{st}, 2^{nd} and n^{th} features respectively (ie nose, mouth... etc). $\mathbf{s}_{t_1}$, $\mathbf{s}_{t_1}$ and $\mathbf{s}_{t_k}$ are snap shots of the stallion taken at times t_1, t_2 and t_k. Hence one or more features may be locked at once. If the user wishes to evolve a single feature in isolation, all other features can be locked.

Unlocking facial features

When a chosen feature is unlocked we need to re-introduce shape variation so that the feature may evolve as it did prior to locking. This could be achieved by simply reverting to the current stallion such that $\mathbf{s}' = \mathbf{s}_t$. However, this would cause an abrupt change in shape of the unlocked facial feature which is both counter intuitive and visually displeasing. Instead we require a continuous shape transition by gradually re-introducing variation into the unlocked feature. To accomplish this smooth transition we establish a decay function, $\alpha\left(t\right)$ and modify 3.32 as follows,

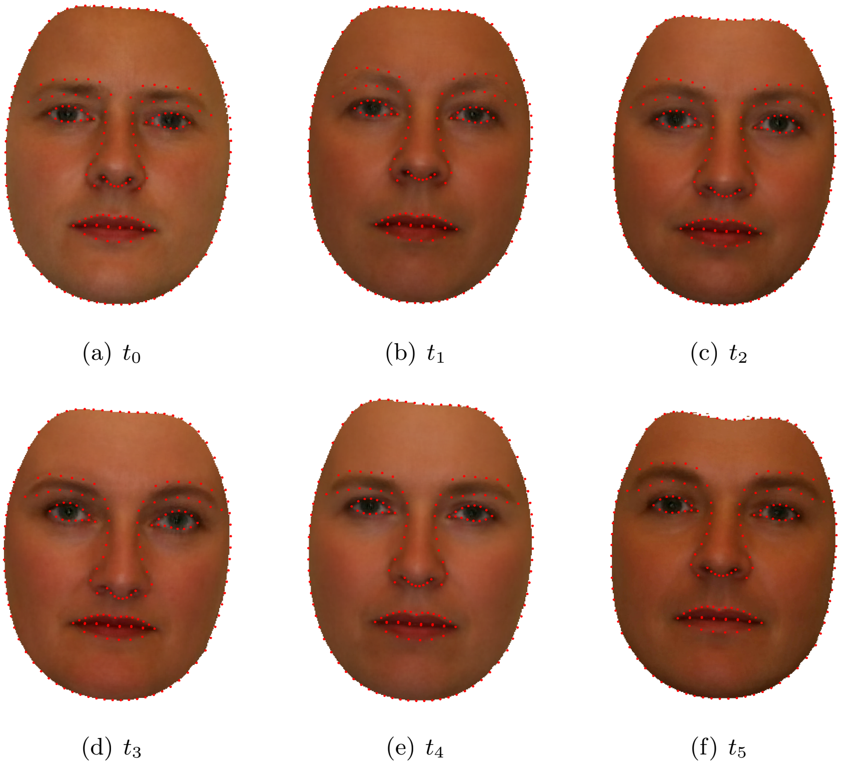
$$\mathbf{s}' = \mathbf{s}_t\left[\mathbf{I} - \alpha\left(t\right)\mathbf{W}_f\right] + \mathbf{s}_{t_0}\alpha\left(t\right)\mathbf{W}_f \tag{3.34}$$

A linear decay function was defined; $\alpha = -0.1t$ in the range $0 < t \leq 10$ that gives an aesthetically pleasing transition between the fixed feature and stallion. An exponential decay may be preferable when a smoother decay is required. For a fixed feature α remains constant $(\alpha = 1)$ and once α has decayed to zero it remains at that value until the feature is fixed again. Equation 3.34 has a nice limiting behaviour. When a feature is unlocked the shape offset ebbs away over time and facial shape as seen by the user $(\mathbf{s}'_t)$ reverts to the underlying stallion $(\mathbf{s}_t)$.


(a) (b) (c)

(d) (e) (f)

Figure 3.15: In images a-f the *mouth shape* and the *perimeter of face* are *locked*, random shape variations evident in other features.



(a) t_0 (b) t_1 (c) t_2

(d) t_3 (e) t_4 (f) t_5

Figure 3.16: Mouth shape and perimeter of face unlocked. The flexibility of the previously locked features increases with t. When $t >> t_0$ the affect of locking a feature decays to zero and the usual evolutionary shape variations resume.

3.9 Summary

In Chapter 3, aspects relating to the construction of EigenFIT in its most basic mode of operation have been described. The chapter began by presenting the argument for a facial composite system that operates in the manner of EigenFIT, which was followed by a brief overview of the system itself. Later sections described the individual components of the system in detail. The first of these components was the user interface which was designed to be cognitively simple to use, thereby allowing the witness to take a more active role in the composite process than has previously been possible.

The specifics of an appearance model of the human face were described, from which new plausible examples of whole-face images could be generated. As well as a basic overview of the appearance model (described in detail in Chapter 2), this section included the acquisition of face data and the probability density function models that allowed plausible new examples to be synthesized.

A likeness to a suspect's face can be generated from an appropriate set of appearance model parameters. The parameter values were determined iteratively using an interactive evolutionary algorithm. Various algorithms were tested and the most promising one (the SMM algorithm) was developed further for inclusion in EigenFIT. The chosen algorithm, and the steps required in its development were described in section 3.5

The ability to apply a hairstyle of choice to a composite image is an essential part of any composite system. Section 3.7 described a simple blending method by which a hairstyle is applied to a composite image within the EigenFIT system. In the subsequent section, a potentially superior method for applying a hairstyle based on a *multi-resolution spline* was investigated. A preliminary experiment, comparing the two methods, demonstrated that the multi-resolution spline was unsuitable for this specific application.

The last section of this chapter described a tool that offered the functionality for locking the shape of one or more selected facial features. This was achieved by adding an appropriate offset vector to the shape of the current stallion. If one of the locked features was subsequently deselected, the corresponding offset vector was allowed to decay to the null vector over a number of generations, and the system would return to its global, evolutionary mode of operation. Hence the lock-feature tool implementation was complementary to the standard evolutionary process.

The core implementation of the EigenFIT system described in Chapter 3 offers a simple and intuitive method for constructing facial composites that may be used directly by the witness under the supervision of a trained operator.

Chapter 4 describes a number of additional tools that offer greater control over the composite process than is afforded by the evolutionary procedure alone.

Chapter 4

EigenFIT - advanced functionality

In the previous chapter, the elements of the EigenFIT system necessary for a predominately evolutionary mode of operation were presented. In that mode of operation, provisionally named EasyFIT, the witness is simply required to make decisions in response to the facial stimuli, displayed on a computer monitor. A method for the limited intervention in the evolutionary procedure via the lock *feature tool* was also described in Chapter 3. The conceptual simplicity of the EasyFIT method is a major strength. However, the changes introduced into the composite in this way are fundamentally random in nature, and additional functionality for allowing a witness/operator to alter the appearance deterministically is desirable. A step by step example illustrating the production of a facial composite using the EigenFIT system is provided in Appendix D.

Although strong evidence has been provided to support the notion that observers achieve face recognition tasks through configurational cues, it is nonetheless common for a witness to remember particular, distinctive features and to offer general, semantic descriptors of the face. In this chapter, we explore means for exploiting this kind of witness information. Integrating *featural* and *semantic* information, with the standard holistic-evolutionary mode of operation, thus aims to provide maximum flexibility to the witness and operator.

4.1 System overview - ExpertFIT mode

Additional functionality is provided through a parallel mode of operation termed *ExpertFIT*. The ExpertFIT mode provides a number of tools which incorporate the following,

- **Blending**: Augmenting 'fit' characteristics from two or more faces in a generation into single, averaged face image (4.3).

- **Facial attribute manipulation**: Learned models of facial attributes enable facial traits to be enhanced or reduced. The age attribute is considered here, although the technique is equally applicable to other attributes such as masculinity and ethnicity (4.4).

- **Local feature manipulation**: The seamless alteration of individual features by translating and scaling the defining feature coordinates and warping the overall texture to the new shape (section 4.5).

- **Applying fine details to a composite**: The functionality for applying fine details such as wrinkles to a composite face[1] (4.6).

Allowing the composite image to be altered in these ways poses an interesting problem with regard to the underlying, numerical representation of the face. Essentially, this relates to the fact that some of the manipulations described above can result in a facial appearance which lies outside the space spanned by the global appearance model. The manner in which this is handled, is an important consideration and an approach to this problem is discussed in detail in section 4.5.1.

4.2 Design of graphical user interface - ExpertFIT

The ExpertFIT mode embodies the same 'simple to use' design principle as EasyFIT mode. ExpertFIT mode is accessed via a drop down menu on the EigenFIT menu bar. All of the EasyFIT functionality is provided, plus some additional tools that are intended to be used by a trained operator. Once the expert mode has been selected, icons located at the top right hand corner of the interface, which were previously greyed out, become active (see figure 4.1 for details).

Below is a brief description of the screen layout for each of the expert tools,

- **Blend**: Vertical sliders allow the operator to form a weighted sum of the faces in the current generation. The blended face appears in the bottom right hand corner. Selecting this image with the mouse replaces the current stallion with the blended image and causes EigenFIT to revert to the main screen.

[1] Note: that although a proof of concept version of this tool has been developed, it remains to be incorporated into the EigenFIT software package

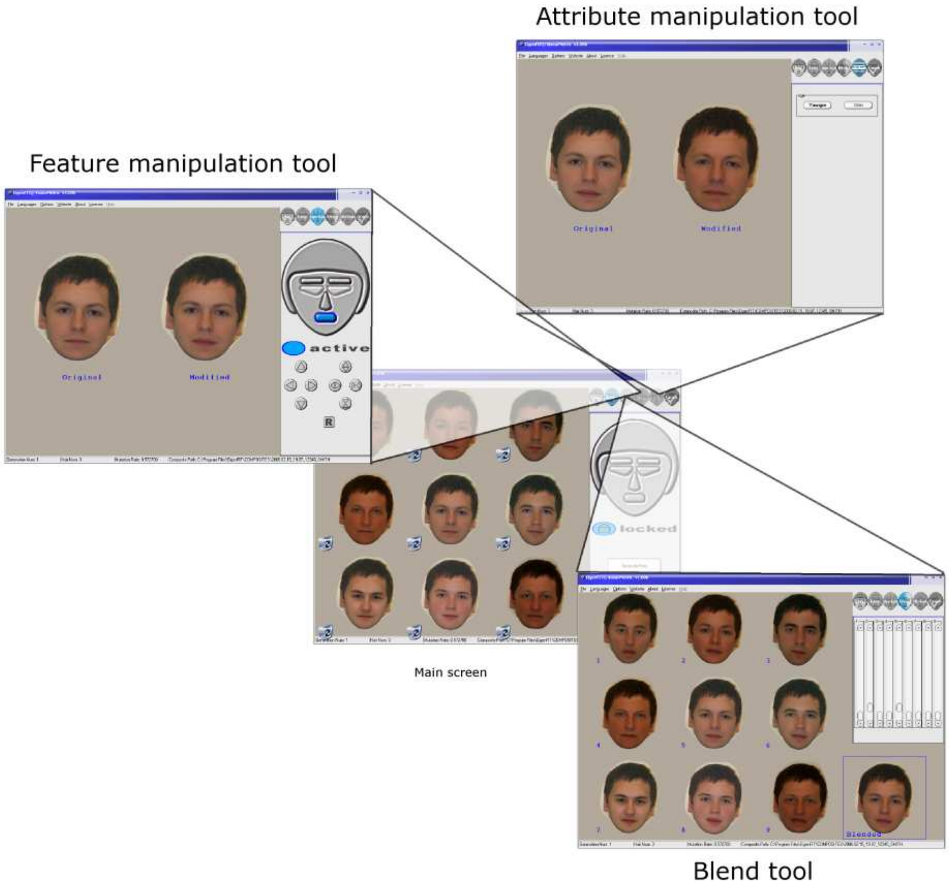


Figure 4.1: The additional functionality provided by the ExpertFIT is made accessible by selecting 'ExpertFIT' from the 'options' menu which enables the icons located in the top right hand corner of the main screen. Selecting an advanced functionality tool changes the screen layout to an appropriate configuration for the chosen tool. EigenFIT reverts to the main screen once the modifications to the stallion (current best likeness) have been approved by the witness.

- **Facial attribute manipulation**: A side by side configuration with the current stallion positioned on the left and a duplicate of the current stallion on the right. Controls in the right hand frame allow the operator to make the duplicate face appear older or younger as appropriate. The side by side layout makes it easy for the witness to observe changes with respect to the current stallion.
- **Local feature manipulation**: A side by side configuration with the current stallion positioned on the left and a duplicate of the current stallion on the right. Position and scaling controls on in the right hand frame allow the operator to modify the shape of one or more features within the duplicate image. The currently selected feature is highlighted in blue on an iconic face, located above the move-scale controls.
- **Applying fine details to a composite (Proposed interface):** The tool invokes a familiar three by three layout of face stimuli. The nine faces are all duplicates of the current stallion, with a different pattern of fine detail applied to each. More examples are displayed using a scroll bar (same arrangement as the hairstyle tool) and the prominence of the fine facial details can be increased or decreased using a slider.

4.3 Blend tool

The simplicity of the specifically designed EA *excludes* the option for carrying over facial characteristics from *more than one* phenotype to the next set of nine faces. The motivation behind the blending procedure is to propagate facial characteristics from more than one face into the following generation. This is achieved by forming a weighted combination of the genotypes (appearance model parameter vectors $\{\mathbf{c}_i\}$) comprising the current generation,

$$\mathbf{c}_s \to \sum_{i=1}^{9} \alpha_i \mathbf{c}_i \quad \textit{with} \quad \sum_{i=1}^{9} \alpha_i = 1 \tag{4.1}$$

Faces for which $\alpha = 0$ do not contribute to the blended face. If the blended face is considered to bear a better likeness to the target than the current stallion, the current stallion is replaced by the blended face. The updated stallion is reconstructed from the appearance model parameter vector $\mathbf{c}_s$ in the usual way. A schematic overview of the blending process is provided in figure 4.2.

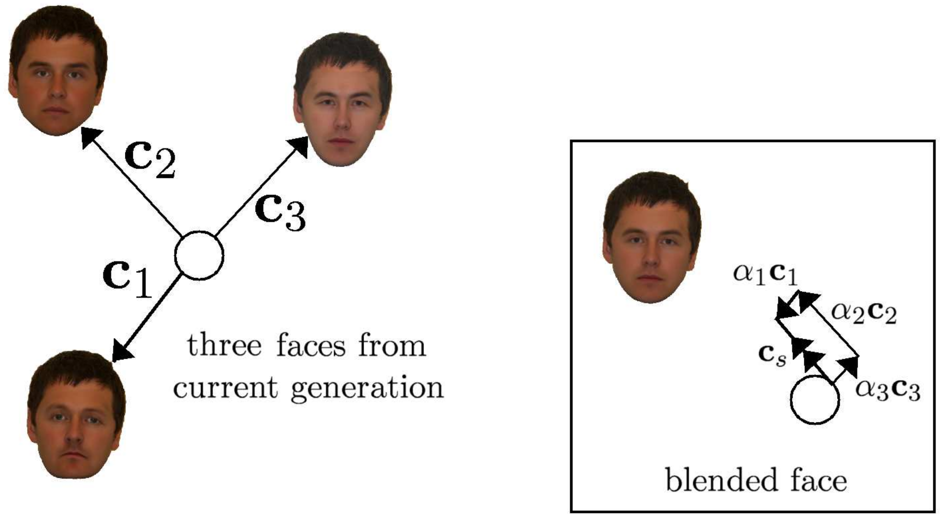


Figure 4.2: A schematic diagram representing the blend process. In this 2d simplification, a linear combination of three faces from the current generation has been formed. The stallion parameter vector c_s is replaced by the linear combination, $\alpha_1\mathbf{c}_1 + \alpha_2\mathbf{c}_2 + \alpha_3\mathbf{c}_3$ where α_i dictates the influence of the i^{th} phenotype on the blend. The usual reconstruction process leads to the blended image. All vectors in this diagram are drawn with respect to the sub-sample mean, denoted by the circular marker.

4.4 Facial attribute manipulation

Comparative semantic labels are often used to describe facial appearance. For instance, a witness may describe a perpetrator as *more masculine* or *older* with respect to the composite image. Although the perception of facial attributes is subjective, a consensus can often be found. For example, if a large number of participants are asked to assign a score estimating the age of a thirty year old subject, the mean value is likely to be approximately thirty years, although individual scores may vary above and below the average. Traditional methods for producing composite imagery are incapable of accommodating such semantic descriptions. Conversely, the controlled manipulation of facial attributes is relatively easy to implement in the EigenFIT framework by identifying directions in the parameter space that relate to a trend in a specific attribute. This section outlines a simple procedure for defining a direction through the appearance model parameter space, corresponding to maximum variation in a specified facial attribute. A method for modifying the strength of a chosen attribute within a given face is also described. A similar approach to modifying facial appearance has previously been described by Burt [14] and Benson [5], in which shape and texture were treated separately. An appearance model offers a more elegant solution in which facial appearance can be modified by perturbing a single vector of parameters that simultaneously control both shape and texture.

4.4.1 Training process

To manipulate a chosen facial attribute, a prior training procedure is required in which a relationship is sought between the attribute of interest and each appearance parameter. For attributes in which a dichotomy exists, such as the sex attribute, the simplest approach is to separate the training examples into two classes C_a and C_b (e.g. males and females). Prototypes (constructed from class means) can then be formed by determining the mean vector of appearance model parameters for each class, $\bar{\mathbf{c}}_a$ and $\bar{\mathbf{c}}_b$.

$$\bar{\mathbf{c}}_a = \sum_{i=1}^{n_a} \mathbf{c}_{ai} \quad and \quad \bar{\mathbf{c}}_b = \sum_{k=1}^{n_b} \mathbf{c}_{bk} \tag{4.2}$$

where n_a and n_b are the numbers of sample faces that constitute C_a and C_b respectively. Having formed two prototypes, a direction in appearance space is calculated as the difference vector between them,

$$\Delta \mathbf{c} = \bar{\mathbf{c}}_b - \bar{\mathbf{c}}_a \tag{4.3}$$

We refer to the vector of appearance model parameters $\Delta \mathbf{c}$ as the *attribute vector*. In the interest of typographic clarity, the notation $\mathbf{c}_{attr}$ will be used instead of $\Delta \mathbf{c}$, where the subscript *attr* refers to the attribute of interest and can be assigned as is appropriate. For attributes that vary continuously (albeit on a discrete scale - e.g. age) with respect to variations in appearance model parameter values, an alternative approach is required. One such approach is to perform a multiple regression analysis, relating the attribute to the parameter values using a regression equation. Regression methods are closely related to the techniques and procedures employed in classification problems. Ramanathan and Chellappa [74] use probabilistic eigenspaces and a Bayesian classifier to determine the age difference indicated by two images of the same subject's face, where the time interval between the first and second image was in the range 1-9 years. Support vector machines (SVMs) have become prevalent in face classification problems in recent years and their use in recognition [46] and sex classification [65] problems have been studied. Details of alternative approaches to manipulating age (a subtly different problem to classification) can be found in Hill [66], Lanitis [59] and [52]. However, a simpler approach that produces visually acceptable results is to arrange the attribute values (scores) in ascending order and perform a *median cut*, thereby forcing a dichotomy. In this case, each sample face decomposed into its appearance model parameters and assigned to either class C_a or class C_b as follows,

$$\mathbf{c}_i \in \begin{cases} C_a & if \quad s_i < MEDIAN\left(\{s\}\right) \\ C_b & if \quad s_i > MEDIAN\left(\{s\}\right) \end{cases} \tag{4.4}$$

Prototypes can be constructed from C_a and C_b as before and the attribute vector is defined as per equation 4.3. The process is represented schematically for the aging attribute in the left hand side of figure 4.4[2]. Aging attribute vectors were calculated for each of the adequately sampled demographic groups in section 3.4.1. Time-lines for all of the demographic groups represented in EigenFIT are illustrated in figure 4.3. Each time-line is an extrapolation of an attribute vector beyond the young and old prototypes for a specific demographic group. Faces were reconstructed at equidistant points on the time-line. The

[2] The aging attribute is currently the only attribute manipulation tool in the EigenFIT system although other attributes may be incorporated into future versions

time-lines are useful for validating the simple straight line approach over an extended age range. In figure 4.3, typical aging traits can be observed. In the time-lines for the male demographic groups, the jaw shape becomes more pronounced as the age increases and the skin tone becomes darker on the chin and above the top lip indicating stronger facial hair growth. In the White Female example, the skin is notably less taut with age, characterised by folds between the base of the nose and corners of the mouth. These changes in facial appearance are indicative of the effects of aging [90, 30].

4.4.2 Modifying an attribute of a composite face

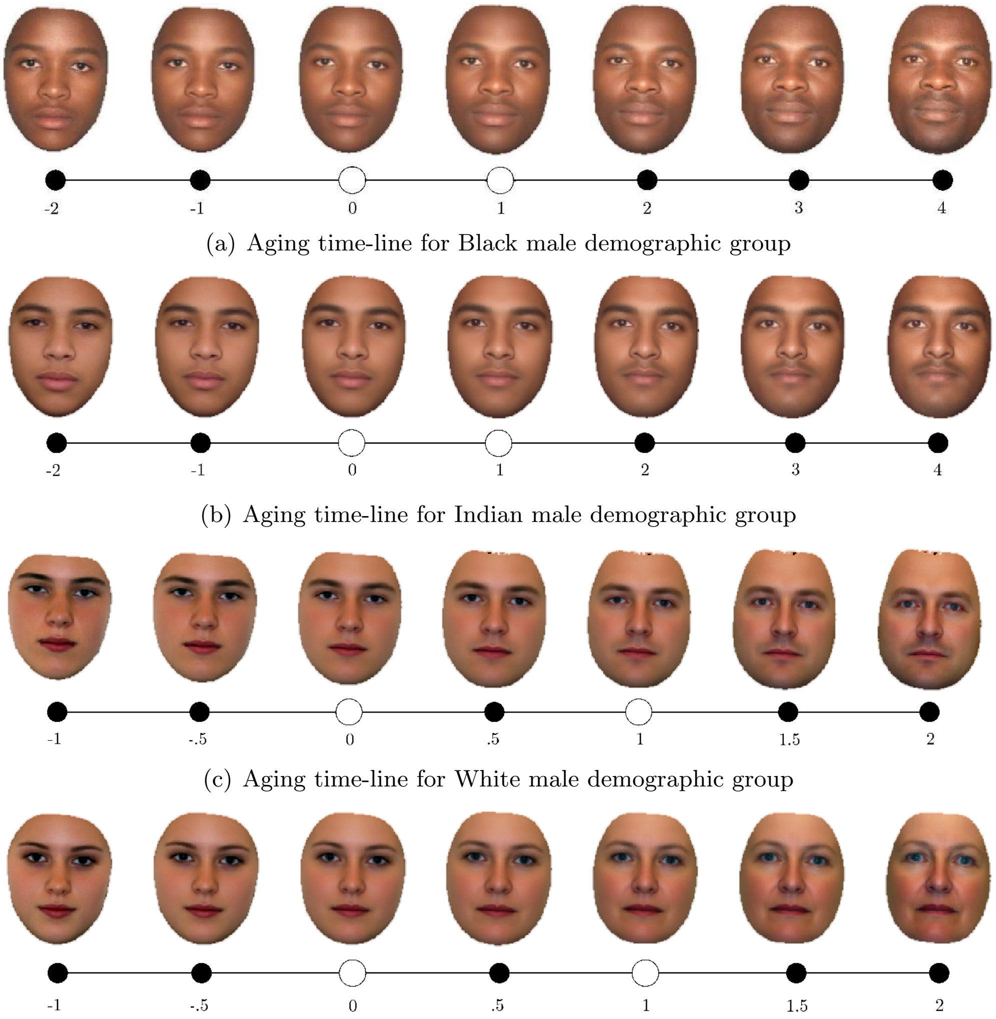
A face is aged or rejuvenated by adding or subtracting a scalar multiple of the aging attribute vector to the appearance model parameters representing the composite face, and then reconstructing the image. In EigenFIT the attribute manipulation is always performed on the current stallion, with corresponding parameter vector $\mathbf{c}_s$. The aging/rejuvenation process can be represented using vector notation as,

$$\mathbf{c}_s \rightarrow \mathbf{c}_s + \alpha \mathbf{c}_{age} \tag{4.5}$$

If the scalar α is positive the stallion will be aged. Conversely, if α is assigned a negative value the stallion be rejuvenated. Here, as elsewhere in this thesis, the → symbol indicates that a process has been executed that updates the current stallion. The process is illustrated schematically in figure 4.4 in which the relevant parameters vectors are added to achieve the desired effect.

4.5 Local feature manipulation

Although one of main strengths of the PCA model is its capability for generating global face images, with facial features displayed in the context of the whole face, there are instances when this holistic approach can be a disadvantage. One important example in which the global manipulation method proves inadequate is when the witness has remembered something distinctive about a particular facial feature and wishes to make a change to a localized region of the composite image. Localized modifications of this nature *can not* be accommodated by the appearance model because in the global PCA framework, alterations to individual features are always accompanied by uncontrollable changes to the face as a whole.



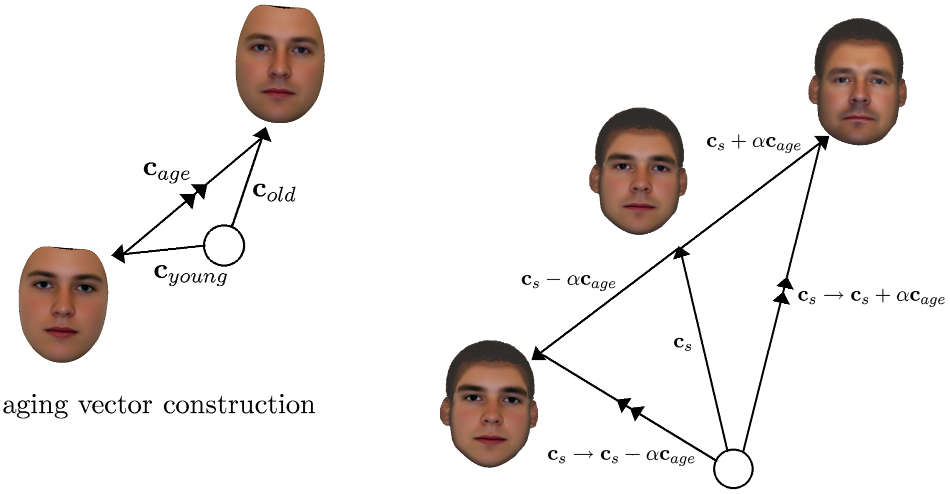
(a) Aging time-line for Black male demographic group

(b) Aging time-line for Indian male demographic group

(c) Aging time-line for White male demographic group

(d) Aging time-line for White female demographic group

Figure 4.3: Aging time-lines for each of the main demographic subsamples. In each case the time-line sequences were produce by adding a proportion (defined by α) of the aging attribute vector to the young face prototype $\bar{\mathbf{c}}_{young}$. For the White demographic groups α was set to each of the following values $[-1\ -.5\ 0\ .5\ 1\ 1.5\ 2]$. The spread of ages in the Black and Indian demographic groups was smaller, therefore α was varied in larger increments $[-2\ -1\ 0\ 1\ 2\ 3\ 4]$ to achieve a similar degree of aging/rejuvenation. White circular markers represent the prototype images through which the time-line passes


$\mathbf{c}_{age}$
$\mathbf{c}_{old}$
$\mathbf{c}_{young}$
$\mathbf{c}_s + \alpha\mathbf{c}_{age}$
$\mathbf{c}_s - \alpha\mathbf{c}_{age}$
$\mathbf{c}_s \rightarrow \mathbf{c}_s + \alpha\mathbf{c}_{age}$
$\mathbf{c}_s$
$\mathbf{c}_s \rightarrow \mathbf{c}_s - \alpha\mathbf{c}_{age}$

aging vector construction

aged stallion and rejuvenated stallion

Figure 4.4: A schematic diagram representing the attribute manipulation procedure. In this 2D simplification, the vector of aging parameters $\mathbf{c}_{age}$ is constructed by determining the difference vector between young-face $\bar{\mathbf{c}}_{young}$ and old-face $\bar{\mathbf{c}}_{old}$ prototypes. The affect of adding and subtracting a scalar multiple of $\mathbf{c}_{age}$ to the stallion $\mathbf{c}_s$ is illustrated by the diagram on the right hand side of the figure. All vectors in this diagram are drawn with respect to the global mean, denoted by the circular marker.

One method for overcoming this limitation is to build independent local PCA models for each of the main facial features [13, 92]. The idea is straightforward in principle. The original training faces are separated into predetermined facial regions, from which localized appearance models can be constructed using the same basic methodology as is used to build global models (previously described in chapters 2 and 3). Facial features generated in this manner are guaranteed to be plausible in appearance, assuming that the correct restrictions are placed on the choice of parameter values. The main issue regarding the use of models based on separate face regions is how to embed the features into the composite image. Unlike the whole face approach, local models do not offer any obvious means for providing credible configurations of features. Forming a seamless join between the separate regions can also prove problematic when using localized texture models. An alternative method was employed in EigenFIT that permited the operator to manipulate the aspect ratio, position and overall size of the individual facial features. The procedure involved shape modifications only, thereby avoiding any potential problems associated with blending texture patches. This straightforward method provided a powerful tool for making adjustments to the ongoing composite image that was not too onerous for the witness. Statements regarding the width, height and position of facial features are easily interpreted and do not require any complex vocabulary that could be misconstrued. As with any user-defined change in facial appearance, the operator themselves may also place implicit constraints on the deformation. Humans are experts in recognising real faces and, as such, are likely to provide a reliable judgement on the plausibility of computer generated faces. With these constraints in place, face shapes that are modified using this *local feature tool* retain a realistic appearance despite the fact that, in general, they lie outside the span of the shape principal components. This implies that the *local feature tool* provides a means for introducing new plausible shape variation that is not afforded by the PCA model itself.

4.5.1 Global and local representations of face shape

The construction of the statistical appearance model described in chapter 2 results in a face being represented by a vector $\mathbf{c} = [c_1\ c_2\ \ldots\ c_n]$ of global appearance parameters. These parameters are global in the sense that altering a single parameter alters both shape and texture of the entire facial appearance. Distinct parameter vectors $\mathbf{b}_s$ for the shape and $\mathbf{b}_t$ for the texture can be obtained directly via the following equations (explained in full in chapter 2),

$$\mathbf{b}_s = \mathbf{Q}_s \mathbf{c} \mathbf{W}^{-1} \tag{4.6}$$

$$\mathbf{b}_t = \mathbf{Q}_t \mathbf{c} \tag{4.7}$$

In turn, the actual global shape vector $\mathbf{x}$ of the face and the shape-normalized texture-map $\mathbf{g}$ of the face can be generated as,

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}_s \mathbf{b}_s \tag{4.8}$$

$$\mathbf{g} = \bar{\mathbf{g}} + \mathbf{P}_t \mathbf{b}_s \tag{4.9}$$

Consider now some arbitrary change introduced into the coordinates of $\mathbf{x}$ by a *local* manipulation of a feature shape[3] such as the nose or mouth so that,

$$\mathbf{x} \rightarrow \mathbf{x}' = \mathbf{x} + \Delta \mathbf{x} \tag{4.10}$$

To represent this vector in the global shape space, spanned by the principal components, we must project the vector onto the shape principal components contained in the columns of matrix $\mathbf{P}_s$,

$$\mathbf{b}'_s = \mathbf{P}_s^T \left(\mathbf{x}' - \bar{\mathbf{x}} \right)$$

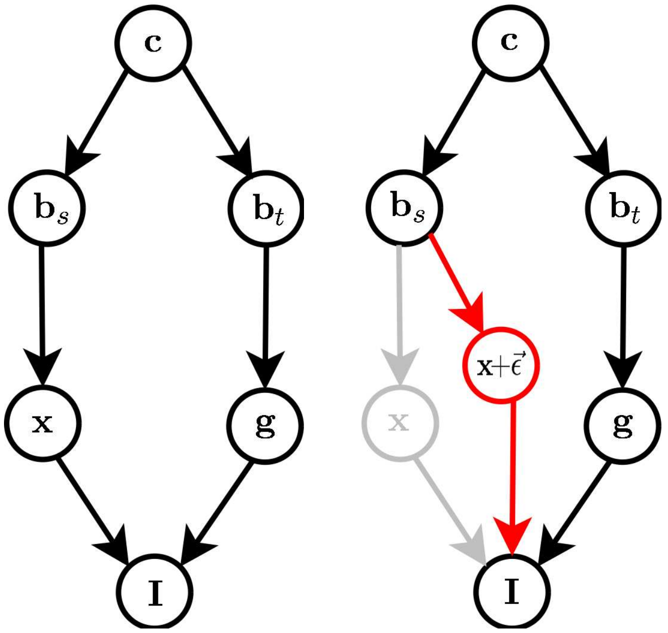
The localized manipulation of coordinates can result in a new shape vector, which does not lie within the span of the shape space and a new shape vector $\mathbf{b}'_s$ will not, in general, enable exact reconstruction of the shape vector $\mathbf{x}'$ according to equation 4.8 (see also figure 4.5). Rather we have,

$$\mathbf{x}' = \bar{\mathbf{x}} + \mathbf{P}_s \mathbf{b}'_s + \vec{\epsilon}_s \tag{4.11}$$

where $\vec{\epsilon}_s$ defines a component of the modified shape that cannot be constructed using a linear combination of the shape principal components $\mathbf{P}_s$, since it is orthogonal to the vector space spanned by the columns of $\mathbf{P}_s$.

It is clear that to maintain an accurate parametric representation of the generated composite, it is necessary to keep a record of both the global appearance

[3] An identical argument can be applied to deterministic changes in the global texture vector, where $\mathbf{g} \rightarrow \mathbf{g}' = \mathbf{g} + \Delta \mathbf{g}$.



(a) Standard image reconstruction process

(b) Image reconstruction process with *deterministic shape modification*

Figure 4.5: A schematic diagram illustrating the image (**I**) reconstruction process starting with a vector of appearance model parameters **c**. In the first level of reconstruction, a shape parameter vector, $\mathbf{b}_s$, and texture parameter vector, $\mathbf{b}_t$, are obtained as shown in equation 4.7. From $\mathbf{b}_s$ and $\mathbf{b}_t$ a face shape vector **x** and texture vector **g** are reconstructed (see equations 4.9 & 4.8). In the standard reconstruction process, **g** is re-shaped into a shape normalized texture-map, which is then warped to face shape **x**, thereby forming the reconstructed face image **I**. In figure (b) a deterministic shape modification is indicated by $\mathbf{x} + \vec{\epsilon}_s$ where the $\vec{\epsilon}_s$ component lies outside of the vector space spanned by the columns of $\mathbf{P}_s$.

vector $\mathbf{c}$ and the component of the shape deformation that is perpendicular to shape principal components, $\vec{\epsilon_s}$. The dimensionality of $\vec{\epsilon_S}$ is high compared to the very compact representation provided by the appearance vector (or indeed its associated associated shape and texture parameters).

One approach to the book-keeping of composite production is to incorporate any local deterministic changes introduced by the operator and witness through repeated projection of the shape and texture vectors onto their respective principal components. Thus, obtaining the nearest global representation, and the component $\vec{\epsilon_s}$. In this work, a simpler but equally effective approach was taken which is summarized as follows,

1. Allow the core evolutionary procedure to continue as in the standard mode of operation, resulting in instances of whole face variations on the current stallion.
2. Any deterministic changes to the shape and texture introduced by the witness are recorded and treated *independently* of the global model as *offset vectors*. In effect, the net deviation of the shape and texture from that predicted by the global model is updated on each occasion that deterministic changes are made.

Let $\mathbf{x}$ be the face shape of the stallion phenotype in the current generation and let $\mathbf{x}'$ be an instantaneous face shape resulting from a local deterministic manipulation. The face shape that is displayed by the witness via the user interface is,

$$\mathbf{x}_{screen} = \mathbf{x} + \Delta x, \ \ with \ \Delta x = \mathbf{x}' - \mathbf{x} \tag{4.12}$$

The major advantage of this approach is that it removes the computational overhead which is associated with recalculation of the global model and simplifies the implementation. In this case the Δx will, in general, lie outside the span of the column space of $\mathbf{P}_s$ but unlike $\vec{\epsilon_s}$ will not be orthogonal to the column space of $\mathbf{P}_s$.

4.5.2 Defining facial regions

Localized shape deformations are achieved by warping the displayed face image from its current shape to a new, modified, face shape as defined by the operator. For the tool described here, a set of landmark points was used to define the shape and position of the facial features. In this section, $\mathbf{x}$ will be used to denote the $2N$ element column vector of control points representing the undeformed, reference face shape [4] and $\mathbf{x}'$ to indicate the face shape after deformation or at any instant during a succession of shape manipulations. The shape vectors can be written as the concatenation of two vectors, $\mathbf{x}_h$ and $\mathbf{x}_v$, containing the x and y coordinates respectively. Subscript labels h and v signify horizontal and vertical deformation directions.

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_h \\ \mathbf{x}_v \end{bmatrix}, \qquad \mathbf{x}' = \begin{bmatrix} \mathbf{x}'_h \\ \mathbf{x}'_v \end{bmatrix} \tag{4.13}$$

In order to deform features independently from the face shape as a whole, the appropriate coordinates in $\mathbf{x}_h$ and $\mathbf{x}_v$ had to be identified. Since only one feature may be modified at any instant, a method was required that accessed the relevant coordinates only, and left the remaining coordinates unaltered. This could be achieved by forming separate coordinate vectors for each of the facial features. However, a more elegant approach was taken that simplified the code and allowed the translation and scaling equations to be expressed in a general vector form. If $L = \{1, 2, 3, \ldots, N\}$ represents the set of all indices into the N element x and y coordinate vectors, then let L_f be a subset ($L_f \subset L$) of these indices corresponding to feature f. For each facial feature a N element binary column vector, $\mathbf{b}$[5], was formed such that $i \in L_f \rightarrow b_i = 1$ and $i \notin L_f \rightarrow b_i = 0$. Hence, entries in vector $\mathbf{b}$ that are equal to 1 indicate landmarks that are repositioned during a local feature manipulation process and entries in $\mathbf{b}$ that are equal to 0 represent landmarks which remain fixed in the user-defined shape change.

4.5.3 Translating facial features

Spatial relationships between the facial features have been shown to play an important part in recognition [100]. The local feature tool provides the means for displacing the centroid of a selected feature. An interface was constructed that enabled the operator to translate the x, y coordinates of a localized region of the face by positioning two sliders corresponding to horizontal and vertical positioning. Eight features were accommodated; *nose*, *left eyebrow*, *right eyebrow*, *left eye*, *right eye*, *mouth*, *face* and *hair*. Some features were coupled so as to preserve the vertical symmetry. The left and right eyes were coupled so that

[4] In this example $\mathbf{x}$ resembles the point model described in chapter 3, with additional landmarks that delineate the hair

[5] Elsewhere in this thesis $\mathbf{b}$ with subscript has also been used to represent the a parameter vector in the shape and texture models previously described.

their respective shape centroids mutually converged, or diverged, depending on the direction of the horizontal slider movement. The left and right eyebrows were also coupled, maintaining symmetry about the vertical axis that dissects the face into its left hand and right hand sides. Different features could be selected from a list box within the interface. The choice of feature determined the **b** vector (**b** vectors̲ for coupled features) that occured in the translation equation. Algorithm 2 provides an overview of the process. It is important to note that when producing facial modifications using sliders, the slider travel has to effect a change with respect to the original image, otherwise the position of the slider is meaningless and the resulting deformations are hard to control. The implication for translating features is that each time a slider is adjusted, its position causes an absolute change in the transition of a feature with respect to it's original centroid and not with respect to its instantaneous centroid as determined by the previous slider adjustment.

Algorithm 2 Algorithm for translating facial features

```
if feature is coupled then
  n=2
else
  n=1
end if
for $i = 1$ to $n$ do
  if horizontal slider adjusted then
    · calculate translation scalar, $T$, according to slider position
    · multiply translation scalar by $(-1)^{i+1}$
    · update x-coordinates
  else if vertical slider adjusted then
    · calculate translation scalar, $T$, according to slider position
    · update y-coordinates
  end if
end for
```

The first **if** statement in algorithm 2determined whether the selected feature was coupled with another feature (eg. eyes). If a coupled feature was selected, two passes were made through the **for** loop. When an adjustment was made to the slider for vertical translation, both of the coupled features moved in unison, either in an upwards or downwards direction. In the first pass through the for loop, the coordinates corresponding to the right feature, specified by $\mathbf{b}_1$, were updated by computing the translation scalar, T, and adding it to the y coordinates of the feature. Similarly, in the second pass the y coordinates, specified by $\mathbf{b}_2$, for the left feature were adjusted by the same amount. For uncoupled features the required translation was computed in a single pass ($n =$

1) through the **for** loop.

Horizontal translation was achieved using an almost identical procedure to the method used for vertical displacement. The one difference was the relative displacement of the coupled features. For the case in which a horizontal translation was required, coupled features were moved simultaneously towards each other or apart, depending on the direction of the slider movement. The first pass through the **for** loop caused the right facial feature to be translated in same direction as the slider movement. Conversely, in the second pass, the polarity of the translation scalar was reversed ($T \rightarrow -T$) causing the left feature to move in the opposite direction to the right feature. The polarity change was obtained using the multiplier, $(-1)^{i+1}$, where i is the loop counter. For an uncoupled feature only one pass was made through the **for** loop with the term $(-1)^{i+1}$ remaining positive. Equation 4.14 shows how the translation scalar, T, is determined. A horizontal translation of a selected facial feature is represented in equation 4.15 where T determines the magnitude of displacement and the vector $\mathbf{b}_f$ determines which feature is displaced. A similar equation can be written for vertical translation with the $(-1)^{i+1}$ multiplier removed.

$$T = -\frac{1}{\mathbf{b}_f^T \mathbf{b}_f} \mathbf{b}_f^T \mathbf{x}'_h + \frac{1}{\mathbf{b}_f^T \mathbf{b}_f} \mathbf{b}_f^T \mathbf{x}_h + (-1)^{m+1} (d_h - 1)\, s_{\mathbf{x}} \tag{4.14}$$

$$\mathbf{x}'_h \rightarrow \mathbf{x}'_h + T\mathbf{b}_f \tag{4.15}$$

The first term in equation 4.14 is the centroid of the current feature (x-coordinates only). Similarly, the second term is the centroid relating to the original position of the selected feature prior to deformation. In this formulation, coordinates corresponding to the selected feature (f) are 'picked out' by the vector $\mathbf{b}_f$. Elements of $\mathbf{x}_h$ relating to unselected facial features are weighted by 0 and therefore have no effect on the centroid of the selected feature, as required. The inner product $\mathbf{b}_f^T \mathbf{b}_f$ is simply the number of landmark points N_f contained in the facial feature of interest. A third term defines the magnitude and polarity of the translation as determined by the slider position represented by the scalar variable d_h. s_h is the sample standard deviation of $\mathbf{x}_h$ and is used here to map the slider value to a translation that is proportionate to the size of the face shape. $(-1)^{i+1}$ changes the polarity of the translation and only applies to the displacement of coupled features in the horizontal direction. Finally, the translation scalar is added to the elements of $\mathbf{x}'_h$ that relate to the selected feature using the binary column vector $\mathbf{b}_f$ to form a vector of updated coordinates. The action of each term in the right hand side of equation 4.14

can be summarized as follows,

- **term 1:** Subtract the centroid of the instantaneous (modified) feature shape, thereby translating to the origin.
- **term 2:** Add reference feature centroid (translate to the reference position)
- **term 3:** Make a translation from the reference position to a new instantaneous position according to the instantaneous slider position.

4.5.4 Scaling facial features

The second available deformation type in the local feature tool allowed the operator to change the size of a chosen feature or alter its aspect ratio by applying a different scaling to the horizontal and vertical coordinate vectors, $\mathbf{x}_h$ and $\mathbf{x}_v$. Coupled features were scaled using consecutive passes through a **for** loop, via a method comparable with the procedure used when translating coupled features. Unlike the translation method, the code for producing horizontal scaling and the code for vertical scaling were identical in every respect. An overview of the algorithm is provided for completeness (algorithm 3).

Algorithm 3 Algorithm for scaling facial features

```
if feature is coupled then
  n=2
else
  n=1
end if
for $i = 1$ to $n$ do
  if horizontal slider adjusted then
    · duplicate original x-coordinates
    · translate duplicated coordinates to the origin
    · scale duplicated coordinates according to slider movement
    · translate back
    · update feature x-coordinates
  else if vertical slider adjusted then
    · duplicate original y-coordinates
    · translate duplicated coordinates to the origin
    · scale duplicated coordinates according to slider movement
    · translate back
    · update feature y-coordinates
  end if
end for
```

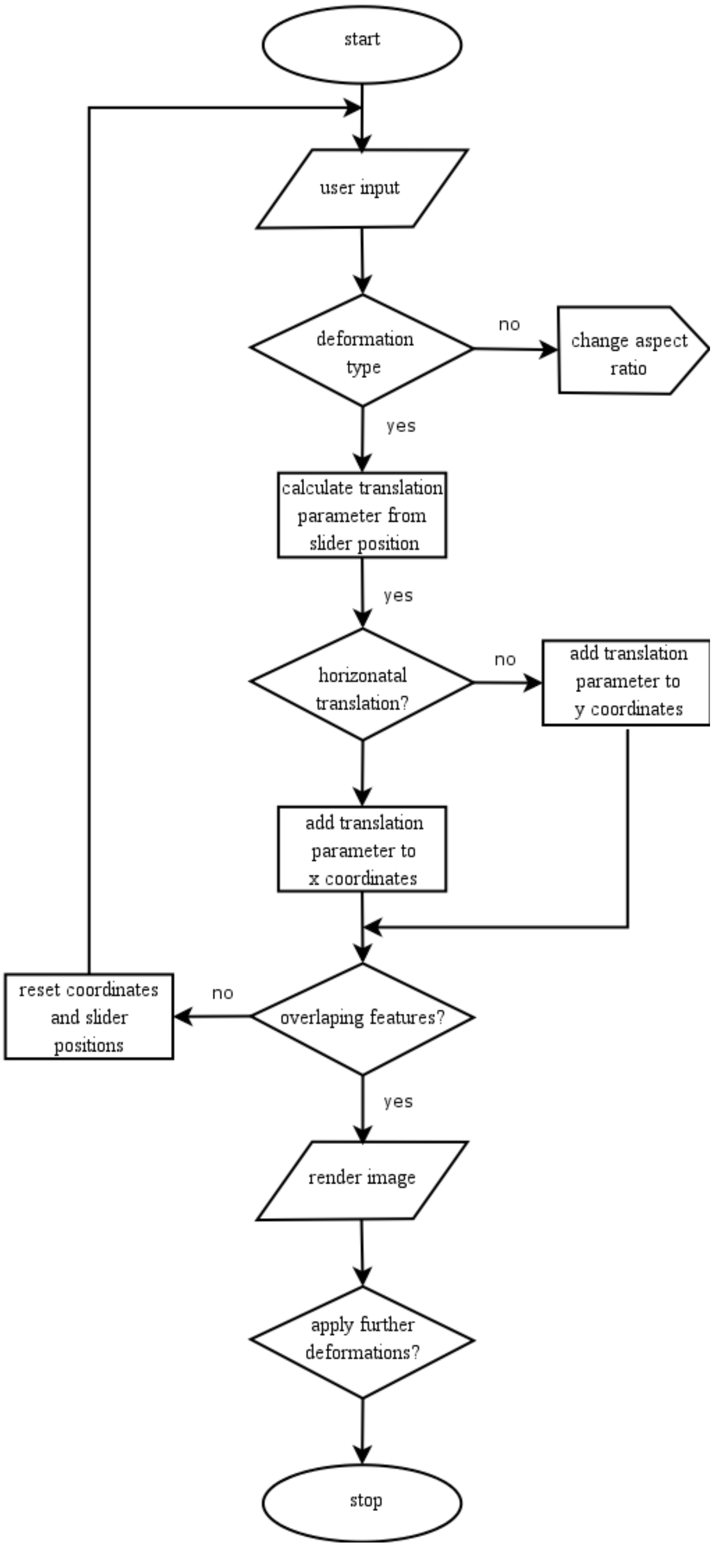
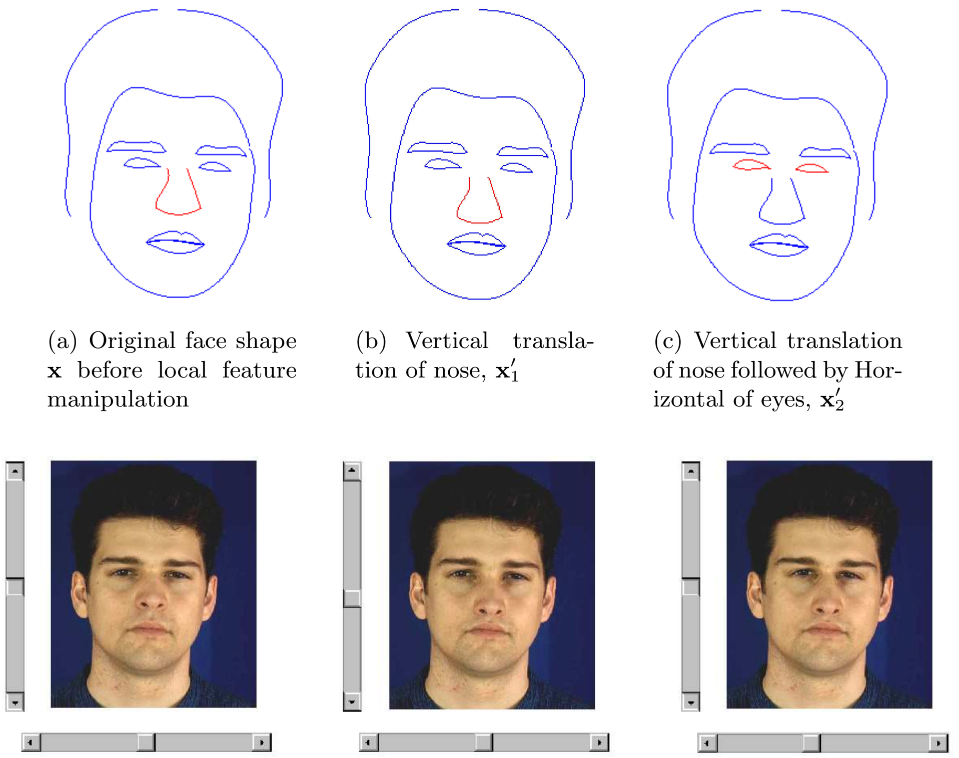

start
user input
deformation type
no
change aspect ratio
yes
calculate translation parameter from slider position
yes
horizonatal translation?
no
add translation parameter to y coordinates
add translation parameter to x coordinates
reset coordinates and slider positions
no
overlaping features?
yes
render image
apply further deformations?
stop

Figure 4.6: A flow chart depicting the decision processes involved in translating facial features using the local feature tool.



(a) Original face shape $\mathbf{x}$ before local feature manipulation

(b) Vertical translation of nose, $\mathbf{x}'_1$

(c) Vertical translation of nose followed by Horizontal of eyes, $\mathbf{x}'_2$

(d) Original face image

(e) Rendered image resulting from nose translation

(f) Rendered image resulting from nose and eye translations

Figure 4.7: Local feature manipulation: *Translation* of facial features using sliders. Sub-figures (d-f) show the original image and images resulting from local feature modifications. Sub-figures (a-c) show the face shape deformations which result in images (d-e). In images (a-c), polynomial curves have been fitted to the landmark points, thereby providing a clear representation of the face shape in each case.

A vector equation describing the scaling process can be written as per equation 4.16. This expression is more complicated than the equation for the translation case and requires some explanation.

$$\mathbf{x}'_{h\_new} = \left(\mathbf{x}_h - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\mathbf{x}_h\right) \circ d_h\mathbf{b}_f + \left(\frac{1}{\mathbf{b}_i^T\mathbf{b}_f}\mathbf{b}_f\mathbf{b}_f^T\mathbf{x}'_h\right) + \mathbf{x}'_h \circ (\neg\mathbf{b}_f) \quad (4.16)$$

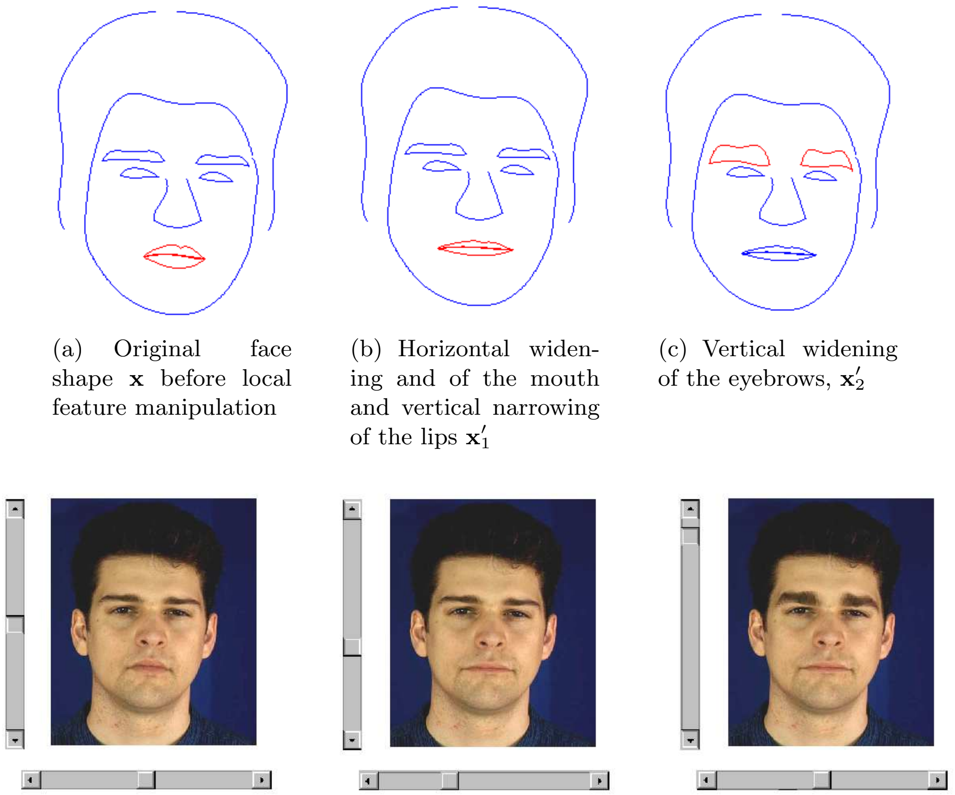
The first term on the right hand side of equation 4.16. $\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\mathbf{x}_h$ calculates the horizontal centroid of the selected feature and replicates this value in every entry of an N element column vector. It is then subtracted from the vector containing the x-coordinates of the original shape vector, $\mathbf{x}_h$, translating the whole face shape to the origin. A binary column vector $\mathbf{b}_f$ is multiplied by a scalar representing the slider movement d_h. Unlike in the translation equations, here the slider value maps directly to a fractional scaling value, thereby scaling the coordinates corresponding to the selected feature. The quantity $\frac{1}{\mathbf{b}_f^T\mathbf{b}_f}\mathbf{b}_f\mathbf{b}_f^T\mathbf{x}'_h$ performs a similar function to $\frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\mathbf{x}_h$. $\frac{1}{\mathbf{b}_f^T\mathbf{b}_f}$ is a normalization factor with $\mathbf{b}_f^T\mathbf{b}_f$ equal to the number of coordinates in the selected feature. $\mathbf{b}_f\mathbf{b}_f^T$ is a binary matrix, that is multiplied by the current x-coordinate vector to form a new N element vector in which the non-zero elements are equal to the horizontal centroid of the selected feature. The last term on the r.h.s. of equation 4.16 retains the current coordinate values for the features that remain unaltered by the scaling transformation. The symbol $\neg$ is used to represent logical negation, thereby setting all zero elements of $\mathbf{b}_f$ to unity and vice versa.

The role of each term on the right hand side of equation 4.16 can be summarized as follows,

- **term 1:** The bracketed part of the first term translates a copy of the whole undeformed/reference face shape to the origin of the horizontal axis. The point-wise multiplying factor $d_h\mathbf{b}_f$ performs a horizontal scaling of the selected feature and sets the horizontal landmarks corresponding to the undeformed features to zero.

- **term 2:** A horizontal translation of the selected feature to its instantaneous position (different to position of reference face shape centroid if prior shape translations have been applied).

- **term 3:** Resets the unmodified feature landmarks to their instantaneous positions such that they remain unaltered during the current scaling manipulation.



(a) Original face shape $\mathbf{x}$ before local feature manipulation

(b) Horizontal widening and of the mouth and vertical narrowing of the lips $\mathbf{x}'_1$

(c) Vertical widening of the eyebrows, $\mathbf{x}'_2$

(d) Original face image

(e) Rendered image resulting from horizontal and vertical scaling of the mouth

(f) Rendered image resulting from scaled mouth and eyebrow shapes

Figure 4.8: Local feature manipulation: *Scaling* of facial features using sliders. Sub-figures (d-f) show the original image and images resulting from local feature modifications. Sub-figures (a-c) show the face shape deformations which result in images (d-e). In images (a-c), polynomial curves have been fitted to the landmark points, thereby providing a clear representation of the face shape in each case.

Once the face shape has been modified to the witness' satisfaction the corresponding phenotype image is updated by warping it to the new face shape using a piece-wise affine warp.

4.6 Applying fine details to a composite

In general, given an appropriate training sample, the AM is capable of capturing most of the natural variation in facial appearance. However, the fine structures such as wrinkles and freckles that often occur in real faces are under represented in face images synthesized from the AM. This shortfall is particularly apparent when generating faces of old subjects since fine facial detail becomes more prevalent with an increase in age. The problem is intrinsic to the process by which new examples of faces are synthesized, whereby a new example of a face is constructed by a weighted average of existing face images. Inevitably this averaging procedure results in a certain degree of smoothing in the synthesized face. Fine details that exhibit low spatial correlation between observations (sample faces) tend to be 'averaged out' by this process. Although wrinkles are often more common in specific regions of the face (for example 'crow's feet' appear at the outer corners of the eyes), their prominence, exact position, and frequency of occurrence vary from subject to subject. Landmarking these delicate features is not practicable, therefore a different approach is required.

The issue of enhancing fine detail in averaged face images has been investigated in previous work. In Tiddeman [91] a prototype face, formed by averaging a sample of face images, was decomposed using a wavelet analysis and the high order details boosted to compensate for the inherent loss of high spatial frequency information. From a theoretical point of view, this approach may be flawed because it does not take into account the poor spatial correlation of fine facial details between sample faces. In fact, if the spatial correlation is assumed to be negligible, and the prototype face image constructed from an infinite sample of images, the fine details will sum to a flat field. It is obvious that boosting this flat field will not enhance wrinkles and other fine structures.

4.6.1 Extracting wrinkle-maps from sample images

In the remainder of this section, an ad-hoc method for applying fine facial detail to a target face is discussed. The idea is to extract the fine facial details from a sample face A with the intention of applying these details directly to a target face B. Let I_A be the image corresponding to subject A and let I_{Ad} be a detail-image containing the high spatial frequencies from I_A. If A is an elderly subject, then image I_{Ad} will be likely to contain facial wrinkles associated with the aging process, therefore the term *wrinkle-map* will be used when referring to I_{Ad}. Wrinkle-maps can be obtained by passing image I_A through a high pass filter, or equivalently by subtracting a smoothed version of the image from itself as follows,

$$I_{Ad}\left(x,y\right) = I_A\left(x,y\right) - \bar{I}_A\left(x,y\right) \tag{4.17}$$

where $\bar{I}_A\left(x,y\right)$ is a blurred version of I_A. The procedure is illustrated schematically in in figure 4.9.

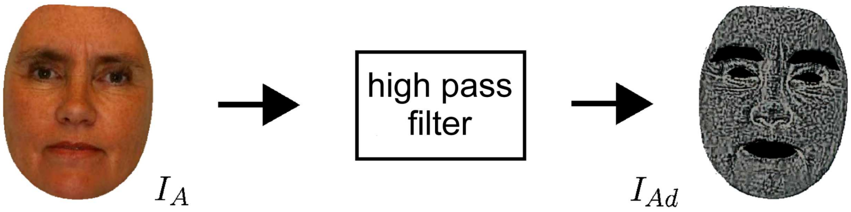


Figure 4.9: Wrinkle-map I_{Ad} extracted from a sample face A using image subtraction 4.17. In this example, I_{Ad} was created by using an 8×8 averaging filter to form a smoothed image $\bar{I}_A$, which was then subtracted from the $340\times 267\times 3$ original I_A.

4.6.2 Applying details to a target face

Once extracted from a subject (for example subject A in figure 4.9), a wrinkle-map can be applied to a target face, thereby increasing its fine detail content. This technique is particularly useful for increasing the apparent age of a subject. The method can be considered as a hybrid form of the standard *high-boost filtering* method commonly used in digital image processing (see Gonzalez [38]),

$$I_{Bhb} = \left(A-1\right)\bar{I}_B + I_{Bd},\ \ A \geq 1 \tag{4.18}$$

$$I'_{Bhb} = \left(A-1\right)\bar{I}_B + I_{Ad},\ \ A \geq 1 \tag{4.19}$$

In the normal implementation of high-boost filtering, a detail image is constructed as defined by equation 4.17 and added to a fraction of the original image (equation 4.18). This procedure makes the fine details more prominent in the high-boost image I_{Bhb}. As the value of the scalar A increases beyond 1, the contribution of the detail image becomes less important. In practice

$A \approx 2$, and the filter kernel used to form $\bar{I}_A$ in equation 4.17 should be small enough that only fine details are copied to subject B. If these precautions are not adhered to the identity of subject B may be altered when the wrinkle-map is applied. Equation 4.19 is a modified version of the standard high-boost procedure, whereby the detail image is not formed from the original image, but from a different image instead. The *hybrid high-boost* image that results, I'_{Bhb}, comprises the low-medium spatial frequencies of image I_B and the high spatial frequencies belonging to image I_A. A schematic overview of this hybrid high-boost method is provided in figure 4.10. Superposition of fine details from **both** subjects can produce undesirable ghosting artefacts in the hybrid high-boost image I'_{Bhb} (i.e. plausible results are not obtained if more than one wrinkle-map is used in the hybrid high-boost image). Passing a low-pass filter over image I_A prior to adding the detail image overcomes this problem. The effect of varying the size of the spatial filter used in the construction of I_{Ad} is illustrated in figure 4.12.

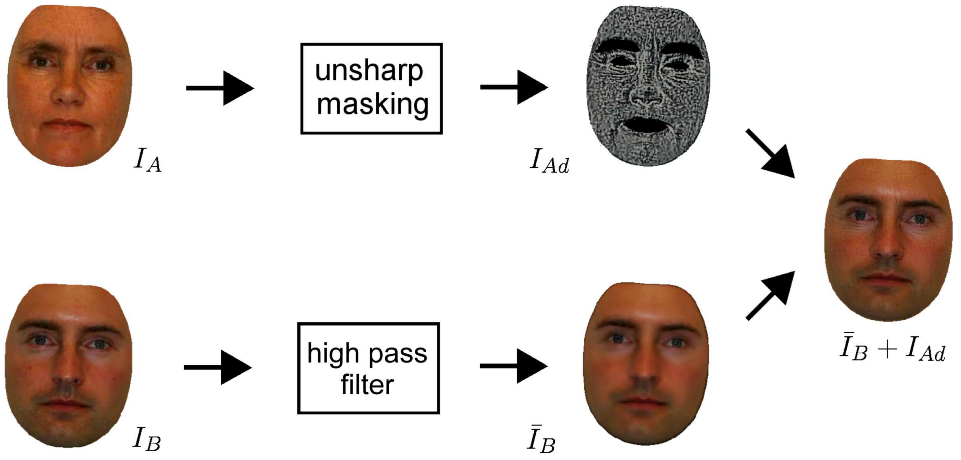


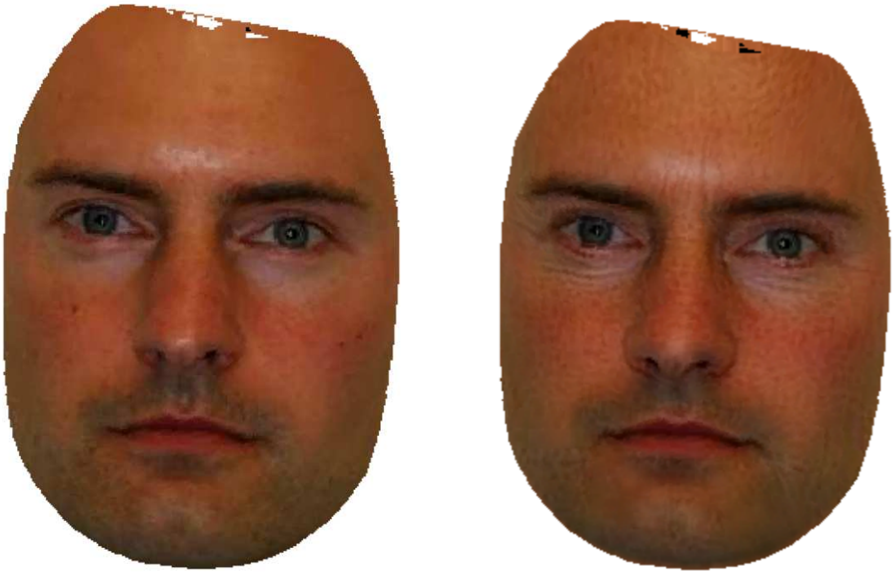
Figure 4.10: Application of a wrinkle map obtained from subject A to a target face B

4.6.3 Application to facial composites

A deterministic method for applying fine facial details in the EigenFIT system is desirable for two reasons,

1. Wrinkles can be added 'at will' to a composite image to increase its apparent age.

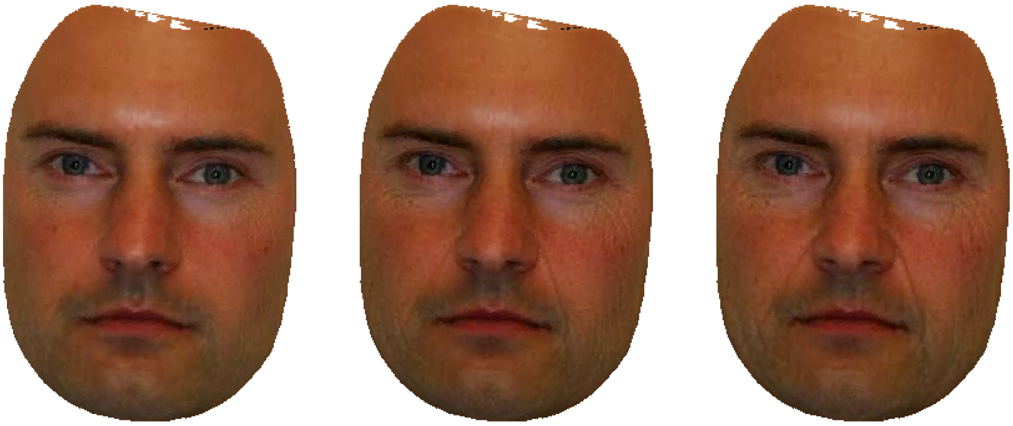
2. Fine facial details are poorly represented by the appearance model causing new instances of faces to appear artificially smooth.



(a) Target face image I_B

(b) Target face image with wrinkle-map from subject B

Figure 4.11: A comparison between a veridical target face image I_B and the same face image with a wrinkle-map applied to it. Both images have been warped to the veridical face shape. Wrinkles are visible around the eyes of image in sub-figure (b) and the whole face exhibits a freckled complexion. Conversely the image in sub-figure (a) is relatively smooth in comparison.



(a) Filter size = 4 × 4 (b) Filter size = 8 × 8 (c) Filter size = 12 × 12

Figure 4.12: The effect of varying the size of the averaging filter on the images obtained from the hybrid high-boost method for applying wrinkle-maps. The identity of the target face is retained even for the 12 × 12 filter image which contains a relatively large proportion of facial detail from another subject. A different wrinkle-map to the one shown in figure 4.11 has been used in this example, to show the generality of the procedure.

Fine facial details are not modelled sufficiently by the AM because, unlike the main facial features in shape normalized images, a spatial correlation does not exist for these image structures over the sample of training images. Hence, the fine details tend to occur in the later principal components which are discarded according to,

$$t_m = 100\frac{\sum_{k=1}^{m}\lambda_k}{\sum_{j=1}^{p}\lambda_j}$$

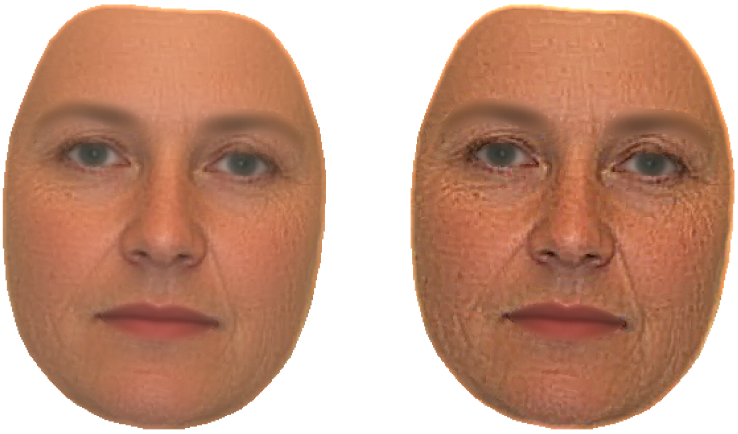
where t_m is the percentage of natural facial variation retained in the AM and m is the cut off point whereby the $(m+1)^{th} - p^{th}$ principal components are removed from the model. Using the assumption that fine facial details are not present in images synthesized from the AM, a wrinkle-map can be added to a composite face without the preliminary smoothing procedure previously described.

The witness' recollection of fine facial structure is likely to be quantitative rather than relating to a specific spatial arrangement of fine details. For example, typical semantic descriptions may be "the perpetrator had a spotty complexion", "the face was slightly wrinkled" or "the face was very wrinkled". Consequently, only a limited number of wrinkle-maps will be required in the composite system. For ease of implementation the prominence of the wrinkle-maps will be controlled by varying the relative proportions of $\bar{I}_B$ and I_{Ad} by adjusting the value of A in equation 4.19, as illustrated in figure 4.13, rather than adjusting the filter size as previously shown in figure 4.12. The wrinkle-maps can then be pre-calculated, and loaded into memory as required.

Although the wrinkle-maps are obtained from the same training examples used to build the appearance model, this does not imply that a perfect PCA representation is possible. The process of mixing spatial frequency content produces a texture map that does not lie in the span of the texture P_g. To illustrate this point let the texture-map of the current stallion phenotype be represented in vector notation as $\mathbf{g}$ and the wrinkle-map by $\Delta\mathbf{g}$. The process of updating the stallion can then be expressed by,

$$\mathbf{g}_{screen} = \mathbf{g} + \Delta\mathbf{g} \tag{4.20}$$

where the component $\Delta\mathbf{g}$ lies outside the span of, but is not orthogonal to, $\mathbf{P}_s$. Hence, the wrinkle-map is treated as an offset vector and $\mathbf{g}$ is generated by the standard evolutionary process as described in chapter 3 which is unaffected by the deterministic process of applying a wrinkle-map. An example of combining the wrinkle-map and attribute manipulation procedures is presented in Appendix D.



(a) $A = 2$, filter size $= 8 \times 8$ (b) $A = 2.5$, filter size $= 8 \times 8$

Figure 4.13: In the interest of ease of implementation, the prominence of each wrinkle map is determined by the relative proportions of target face and wrinkle-map. The relative proportions can easily be controlled by varying the scalar A in the hybrid high-boost equation 4.19

4.7 Summary

A suite of advanced tools has been designed and smoothly integrated with the core EigenFIT implementation. The *blend tool* allowed the witness to blend two or more faces in the current generation, permitting facial characteristics from multiple faces to be propagated into the next generation. Incorporation of the blended face into the standard evolutionary process was achieved by forming a weighted average of genotypes (in this context, transformed appearance model parameter vectors), which replaced the genotype corresponding to the current stallion.

A simple, general, method for affecting facial attributes was outlined. Implementation of this method for the specific case or aging/rejuvenation was discussed in detail, in which the genotype relating to the current stallion was perturbed by the addition of an appropriately scaled *attribute vector*.

Deterministic face-shape deformations were made possible with a *local feature manipulation tool*, with which the relative position and aspect ratio of a selected facial region could be adjusted as required. In general, the exact encoding of a shape manipulation attained in this manner is not possible using the appearance model framework. Hence, local shape alterations were stored in an offset vector, allowing the underlying standard evolutionary process to continue in the usual way. Prior to display, the offset vector was recombined with the underlying face shape in a layer of computation that exists between the appearance model and user interface components of the EigenFIT system.

The concept of *wrinkle-maps* was introduced as a deterministic method for applying fine facial details to a facial composite. Since such changes to a composite image can not be accommodated by the appearance model alone, the selected wrinkle-map was stored as an offset in vector form. In a computational step that preceded screen-display, the offset was added to the texture-map, generated by the underlying evolutionary process.

Chapter 5

Efficient model-based warping method

One of the primary aims of the work described thus far is to provide a mechanism for producing facial composite images quickly. Long time delays between the synthesis of successive generations of faces must be avoided since this is both distracting for the witness and considerably lengthens the overall composite process. When synthesizing a new example of a face, a bottleneck is encountered, due to the necessary image warping stage. In Chapter 2, section 2.3.1 an image warping procedure was described that utilized a piece-wise affine transformation, reverse pixel mapping and nearest neighbour interpolation. However, the limits on natural face shape variation and the procedure for synthesizing a face from an appearance model suggest an approach to this constrained image warping problem that is computationally more efficient. In this chapter, a novel method for image warping is described in which an image warp is effected by the superposition of a set of pre-defined forward mappings referred to in this work as *displacement fields*.

5.1 Introduction

Image warping is a geometric transformation that maps all pixel positions in one image plane to positions in a second plane. Various methods for achieving image warps have been previously described (see Glasbey and Mardia [37]). The choice of warping method is dictated by a compromise between a smooth geometric distortion, accuracy of control point (shape) alignment and speed of execution.

The piecewise affine method is generally accepted to be the fastest general warping method. It has been used in deformable template methods for real time pattern recognition problems [21]. It entails segmenting the source image and destination image into corresponding triangular regions. For each triangular region an affine transformation must be determined which is then applied to every pixel location, and requires the application of a matrix multiplication for each and every pixel location. The piece-wise nature of this method can lead to visible facets in the destination image.

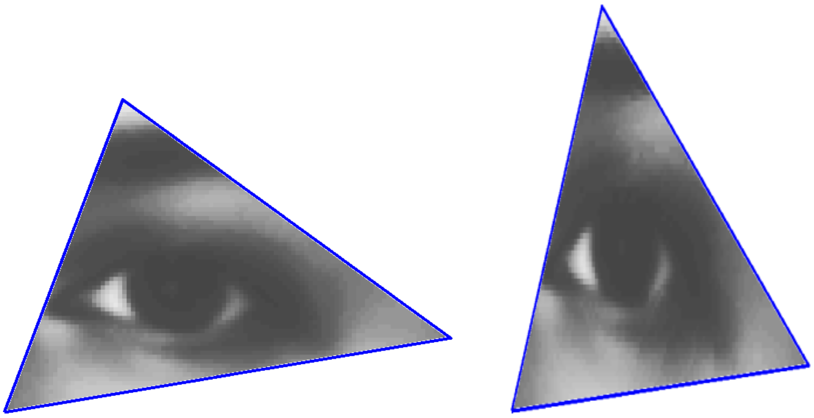
An approach that produces a continuous warp is Bookstein's thin-plate spline (TPS) method [6]. This is essentially based on the concept of a surface which is warped to assume a different form. It is a *global* (as distinct from piecewise) transformation which maps source landmarks to destination landmarks but in such a way as to minimize the overall bending energy of the surface. In this way, the TPS produces very smooth warps and avoids a problem which can occur with the piece-wise approach in which adjacent triangular regions do not blend naturally. In general, perfect registration of control points is not achieved using this method. The TPS method is typically more computationally demanding than the piecewise affine method and therefore is not a compelling choice for real-time applications such as the one described in this thesis.

Although a reverse mapping from the destination image to the source image is often preferred, sometimes it is difficult or even impossible to obtain. This problem arises when complex surfaces are modelled using 3-D computer graphics. If the reverse mapping is impractical to compute, forward mapping techniques must be employed. Texture splatting is one such method [45] that can be applied to both 3-D computer graphics problems and 2-D image warping problems. Splatting performs two roles; it carries over pixel values from the source to the destination image and it interpolates between mapped pixels in the destination image. This is achieved by carrying over small patches of colour rather than individual pixels (i.e. not a one to one pixel mapping). Various patch shapes have been used, most notably ellipses [40, 102] although methods based on lines, triangles and rectangles have also been implemented.

In the remainder of this chapter, an efficient model-based warping method is described that utilizes the constraints imposed by the appearance model and the limits of natural face shape variation. These constraints can be summarized as follows,

1. Source images are all in shape-normalized form such that a *correspondence* exists between the k^{th} pixel in every base image.

2. The areas of the convex hulls of the face shapes do not vary very much



(a) source

(b) destination

Figure 5.1: Affine image warp from source image (a) to destination image (b)

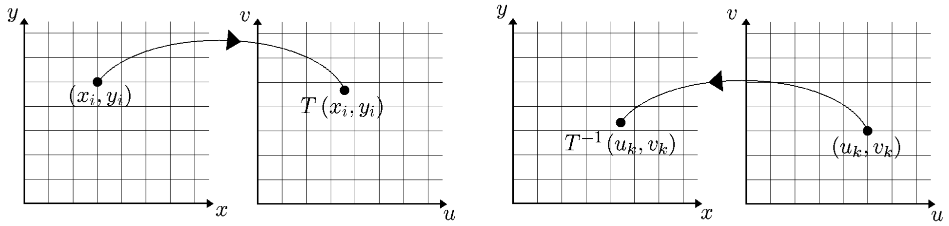
due to the fact that scaling variations are filtered out in the alignment procedure and large deformations from the mean shape are not present in the face pattern class.

3. The shape variation can be expressed adequately as a finite, linear combination of modes representing deviations from the mean shape.

In the following section, the procedure for constructing a displacement field is described and illustrated with the aid of a simple example. In section 5.4 an image warp is effected by the superposition of two displacement fields. The efficiency of the displacement field method is measured against the standard piece-wise affine/reverse mapping method and a two pass splatting method based on a naive implementation of Heckbert's [45] method. Finally the displacement field method is applied to the specific problem of warping faces in which each displacement field corresponds to a PCA mode of shape variation.

5.2 Forward mapping vs reverse mapping

When referring to geometric transformations, the spatial locations in the original image (prior to the warp) as described as the source coordinates (x, y) whereas the locations in the resultant image (after the warp) are described as the destination coordinates (x', y'). Image warping is thus a geometric operation that defines a coordinate mapping from the source image to the destination image and assigns the intensity values from corresponding locations accordingly. There are, however, two ways of considering this process - if we take each coordinate location in the source and calculate its corresponding location in the destination image, we are considering a *forward mapping* from source to destination. Conversely, we may consider a *reverse mapping* of destination-to-source, in which we successively consider each spatial location in the warped (destination) image and calculate its corresponding location in the source (see figure 5.2).



(a) **Forward map:** The transformation T maps the point (x, y) in the source image to the point $T(x, y)$ in the destination image.

(b) **Reverse map:** The inverse transformation T^{-1} maps the point (u, v) in the destination to the point $T^{-1}(u, v)$ in the source image.

Figure 5.2: Grid nodes represent integer pixel coordinates. In general, the transformed coordinate pairs $T(x, y)$ and $T^{-1}(x, y)$ are not integer values.

Forward mapping in which pixels are *carried over* from source to destination has several problems associated with it. In particular, depending on the specific transformation defined, pixels in the source may be mapped to positions beyond the boundary of the destination image. Also, some pixels in the destination may be assigned more than one value whereas others may not have any value assigned to them. The unassigned pixels are particularly problematic because they appear as holes in the destination image that are aesthetically unpleasing. In practice, a pixel filling algorithm is often used instead [17]. Unlike the carry over approach, a pixel filling algorithm determines the reverse mapping from destination to source. Using this method, every pixel in the destination is mapped to a single position with rational coordinates within the source. In general, the mapping will send each pixel to a position that lies between the centroids of four surrounding source pixels. The corresponding point in the destination image is generally assigned a value according to an interpolation rule. The fastest interpolation method is to assign the value of the closest pixel. This is known as *nearest neighbour* interpolation or *zero order* interpolation. Bilinear interpolation yields more accurate results but requires a hyperbolic paraboloid to be fitted to the four closest pixels, a calculation that requires more computational time than the nearest neighbour scheme.

Consider figure 5.3. Here the interior pixel coordinates belonging to the triangle in the source image (a) are forward mapped resulting in destination

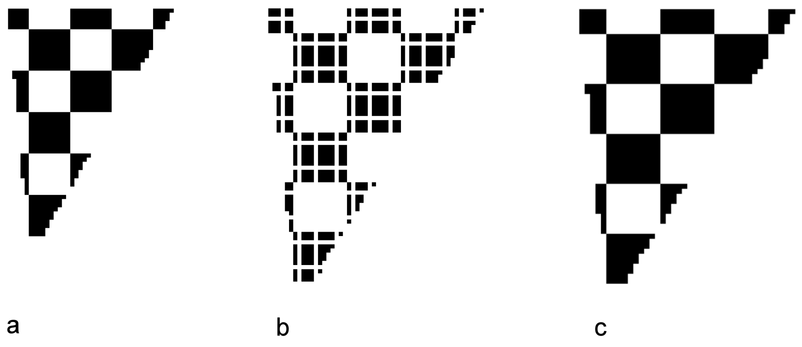


Figure 5.3: For the reverse mapping method, every pixel in the destination image is assigned a single intensity value. The forward mapping method does not guarantee that every pixel in the destination image is assigned a single value. This may result in holes in the destination image. In the figure above (a) is a source image, (b) a destination image formed using the forward mapping method and (c) a destination image formed using the reverse mapping method.

image (b). Artefacts appear when the transformation causes an expansion in the vertical direction, horizontal direction, or both directions simultaneously.

5.3 Displacement field construction

In a general image warping problem, a transformation is sought that attempts to map one set of control points (shape) to another,

$$T(\bar{\mathbf{x}}) = \mathbf{x} \tag{5.1}$$

In contrast, the displacement field method proposed here makes use of condition 1, namely that the base shape (normally related to the source image), $\bar{\mathbf{x}}$ is known[1] and is constant for every possible image warp and that the destination shape $\mathbf{x}$ represents a statistically likely deformation of $\bar{\mathbf{x}}$ that can also be established prior to performing the warp (condition 3). This means that the transformation can be pre-determined and hence removed from the real-time computation required by the warp. More importantly, the computation required to apply this transformation to each and every pixel coordinate pair can also be calculated off-line. The procedure for constructing a displacement field is outlined by the steps below and illustrated by the simple example provided in figures 5.4(a)-5.5. Here, a piece-wise affine transform has been used to

[1] Conversely, if the destination shape was constant a more desirable reverse mapping would be possible but unfortunately this is not the case in the application considered in this work.

determine the displacement field, although other warping methods are equally applicable.

Off-line computation

1. Consider the regularly spaced grid illustrated in figure 5.4(a). Let each node in the grid represent a pixel location within a $M \times N$ digital image, I_s, and let $\bar{\mathbf{X}}$ and $\bar{\mathbf{Y}}$ be 2D arrays containing the x and y pixel coordinates respectively. A convenient representation is provided, using complex notation,

$$\bar{\mathbf{Z}} = \bar{\mathbf{X}} + \bar{\mathbf{Y}}i \tag{5.2}$$

2. Define $\bar{\mathbf{x}} = [\bar{x}_1\ \bar{x}_2\ \ldots\ \bar{x}_n\ \bar{y}_1\ \bar{y}_2\ \ldots\ \bar{y}_n]^T$ as a base shape, such that it occupies the entire area of image I_s.

3. Let $\mathbf{x}$ define a new shape representing a (constrained) deviation from the base shape such that $\mathbf{x}_1 = \bar{\mathbf{x}} + \Delta\mathbf{x}_1$.

4. Compute the Delaunay triangulation of the convex hull defined by $\bar{\mathbf{x}}$ thereby forming a mesh. Use the same order of triangulation to segment $\mathbf{x}$ into corresponding triangular regions (see figure 5.4(b)).

5. Let $\bar{\mathbf{v}} \subset \bar{\mathbf{x}}$ be a vector containing the coordinates of the vertices of a single triangle in the mesh defined by $\bar{\mathbf{x}}$. Let $\mathbf{v} \subset \mathbf{x}$ be vector of vertex coordinates in the corresponding triangle of the mesh defined by $\mathbf{x}$. Using equation 2.45 from Chapter 2, section 2.3 determine the transformation T that maps the coordinates in $\bar{\mathbf{v}}$ to the coordinates in $\mathbf{v}$ (see figure 5.4(b)) such that,

$$T\left(\bar{\mathbf{v}}\right) = \mathbf{v}_1 \tag{5.3}$$

6. Apply the transformation T to all pixel coordinate pairs, $\{\bar{\mathbf{z}}_i\}$ (where $\mathbf{z} = x + yi$), located within the triangle specified by the vertices $\bar{\mathbf{v}}$, thereby mapping them to new locations, $\{\mathbf{z}_i\}$, within the triangle defined by $\mathbf{v}$.

7. Compute the displacement vector $\Delta\mathbf{z} = \mathbf{z} - \bar{\mathbf{z}}$ of every pixel in the source image resulting from transformation T.

8. For every pixel, separate $\Delta\bar{\mathbf{z}}$ into x and y components. Insert the x component into the real part of a complex 2D array, $RE\left[\Delta\mathbf{Z}\right]$, at location

$\bar{\mathbf{z}}$. Similarly insert the y component into the imaginary part, $IM\left[\Delta\mathbf{Z}\right]$, at location $\bar{\mathbf{z}}$.

9. Repeat steps 5-8 for all corresponding triangles in the meshes defined by the shapes $\bar{\mathbf{x}}$ and $\mathbf{x}$ and in doing so construct the *displacement field* $\Delta\mathbf{Z}$ (see figure 5.5).

10. Write displacement field $\Delta\mathbf{Z}$ to file so that it may be loaded into memory when required.

Using the same procedure, displacement fields representing other distinct deformations of the base shape can also be determined, forming the set $\{\mathbf{Z}_i\}$. The procedure demands that every synthesized image has the same dimensions as the displacement fields.

5.4 Superposition of displacement fields

The previous section described the off-line process for constructing a displacement field. This section explains how an image warp can be effected by a linear combination of t distinct, pre-determined, displacement fields. The objective is to allow the t warps defined by the pre-computed displacement fields to be accomplished with minimal computational expense *and* to allow intermediate warps to be determined by weighting and adding the displacement fields as required. Hence a wide range intermediate warps can be generated from a relatively small number (condition 2) of predetermined warps as described by the steps below and illustrated in figures 5.6(a)-5.6(c). As with any forward mapping method, holes are liable to appear in the destination image due to undefined pixel values. However, because intra class variations in faces shape are small the occurrence of these artefacts is limited and they can be removed using a standard, order-statistic filtering method [38].

Real-time computation

1. Initialize destination image $I_d = \mathbf{0}_{M\times N}$, where $\mathbf{0}_{M\times N}$ is a 2D array in which each element is set to zero.

2. Load displacement fields $\{\Delta\mathbf{Z}_i\}$ into memory.

3. Add the required linear combination (superposition) of displacement fields, $\{\Delta\mathbf{Z}_i\}$, to the grid of regularly spaced pixel coordinates, $\bar{\mathbf{Z}}$, thereby

determining a forward mapping of pixel coordinates.

$$\mathbf{Z} = \bar{\mathbf{Z}} + \sum_{i}^{t} \Delta \mathbf{Z}_i b_i \tag{5.4}$$

where b_i is a scalar value that determines the contribution of the i^{th} displacement field to the forward mapping.

4. In general, the coordinate pair represented by the complex number $\mathbf{z}$ will consist of non-integer values. Hence, for every $\mathbf{z}$ determine the nearest integer pixel coordinates (nearest neighbour interpolation). Fill in pixel values in the destination image according to the following relationship,

   $$I_d\left(\mathbf{z}\right) = I_s\left(\bar{\mathbf{z}}\right) \tag{5.5}$$

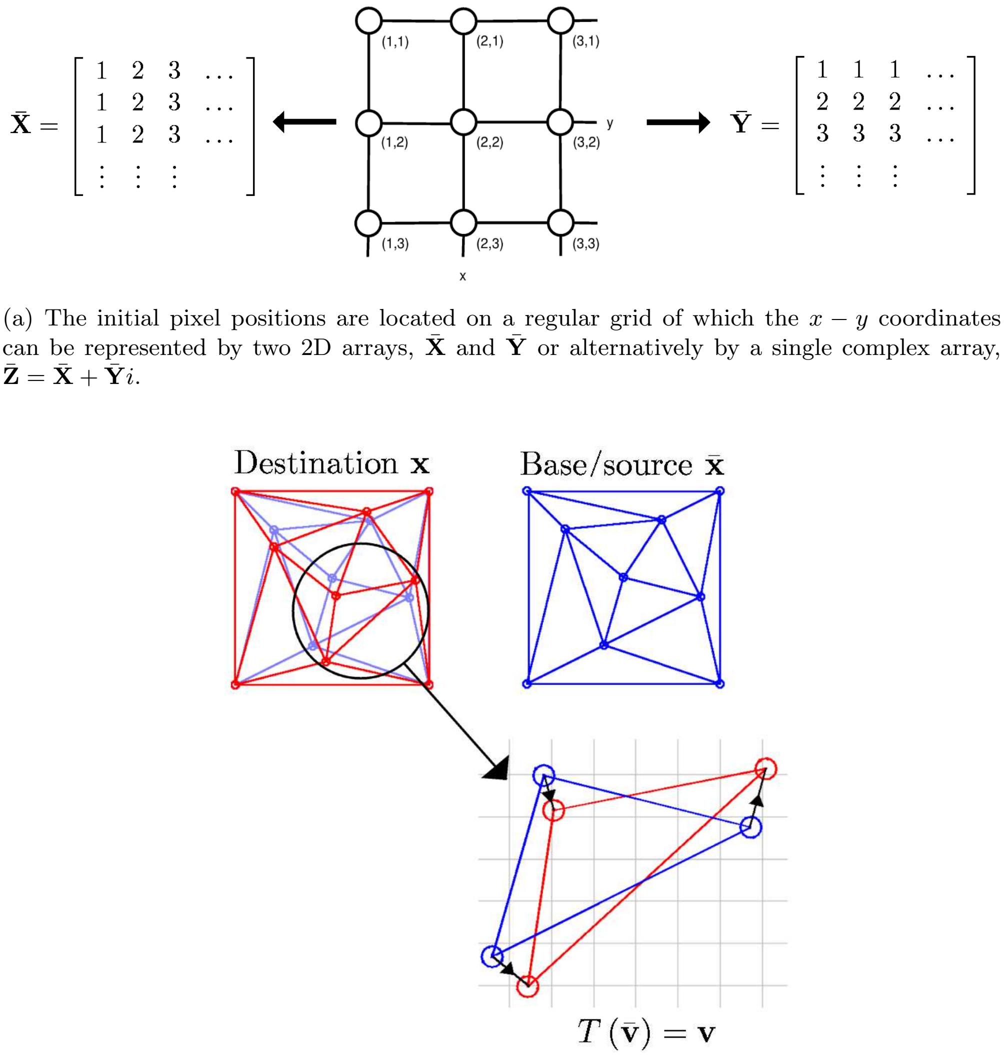
   Note that the coordinate pairs $\{\mathbf{z}_i\}$ are obtained via a column-wise linear index into $\mathbf{Z}$,

   $$\mathbf{z}_{(r+M[c-1])} = RE\left[\mathbf{Z}\left(r,c\right)\right], IM\left[\mathbf{Z}\left(r,c\right)\right] \tag{5.6}$$

5. The removal of the holes from the image is achieved in 2 steps. Firstly, the entire image is filtered using a median filter. This fills the holes but induces a modest loss of resolution throughout the image that may be perceived as blurring. To mitigate this effect, the forward mapping of values from source to destination is repeated thereby replacing all the filtered values by the original values (back fill), except at the locations of the holes, as desired.

5.5 Comparison with other image warping methods

In the previous section, a simple example was used to illustrate the warping of a digital image by the superposition of two displacement fields. To test the performance of this method, the same image warp was performed using a piece-wise affine method (with reverse mapping and nearest neighbour interpolation) and a two pass pixel splatting method, adapted from Heckbert's [45] original method (see Appendix E). All three methods were coded in MATLAB. The superposition method was found to be faster than the reverse mapping and considerably faster than the splatting algorithm (see table 5.5).



$$\bar{\mathbf{X}} = \begin{bmatrix} 1 & 2 & 3 & \dots \\ 1 & 2 & 3 & \dots \\ 1 & 2 & 3 & \dots \\ \vdots & \vdots & \vdots & \end{bmatrix}$$

$$\bar{\mathbf{Y}} = \begin{bmatrix} 1 & 1 & 1 & \dots \\ 2 & 2 & 2 & \dots \\ 3 & 3 & 3 & \dots \\ \vdots & \vdots & \vdots & \end{bmatrix}$$

(a) The initial pixel positions are located on a regular grid of which the $x - y$ coordinates can be represented by two 2D arrays, $\bar{\mathbf{X}}$ and $\bar{\mathbf{Y}}$ or alternatively by a single complex array, $\bar{\mathbf{Z}} = \bar{\mathbf{X}} + \bar{\mathbf{Y}}i$.

(b) For each pair of corresponding triangular regions a transformation T is defined that maps the vertices of the source triangle $\bar{\mathbf{v}}$ to the vertices of the destination triangle $\mathbf{v}$. The interior pixel coordinates are located at the nodes of the grid in the above image.

Figure 5.4: The displacement field method requires that the dimensions of the source and destination images are the same and that their meshes (shown above) cover the entire image plane. Blue has been usedd to denote a source/base shape and red has been used to denote a destination shape.

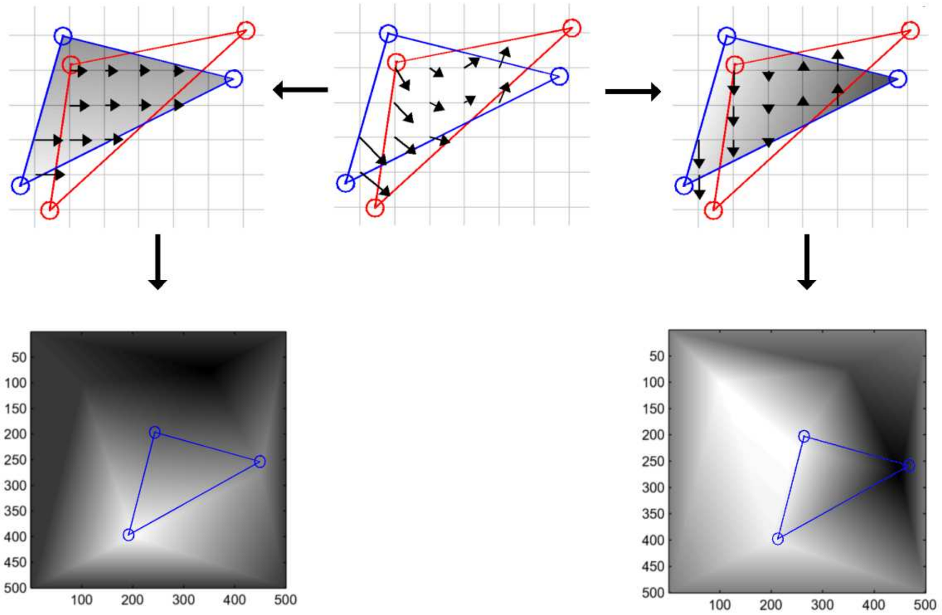


Figure 5.5: Individual displacement vectors $\{\Delta\mathbf{z}_i\}$ ($\Delta\mathbf{z} = \Delta x + \Delta y i$) are located at positions $\{\bar{\mathbf{z}}_i\}$ and indicated by small black arrows of varying length. An $M \times N$ complex array containing all such vectors, is referred to as a displacement field and denoted by $\Delta\mathbf{Z}$. Images of the $RE\,[\mathbf{Z}] = \mathbf{X}$ and $IM\,[\mathbf{Z}] = \mathbf{Y}$ arrays are illustrated above, where intensity of each pixel indicates its displacement from the regular grid defined by $\bar{\mathbf{Z}}$. White pixels indicate a large downward displacement or displacement to the right for $IM\,[\mathbf{Z}]$ and $RE\,[\mathbf{Z}]$ respectively. Black pixels indicate a large upward displacement or displacement to the left for $IM\,[\mathbf{Z}]$ and $RE\,[\mathbf{Z}]$ respectively.

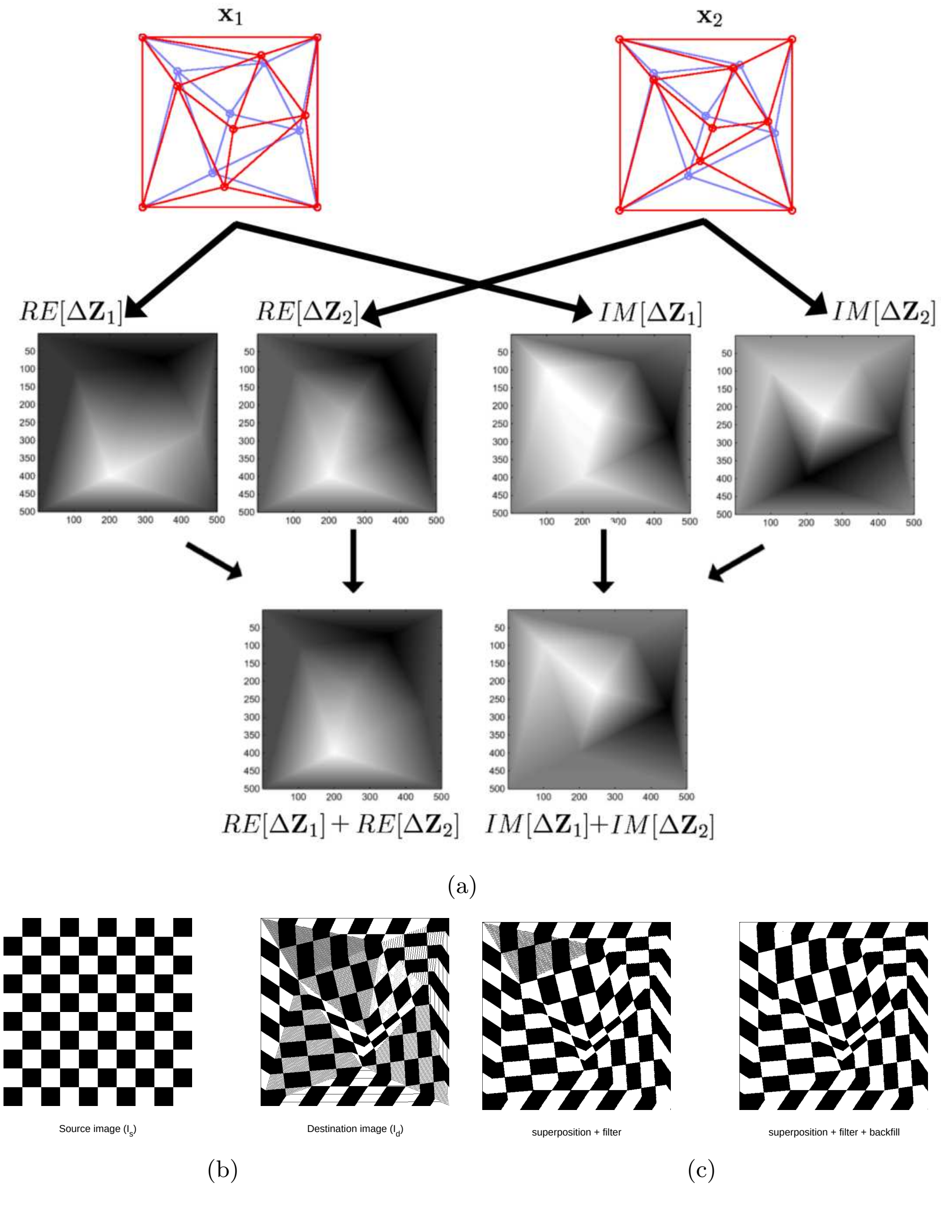
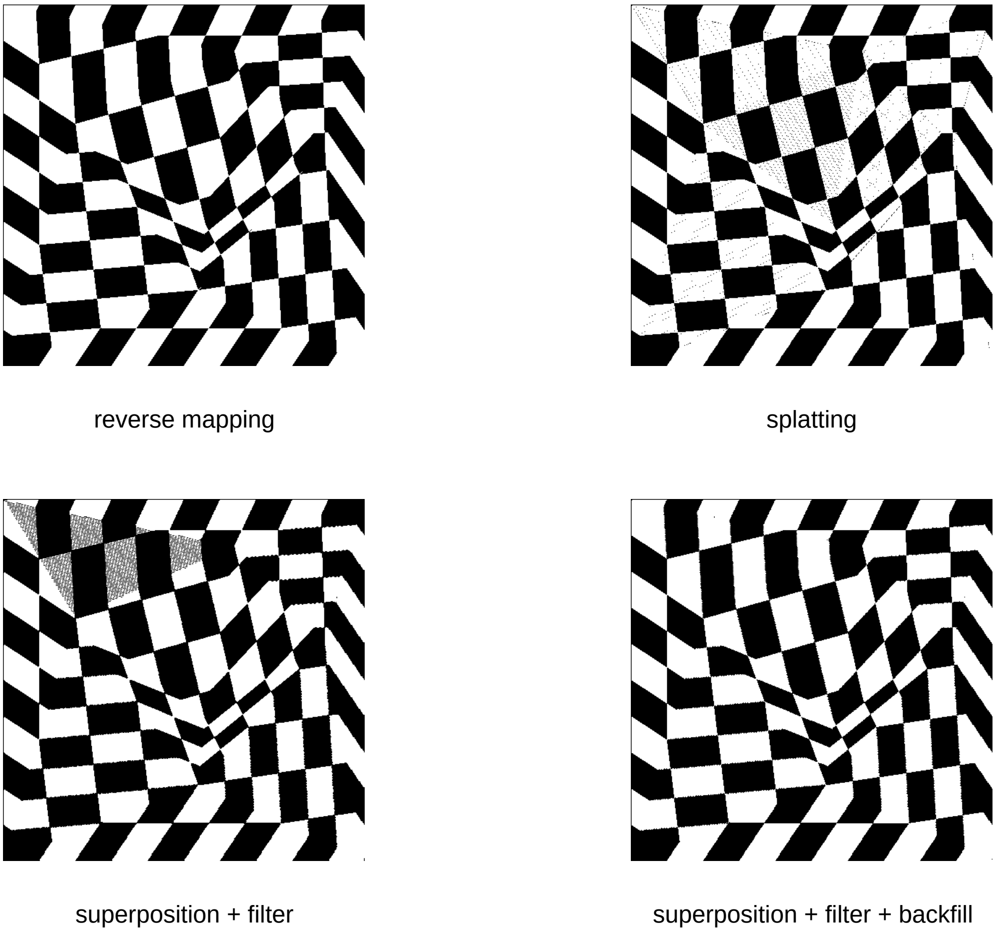


Figure 5.6: Forward pixel mapping by a superposition of two displacement fields $\Delta\mathbf{Z}_1 b_s^1 + \Delta\mathbf{Z}_2 b_s^2$ where $b_s^1 = b_s^2 = 1$. Each pixel intensity value in the destination image is assigned by interpolating the mapped coordinate pair to the nearest integer values (nearest neighbour interpolation) and copying the intensity value from the corresponding pixel in the source image. Note that



reverse mapping

splatting

superposition + filter

superposition + filter + backfill

Figure 5.7: Visual comparison of the three different warping methods. Holes are visible in the image created by the superposition method, although these can be removed by applying an appropriate image filter to the destination image and then filling in the known pixels determined by the forward warp.

Even with the additional hole removal procedure, the superposition method was still quicker than the reverse mapping method. A measure of the accuracy of each warp was obtained by determining the rms pixel error, taking the reverse mapping warp as the bench mark. The accuracy results are provided in table 5.5. Although the rms errors for the splatting method and the superposition + hole removal method are very similar, the destination image corresponding to the latter method generated a more visually pleasing result. On close inspection of figure 5.7, small artefacts can be seen on the boundaries between white and black polygon regions in the 'superposition + filter + backfill' image. In the equivalent 'splatting' image, noise appears as black specks in a somewhat random pattern.

Quantitative comparison of warping methods relating to figure 5.7				
Warp method	reverse mapping	superposition method	superposition + filter + back fill	splatting method
Computation time(s)	1.192	0.43	0.43+0.08+0.07	185.6970
rms pixel value error	0	0.181	0.092	0.091
Image: 340 × 267, 8bit RGB				
Hardware: fujitsu-siemens, lifebook s, Pentium M 1400MHz processor with 776,688 KB RAM				
Software: MATLAB 6.5 on Windows 2000				

Table 5.1: Comparison of coordinate mapping methods. The superposition method was fastest despite the additional processing required to remove holes in the destination image.

5.6 Model based displacement field method

The efficiency of the displacement field method for image warping has been demonstrated using a naive example. Here the method is extended to incorporate the modes of natural shape variation determined by a PCA model of the human face. We begin by reviewing the method described in Chapter 2/3 for generating a new instance of a pattern from the modelled pattern class.

1. In the application described in this thesis an evolutionary algorithm is used to generate new sets of appearance model vectors (in our implementation a set comprises nine such vectors) $[\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n]$.

2. Each appearance model vector $\mathbf{c}$ is then decoupled into its associated shape parameter vector $\mathbf{b}_s$ and texture parameter vector $\mathbf{b}_g$.

3. The shape coordinates $\mathbf{x}$ of each face in the new generation are calculated as a linear sum of the shape principal components.

$$\mathbf{x} = \mathbf{P}_s\mathbf{b}_s + \bar{\mathbf{x}} \tag{5.7}$$

4. The shape-normalised texture of each face in the new generation is calculated as a linear sum of the texture principal components.

$$\mathbf{g} = \mathbf{P}_g\mathbf{b}_g + \bar{\mathbf{g}} \tag{5.8}$$

5. Finally, the shape-normalized texture of each face in the new generation is geometrically transformed (i.e. warped) to correspond to the required face shape coordinates and then displayed.

$$I_d\left(T^{-1}(u,v)\right) = I_s(u,v) \tag{5.9}$$

   This step causes a *computational bottle-neck* in the process of generating a new example of a pattern/face.

Chapter 3 demonstrated that any face shape can be approximated using a linear combination of a relatively small number (t_s) of shape principal components. Therefore, it is reasonable to assume that a *finite* set of $t \leq t_s$ image warps can also be defined that allow a face image to be warped quickly from its shape normalized form to any plausible face shape. Accordingly a set of displacement fields have been defined in which each displacement field corresponds to an eigen-mode of shape variation. A superposition of these model-based displacement fields can be used to replace steps 3 and 5 in the face generation process described above, thereby reducing the size of bottle-neck by calculating the image warp implicitly.

5.6.1 Construction of model-based displacement fields

The procedure outline in section 5.3 was used to construct t displacement fields that enabled any face shape to be approximated via an efficient warping method. Here the base shape $\bar{\mathbf{x}}$ is the mean face shape (plus image corner landmarks), and the destination shape $\mathbf{x}_i$ is defined by a one standard deviation perturbation from the mean according to the mapping,

$$\bar{\mathbf{x}} \mapsto \mathbf{p}_s^i \left(\lambda^i\right)^{\frac{1}{2}} + \bar{\mathbf{x}} \tag{5.10}$$

where $\mathbf{p}_s^i$ is the i^{th} shape principal component. If I_s is a $M \times N$ source image containing a shape normalized face, then a piece-wise affine mapping of its pixel coordinates governed by equation 5.10 can be used to construct the i^{th} displacement field $\Delta\mathbf{Z}_i$. Hence the displacement field $\Delta\mathbf{Z}_i$ is the image warp analogue of the shape principal component $\mathbf{p}_s^i$.

5.6.2 Superposition of model-based displacement fields

Using the procedure described in section 5.4 a superposition of displacement fields is obtained as follows,

$$\mathbf{Z} = \bar{\mathbf{Z}} + \sum_i \Delta\mathbf{Z}_i b_s^i \tag{5.11}$$

where $\mathbf{Z}$ defines a forward mapping of pixel coordinates, $\bar{\mathbf{Z}}$ represents a grid of regularly spaced pixel coordinates and b_s^i is a model parameter dictating the influence of the i^{th} displacement field on the image warp. By comparing equation 5.11 with the equation below,

$$\mathbf{x} = \bar{\mathbf{x}} + \sum_i \mathbf{p}_s^i b_s^i \tag{5.12}$$

it can be seen that $\Delta\mathbf{Z}_i$ is analogous to $\mathbf{p}_s^i$ and (b_s^i) are in fact the PCA shape model parameters in both cases.

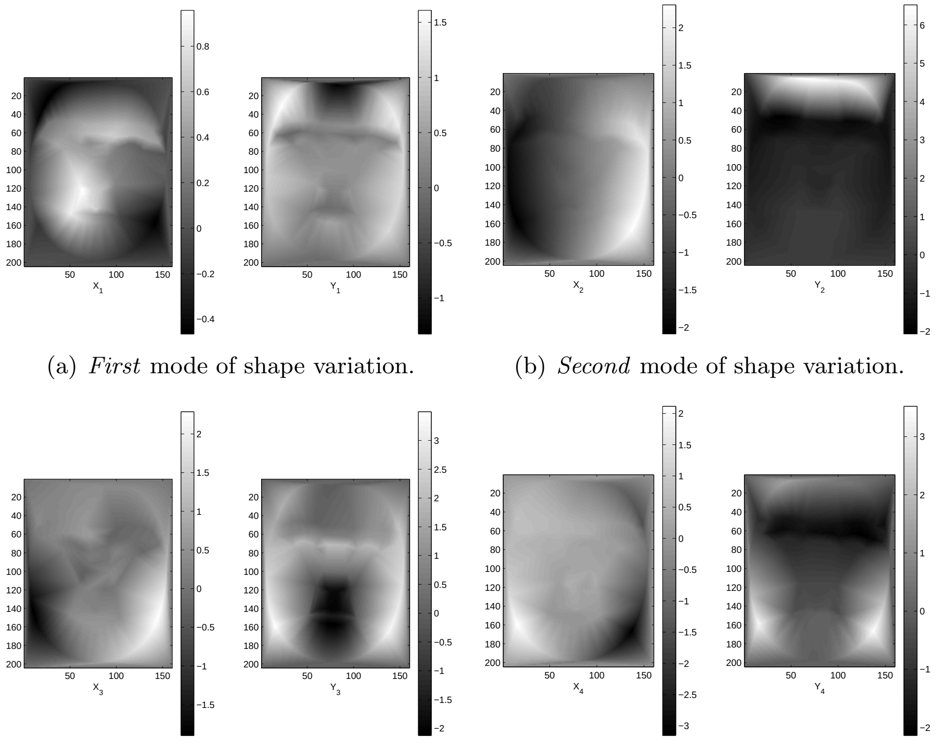
The displacement fields corresponding to the first four eigen-modes of shape variation are illustrated in figure 5.8. Once the superposition has been formed the pixel intensity values in the destination image can be assigned according to,

$$I_d\left(\mathbf{z}\right) = I_s\left(\bar{\mathbf{z}}\right) \tag{5.13}$$

To avoid holes appearing the resulting destination image, I_d, was enhanced using a 3×3 median filter. Figure 5.9 shows typical results obtained using the model-based displacement fields method.

5.6.3 Application to facial composites

An example of a face image that has been warped using a superposition of 30 displacement fields corresponding to the first 30 PCA modes of shape variation is shown in figure 5.9. Using more displacement fields increases the computation time, due to the time taken to load the displacement fields, one by one, from the hard disk into RAM. This bottleneck can be avoided by pre-loading the displacement fields, or loading them in blocks of ten or more. The number of displacement fields that can be held in memory will depend on the image size and the capacity of RAM. The computational burden of loading the displacement fields is less apparent when more than one image is generated because the displacement fields are only loaded once for the set and not once for each face image. For example, in the facial composite application described in this thesis, faces are generated in sets of nine. Hence the computational cost per image of loading the data is only $1/9^{th}$ of that for a single face.



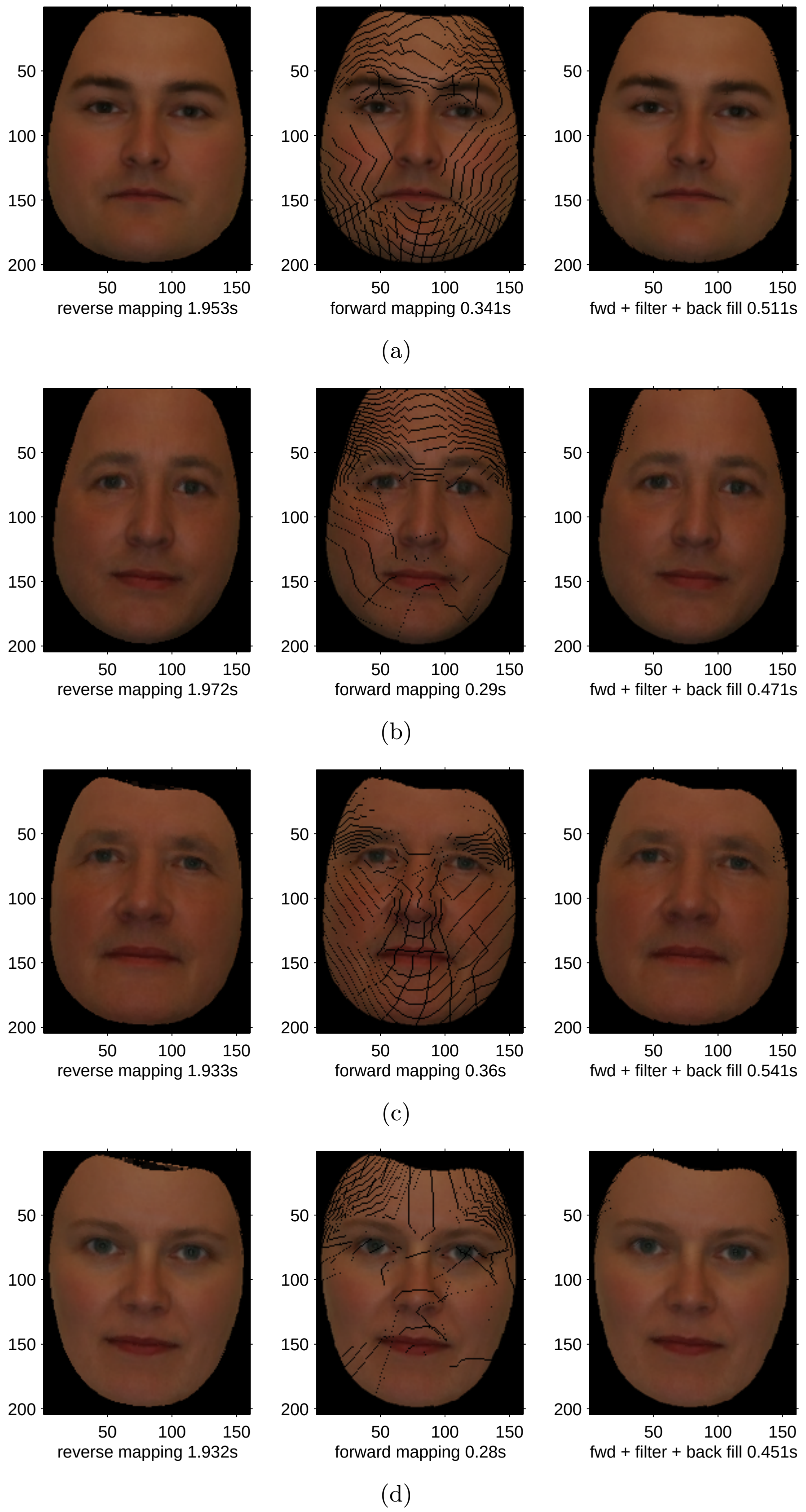
(a) *First* mode of shape variation.

(b) *Second* mode of shape variation.

(c) *Third* mode of shape variation.

(d) *Fourth* mode of shape variation.

Figure 5.8: The real ($\Delta\mathbf{X}$) and imaginary ($\Delta\mathbf{Y}$) components of the model-based displacement fields corresponding to the first four modes of shape variation.



(a)

(b)

(c)

(d)

Figure 5.9: A visual comparison between the reverse mapping method and the superposition of displacement fields (implicit forward mapping) method. In these examples 30 204 × 160 displacement fields were used corresponding to the first 30 modes of shape variation.

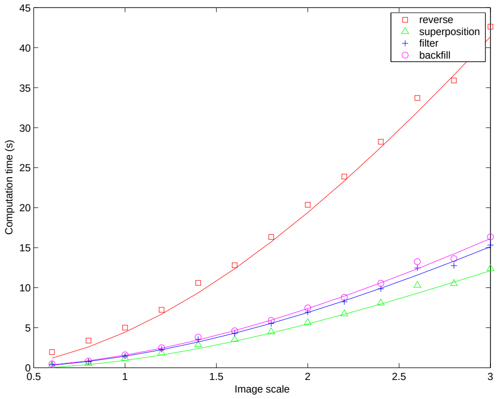


Figure 5.10: Time required to warp a single image using 30 eigen-modes as a function of image size.

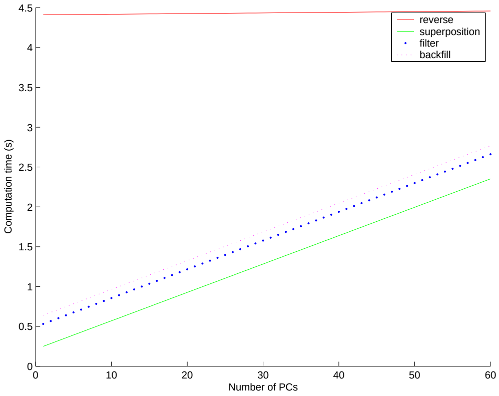


Figure 5.11: The relationship between number of fields and time to compute mapping

5.6.4 Scalability

The problem size increases approximately as the square of a scaling factor with respect to a 340×267×3 RGB image (see figure 5.10). When a face is warped using a linear combination of thirty displacement fields, the superposition method is significantly faster. The plots shown in figure 5.10 do not take into account the time required to load the displacement fields into memory. The assumption has been made that there is sufficient physical memory available such that all the required displacement fields can be loaded prior to warping. This is a reasonable assumption to make for the size of images required in the composite application. The time required to compute a forward mapping using the superposition method is linearly dependent on the number of displacement fields used. Figure 5.11 shows the experimentally obtained relationship between the number of displacement fields used (340 × 267 × 3) and the time taken.

5.7 Summary

A model-based warping procedure has been described and implemented and compared to the standard (reverse mapped) piece-wise affine method described in Chapter 2. Under certain conditions in which the warp corresponds to modest deviations from the mean shape and can be adequately approximated by a superposition of eigenmodes, the model-based procedure has a superior performance. The method employs a set of pre-defined pixel displacement fields, a superposition of which yields a 2D coordinate map enabling pixel values in the destination image to be assigned. In this work, the displacement fields were constructed using the piece-wise affine method but the approach could easily be extended to other generating transforms such as thin-plate splines thereby providing smoother image warps at no additional computational cost.

As expected, the relative efficiency of the displacement field method was shown to decrease linearly with the number of eigen-modes (displacement fields) used. For the specific application to facial composites in this thesis, a relatively large number of shape eigenmodes has been considered. Using 30 eigenmodes to approximate the shape still resulted in an overall increase in computational speed of a factor of 3 without significantly compromising image quality. For other applications, to which this technique is equally applicable in principle, the gain may be much greater and the results presented herein suggest that an application requiring only 10 eigenmodes to represent the warp could be achieved a factor of 8 times more quickly.

Chapter 6

Appearance models and facial caricatures

Although producing caricatures is artistically a somewhat subjective process, a previous attempt has been made to determine a mathematical formula which mimics the effect. This mathematical formula, which will be referred to in this work as the *uniform caricature* method, is described in this chapter and implemented using the appearance model framework. Although the uniform approach would seem to provide a satisfactory technique for generating caricatures, no prior work exists in which it has been proven to be the definitive method. The main objective of this chapter is to investigate the new concept of *non-linear caricatures* and how the established uniform method for caricature is in fact, only a special case of a more general paradigm. Examples of photographic quality caricatures are presented, that were generated by perturbing the appearance model parameters, corresponding to a veridical face image, according to different transformation functions. The results of a simple experiment are presented which suggest that non-linear transformations such as those we propose can accurately capture key aspects of the caricature effect.

6.1 Introduction

The primary purpose of the facial composite process is to generate an image which has the maximum probability of triggering recognition by one or more witnesses. It is important to differentiate this from the closely related (but nonetheless distinct) goal of achieving the most ”realistic” rendition of the subject. Thus, if it were possible to produce an image which did not correspond exactly to how the suspect appears in reality but evoked recognition more easily or effectively than the veridical image, we should prefer the former. In this context, the caricature effect, in which prominent characteristics of an individual face are exaggerated (often grossly) is of considerable interest. The caricature effect is a remarkable phenomenon because, despite huge distortions in the appearance of the subject, human-beings are still able to effectively recognise the individual portrayed and, indeed, some psychological studies have even suggested [76, 86] that caricatures are more effectively recognized than the veridical images from which they were generated.

The scientific basis for the caricature effect is still far from being fully understood. Traditionally, caricature has been the domain of the artist who produces them primarily for comic or satirical effect. In qualitative terms, the artist identifies facial features or characteristics which deviate significantly from the norm or average and systematically exaggerates those deviations. However, the precise manner and degree of exaggeration shows considerable variation between individual artists and the process is to some degree specific to both the subject and the artist. A skilfully drawn caricature may even capture characteristics of the subject's personality - this is especially true when the caricature is satirical in nature.

6.2 Motivation

These observations suggest that caricatures may in some sense capture the essential aspect of identity and even enhance it. If this hypothesis has some basis in fact and a means to automatically manipulate facial images to produce caricature effects can be established, it is possible that a composite system which allows the caricature effect to be produced may be faster and more effective. Psychological tests by Frowd et al [34], investigating the possible benefits to facial composite production, would seem to suggest that the caricature effect is beneficial to the composite process under certain conditions. Since the interesting concept of caricatured composites is new, its effectiveness remains unproven in real composite situations. Therefore, the work presented in this chapter is somewhat speculative in relation to the composite system described in Chapter 3 and Chapter 4. The aim here is to provide the groundwork for a caricaturing tool that may be prove useful in the future.

6.3 Current approach to caricature

Brennan [11] is generally attributed with the first automatic caricature generator. This worked on line drawings, constructed from connected point configurations, outlining the shape of the main facial features. The generation of near photo-realistic caricatures was first developed by Benson and Perrett [4] and subsequent authors have followed their basic approach [14, 72]. Benson and Perrett [4] considered the shape and texture characteristics of the images separately. O'Toole et al [68] applied the standard caricaturing algorithm used by Benson and Perrett to 3D representations of the human face. Their results suggested that the caricaturing procedure may also increase the perceived age of the face. In order to understand clearly the standard method for computer generation of caricatures and its relation to that proposed herein, we will describe their approach in detail.

1. Facial landmarks corresponding to key points are identified on each image in a sample of face images. The corresponding set of (scaled and aligned) Cartesian coordinates $\{x_i, y_i\}$ for each image constitutes a shape vector for the given face, which we denote by $\mathbf{X}$. The mean shape vector over the sample, $\bar{\mathbf{X}}$ is calculated.

2. Each image is warped to the mean shape vector of the entire sample by standard Delaunay triangulation methods. We refer to such warped images as the shape-normalized texture patterns, in which the colour values are stored as the elements of a vector $\mathbf{T}$. The average of the shape-normalized texture patterns is termed the facial prototype, $\bar{\mathbf{T}}$.

3. To generate a caricature of any face with texture $\mathbf{T}$ and shape $\mathbf{X}$, we (1) calculate the texture difference vector $\Delta\mathbf{T} = \mathbf{T} - \bar{\mathbf{T}}$ between the shape-normalized texture pattern and the texture of the facial prototype, (2) calculate the shape difference vector $\Delta\mathbf{X} = \mathbf{X} - \bar{\mathbf{X}}$ between the shape vector of the face and the mean shape vector corresponding to the facial prototype, (3) add some linear multiple of the texture and shape difference vectors, $\mathbf{T}' \rightarrow \mathbf{T} + a\Delta\mathbf{T}$ and $\mathbf{X}' \rightarrow \mathbf{X} + b\Delta\mathbf{X}$, where a and b are the boost parameters, and (4) finally, warp the texture map $\mathbf{T}'$ to the required shape $\mathbf{X}'$, thereby forming the caricatured face image.

The procedure is represented diagrammatically in figure 6.1 in which the difference between a veridical point relating to a subject, and the prototype (typically relating to the sample mean face) is exaggerated by a small collinear perturbation. Note that the prototype does not necessarily lie at the origin, although in our own work, we have constructed a vector space in which the mean face prototype does lie at the origin. A key point to note about this method is its linear (in fact, uniform) treatment of all local deviations from the prototype. Thus, in Benson and Perrett's [4] approach, all differences in local pixel values

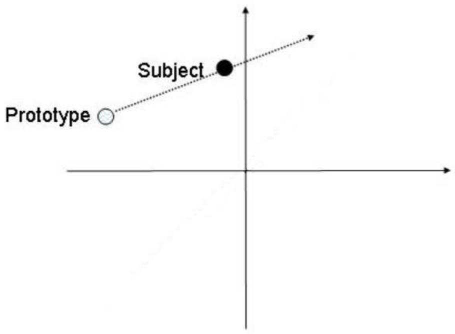


Figure 6.1: Schematic depicting the uniform (linear) caricature approach (as per Benson & Perrett, 1991). The difference vector between the veridical (representing the original image of the subject) and the prototype is simply extended collinearly. The diagram can be interpreted as a small perturbation in the shape of the veridical face or a small perturbation int the texture of the veridical face.

between the veridical and the prototype are multiplied by an identical factor (a for the texture and b for the shape). This model for caricaturing thus effectively gives all directions equal importance. In anticipation of alternative methods, we term the currently accepted model for caricature the *uniform method.* In the next section, we will present other methods of caricature (within which the uniform method is simply a special case), suggesting that other mathematical forms may be appropriate in creating caricatures that maintain and even enhance recognition capacity.

6.4 Caricatures in appearance space

All information regarding facial appearance can expressed in a vector of appearance parameters $\mathbf{c} = [c1, c2, \ldots, c_n]^T$. Recall that in Chapter 3, an appearance model of the human face was described in which the appearance parameters were distributed as independent normal variations[1]. It follows that the likelihood of a given face occurring in the population can be measured simply as the scaled distance from the origin. Specifically, the log of the probability density for an appearance vector $\mathbf{c}$ is given by $-\log p = L$, where

[1] In this chapter it is assumed that p can be modelled as a single multivariate normal distribution. Although p can be accurately modelled using a mixture of multivariate normal distributions (see Chapter 3), the simplicity of the single multivariate distribution is preferred in this chapter

$$L = \sum_{i=1}^{N} \frac{c_i^2}{\sigma_i^2} \tag{6.1}$$

p is the normal density function and σ_i^2 is the variance associated with the i^{th} axis in the appearance parameter space. Thus, uniform caricaturing moves a face to a region in appearance space where faces are statistically less likely to occur (i.e., of lower exemplar density), but crucially, the shift is precisely along that original direction that minimizes the exemplar density. Over a suitable sample of faces, a prototype appearance vector $\bar{\mathbf{c}}$ is easily calculated, and a uniform caricature $\mathbf{c}'$ is created by the simple transformation

$$\mathbf{c}' = \mathbf{c} + \gamma \mathbf{I} \left(\mathbf{c} - \bar{\mathbf{c}} \right) \tag{6.2}$$

where

$$\bar{\mathbf{c}} = \sum_{i=1}^{n} \mathbf{c}_i \tag{6.3}$$

γ is a scalar boost parameter and $\mathbf{I}$ denotes the identity matrix. The reconstruction of the caricatured face from the appearance vector is then obtained by applying equations 2.66-2.69 from Chapter 2 to $\mathbf{c}'$, and warping to the required shape. If different boosts γ_S, γ_T are required for the shape and texture components, this is easily achieved through the decoupled shape and texture appearance parameters $\mathbf{b}_S$ and $\mathbf{b}_T$, which are calculable from $\mathbf{c}$.

$$\mathbf{b}'_S = \mathbf{b}_S + \gamma_S \mathbf{I} \left(\mathbf{b}_S - \bar{\mathbf{b}}_S \right) \tag{6.4}$$

$$\mathbf{b}'_T = \mathbf{b}_T + \gamma_T \mathbf{I} \left(\mathbf{b}_T - \bar{\mathbf{b}}_T \right) \tag{6.5}$$

For any given face, it is a simple matter to assess the number of standard deviations by which each appearance parameter deviates from the prototype $\bar{\mathbf{c}}$. It would seem plausible that those global facial components which deviate drastically from the mean are those largely responsible for encapsulating the distinctive qualities of the individual face. Preferentially boosting these components might be expected to achieve a subtly different kind of caricature that enhances identity-related components (see figure 6.2). This is the key idea and suggests that, in the more general case, *the boost factors γ_S and γ_T (simply* *scalars in uniform caricature) become diagonal transforming matrices.* Thus, a more general from of caricature transform is proposed by rewriting equation 6.2 as,

$$\mathbf{c}'_S = \mathbf{c} + \Gamma\Delta\mathbf{c} \tag{6.6}$$

where ($\Delta\mathbf{c} = \mathbf{c} - \bar{\mathbf{c}}_S$). Γ is a diagonal matrix that weights each individual element of the difference vector, $\Delta\mathbf{c}$, according to a function, γ of parameter magnitude measured in units of standard deviation. Using the symbol t to represent the number of normalized standard deviations by which the model parameter deviates from the mean value, a suitable choice of function $\gamma\left\{t\right\}$ is required which will determine the boost factors along the diagonal of the transformation matrix Γ,

Transform Matrix:

$$\Gamma = \begin{bmatrix} \gamma(t_1) & 0 & \dots & 0 & 0 \\ 0 & \gamma(t_2) & & & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & & & \gamma(t_n) & 0 \\ 0 & 0 & \dots & 0 & \gamma(t_{n-1}) \end{bmatrix} \tag{6.7}$$

where the number of standard deviations by which the k_{th} appearance parameter deviates from the mean value is denoted by t_k. For possible functional mappings were chosen on an empirical bases. These are defined below and also illustrated in figure 6.3.

Uniform Function:

$$\gamma(t) = C \qquad -\infty < t < \infty \tag{6.8}$$

Step Function

$$\begin{array}{cc} \gamma(t) = C & |t| \geq t_{MIN} \\ \gamma(t) = 0 & |t| < t_{MIN} \\ & C > 1 \end{array} \tag{6.9}$$

Quadratic Function

$$\gamma(t) = at^2 + bt \qquad -\infty < t < \infty \tag{6.10}$$

Stretch-Shrink

$$\begin{aligned} &\gamma(t) = C && |t| \geq t_{MIN} \\ &\gamma(t) = B && |t| < t_{MIN} \\ & && C > 1, B < 1 \end{aligned} \tag{6.11}$$

Clearly, equations 6.8-6.11 give enhanced weighting to appearance components that deviate significantly from the norm and give rise to non-linear caricature effects. The relative enhancement that is provided can be controlled by the precise choice of constants a, b, B, and C. Nonlinear caricatures are amenable to a very simple geometric interpretation. Consider that the difference vector, $\Delta\mathbf{c}$, between the veridical appearance and the norm (which lies at the origin in our model): The caricature is created by applying the transformation matrix, Γ, to the $\Delta\mathbf{c}$ and adding this product to vector $\mathbf{c}$. The addition of $\Gamma\Delta\mathbf{c}$ for the generalized non-linear case has the effect of altering the length and direction of the veridical, whereas the uniform caricature effects only a lengthening of $\mathbf{c}$. In order to make a comparison between the proposed methods for caricature, the vector $\Gamma\Delta\mathbf{c}$ was scaled to have the same Mahalanobis distance measure in each case (i.e., uniform, step, quadratic, and stretch-shrink methods). The Mahalanobis distance is defined by $(\Gamma\Delta\mathbf{c} - \bar{\mathbf{c}})^T \mathbf{S}^{-1} (\Gamma\Delta\mathbf{c} - \bar{\mathbf{c}})$, where $\bar{\mathbf{c}}$ is the mean or prototypical face and $\mathbf{S}^{-1}$ is the inverse of the covariance matrix constructed from the $\mathbf{c}$ vectors corresponding to the sampled faces. The Mahalanobis distance measure was used because it takes in account the relative importance of each axis (characterized by its associated variance). We are free to make relatively large displacements from the veridical along the most significant axes and retain plausibility. Conversely, only small displacements along the least significant axes are allowed, preventing highly implausible caricatures from occuring.

6.5 Generation of non-linear caricatures

As a basic test of the methodology and also to visually explore the nature of both conventional (uniform) and non-linear caricatures, an appearance model was generated according to the procedure previously outlined in this thesis on a sample of 71 faces, using 134 landmark points per face (see figure 6.4). The sample contained a number of famous faces, 4 of which were caricatured according to the procedures defined in section 6.4 by equations 6.8-6.11. Appearance model caricatures using the uniform and non-linear methods are illustrated in figure 6.5 for Jackie Chan, Marylin Monroe, George W. Bush, and Mel Gibson,

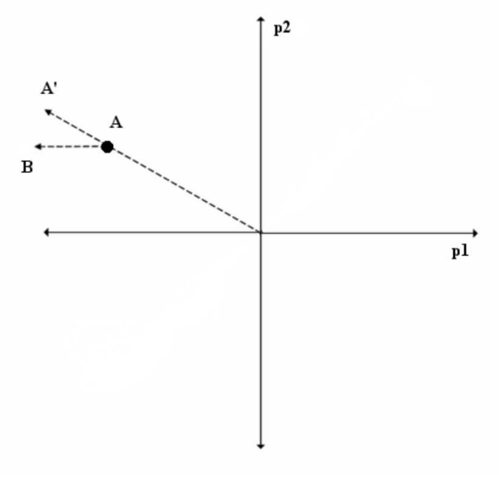


Figure 6.2: Schematic depicting the idea behind nonlinear caricature. The basic hypothesis is that when the extension along the axes in appearance space is large, these directions should be preferentially weighted. In the three-dimensional representation of face space above, the uniform caricature is defined by A'. The nonlinear caricature B results because each appearance parameter receives a different weighting as a function of the number of standard deviations from its mean.

respectively. The non-linear caricatures have been boosted by the same magnitude (Mahalanobis distance) as the uniform caricatures. Figure 6.5 is certainly suggestive of the fact that an effective caricature can be achieved via nonlinear mappings. The differences in the caricatures produced by the uniform, quadratic, and step functions are subtle but apparent upon careful inspection. Although some images depict a rather strong degree of caricaturing, the majority of transforms maintain a connection with the basic identity of the subject. The possibility that caricaturing using these methods may enhance identity is suggested; in particular, the *step function* weighting produces caricatures that are clearly different from the veridical, still appear to maintain *basic identity*, but do not introduce the rather comic effect characteristic of the uniform transform. Caricatures generated by the stretch-shrink function retain some aspects of the identity of the original face but clearly produce significantly different results from the other methods.

6.6 Preliminary experiment on nonlinear caricatures

Effective caricatures were achieved by skilled artists long before the mechanism of caricature was subjected to scientific study. Moreover, caricatures of the same subject by different artists often exhibit great variety, suggesting that

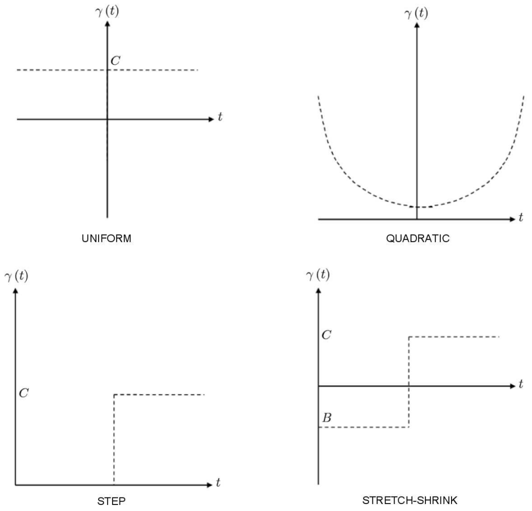


Figure 6.3: Empirically selected weighting functions for the generation of nonlinear caricatures.

there is some not inconsiderable flexibility in the caricature method. Given the largely nonlinear behavior of nature, such a nonlinear cognitive model is at least plausible. Whether some nonlinear mapping of the appearance parameter deviations from the prototype better models the cognitive process of recognition and can thereby produce a better-recognized/more-distinctive caricature than the uniform model must clearly be the subject of carefully conducted experiments. Such detailed experiments lie outside the immediate scope of this work.

A preliminary experiment in which these issues were explored was conducted. Motivated by the results shown in figure 6.5, our line of inquiry involved the notion that effective caricatures must generally satisfy two basic criteria. They must maintain identity (i.e., be recognizable as the subjects they are intended to depict), and they must, in the broadest sense, have a comic/humorous appearance. Thirty randomly selected participants were asked to view four different caricatures (the uniform, step, quadratic, and shrink-stretch mechanisms) of four famous persons (Jackie Chan, Marilyn Monroe, George W. Bush, and Mel Gibson). The images displayed to the participants were the four rightmost columns of figure 6.5. The true, veridical, images of the subjects in the leftmost column were not displayed. Prior to showing the caricatures, the participants were asked, "Do you know what the subject (e.g., Chan/Monroe/Bush/Gibson) looks like?" For each subject, they were then asked,

1. "Which of the four caricatures do you find most humorous/comic in ap-

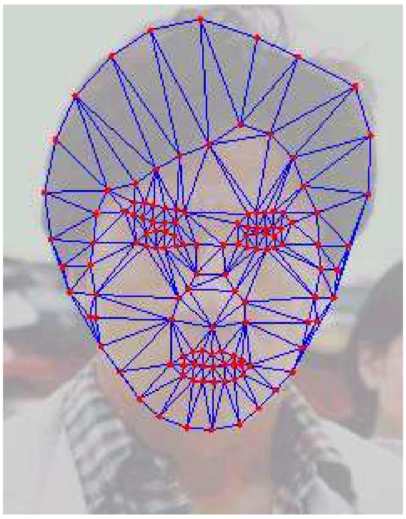


Figure 6.4: Landmark points used to describe face shape.

pearance?"

2. "Which of the four corresponds most closely to how you think the subject really looks?"

In those cases in which the participant did not know what the subject looked like, the second question was not asked. The results of this experiment are displayed in table 6.1 and summarized graphically in figure 6.6. These results suggests that the uniform caricature transform best captures the comic/humorous aspect, whereas the step transform produces a caricature that maintains the closest connection with the real appearance of the subject. The significance of these results was assessed using a χ^2 test. Two null hypotheses were examined.

1. *The uniform caricature mechanism is no more humorous than the others*: The calculated χ^2 value was 53.8 with 1 degree of freedom, yielding a probability of much less than $1/1,000$ that our experimental data would be obtained if the null hypothesis were true.

2. *The step caricature mechanism is no more effective at capturing realistic appearance than the others*: The calculated χ^2 value was 109.0 with 1 degree of freedom, yielding a probability of much less than $1/1,000$ that our experimental data would be obtained if the null hypothesis were true.

In both cases, the results provide a highly significant confidence level. Since the distance from the veridical is the same for each type of caricature, it would seem that use of the step transform, in which only the dominant spectral components are boosted, does indeed maintain a better connection with the real

Subject	Aspect	Uniform	Step	Quadratic	StretchShrink
Chan	Humorous/comic	3	1	9	16
	Similarity to real	2	21	0	1
Monroe	Humorous/comic	19	4	5	2
	Similarity to real	2	11	9	7
Bush	Humorous/comic	21	1	3	4
	Similarity to real	1	22	2	5
Gibson	Humorous/comic	21	4	3	2
	Similarity to real	1	16	10	1
Total	Humorous/comic	64	10	20	24
	Similarity to real	6	70	21	40

Table 6.1: Results of an experiment in which participants were asked to compare the caricatures presented in figure 6.5 on the basis of identity and comic effect .

appearance of the subject and may be more closely associated with identity. The subtly different question of which caricature is best recognized could be examined more precisely by conducting a careful experiment in which the best recognition may be determined by speed of response to the stimulus [62] or by some other approach. One significance of such an experiment lies in the following argument. If the nonlinear caricatures are best recognized, this will indicate that movement toward a region of minimum exemplar density is not the full explanation for why caricatures are effective. This follows from the fact that the distribution of parameters in appearance space is governed by a multivariate normal distribution. Thus, the nonlinear caricature (which adds a vector in a different direction from the direction defined by the veridical image and the prototype) is necessarily in a region of higher exemplar density (closer to the origin) than the uniform caricature is. Conversely, if uniform caricatures are best recognized, we obtain an interesting and remarkable result-namely, that the simplest of all the linear models is, in fact, the best one.

6.7 Summary

This chapter described how a veridical (original subject image) face is caricatured by making small adjustments to its corresponding appearance model parameters. The work was motivated by previous psychological literature, indicating that a caricature of a subject is more readily recognizable than the veridical image itself. The significance of these studies in relation to facial composites is clear - namely that caricatured composite images may represent an effective tool for use in criminal investigations. Previous methods for auto-

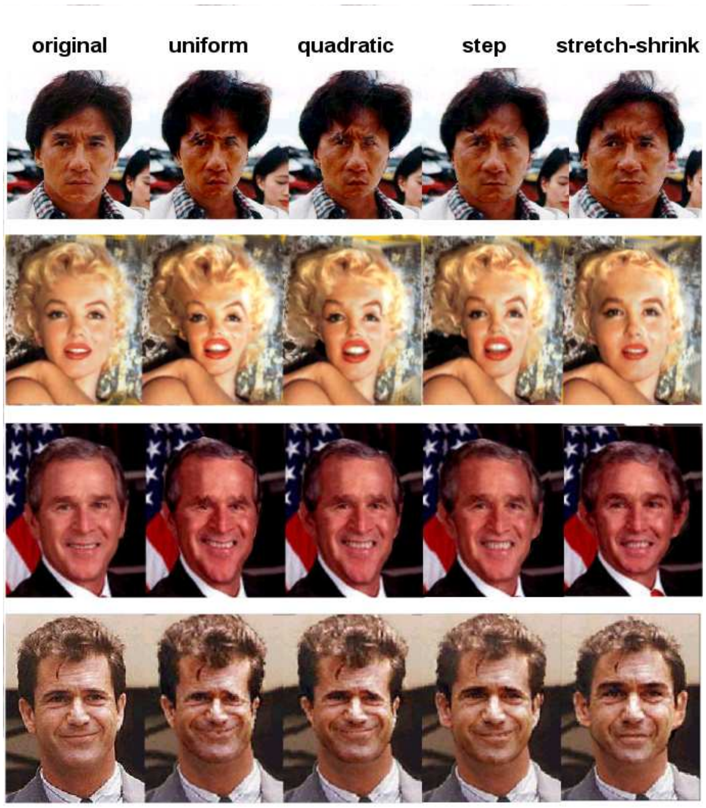
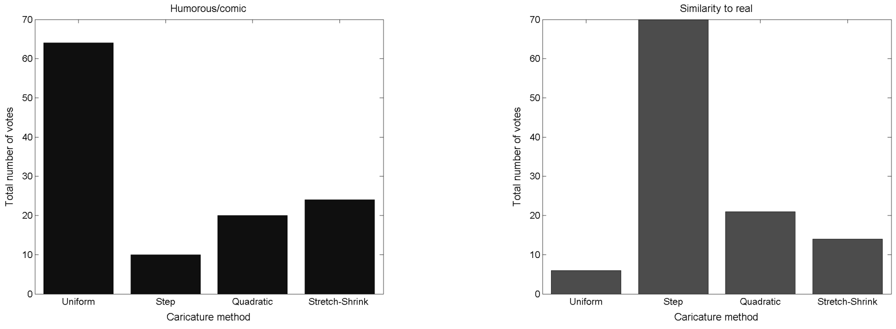


Figure 6.5: Examples of non-linear and uniform caricatures (Chan, Monroe, Bush, and Gibson). The non-linear caricatures depicted in this figure have been boosted to the same extent as the uniform caricatures (see the Generation of non-linear caricatures section). For the top row of images, the parameters for each of the four models are as follows: uniform caricature, $\lambda_S = 1$ and $\lambda_T = 0.3$; quadratic caricature, $b = 0$, $\lambda_S = a = 0.3$, and $\lambda_T = 0.3\lambda_S$; step caricature, $t_{MIN} = 1.7sd$, $\lambda_S = 1.5$, and $\lambda_T = 0.3\lambda_S$; stretch-shrink caricature, $t_{MIN} = 2.0sd$, $\lambda_S = 1$, $\lambda_T = 0.3\lambda_S$, and $B_S = -1$ and $B_T = 0.3B_S$.



(a) (b)

Figure 6.6: Performance of different methods for caricature summed over the four famous test faces. The established, uniform caricature method provided the most humorous image, whereas the step caricature method was judged to give a greater similarity to the subject

mated caricaturing have all followed the same basic mathematical formula (the uniform caricature method). In this chapter, alternative, non-linear methods were investigated. A preliminary experiment was conducted in which the standard uniform method was compared to other empirical methods for generating caricatures. The results of this experiment suggested that the uniform method may not be optimal with respect to identity.

Chapter 7

Summary and conclusion

It is clear from psychological studies that established feature based methods for generating facial composites are limited and incomplete. In this thesis the technical design and implementation of a composite system was presented that is capable of generating inherently global or whole face stimuli. Unlike current commercially available composite software, the system described here appeals to our recognition ability rather than the inferior process of recalling facial descriptions. Particular attention has been given to the integration of the semantic, global and feature based knowledge of facial appearance provided by the witness. A brief summary of the material presented follows.

7.1 Summary of thesis

The thesis began with an account of the established methods/systems for producing composite imagery. The functionality of these systems was described and a summary of the psychological literature validating their use was presented. With the exception of the artist's sketch, all these methods were/are feature based, forcing a facial likeness to be constructed by the selection of appropriate features from a card based catalogue or computer database. New methodologies for constructing facial composites were discussed with the emphasis on principal components analysis (PCA) and evolutionary approaches. The background literature outlining the use of PCA in facial modelling and the development of evolutionary algorithms was also provided.

Although a limited number of publications examining the cognitive advantages of whole-face evolutionary approaches exist, only Gibson et al [36] have previously published work giving a detailed account of the technical design and implementation of such a system. To aid the reader with an interest in facial composites but who is unfamiliar with PCA and evolutionary computation, the mathematical details of these techniques have been provided in the second chapter of the thesis. The chapter began with an historical introduction to the statistical technique of PCA followed by the procedure for deriving principal components from a data sample. An illustrative example of how PCA can be applied to shape and image data to yield a compact representation was provided. In the final section of Chapter 2, variations of the evolutionary algorithm theme were discussed.

The main aim of the thesis was to design a facial composite system that does not rely on a piece-wise ensemble of independent facial features, but instead appeals to our ability for recognising faces as a global (whole face) entities. Chapter 3 contained work relating to the core design and implementation of such a system which was refered to throughout as EigenFIT. The chapter began by making clear the motivation for the development of the EigenFIT facial composite system. Sections containing an overview of the system and its graphical user interface provide the reader with an understanding of how the system functioned in its most basic mode of operation. The construction of the appearance model (AM) from which novel face stimuli were synthesised was covered in section 3.4. In the section that followed, a purposely designed evolutionary algorithm (EA) was described which aided in the determination of the AM model parameters, that yielded a good likeness to the target face. Optimization of the EA using a virtual witness, enabled the model parameters, and hence the composite image, to be obtained quickly. An essential part of any composite system is the provision for applying a hairstyle. Since hair can not be effectively modelled in an appearance model, an alternative approach was required which was the subject of section 3.7. The final section in Chapter 3 explained a method for allowing the shape of one or more facial features to be fixed at a chosen stage in the previously described, global, evolutionary procedure.

The core implementation of the evolutionary whole-face approach (described in detail in Chapter 3) was carefully designed with ease-of-operation being one of the main objectives. Although the benefits of evolutionary composite methods have been documented, a purely evolutionary approach does not allow deterministic changes to be made to the composite image, which are a necessary component for any practicable system. Chapter 4 detailed methods for modifying the composite image which enhanced the standard global evolutionary approach. The aim of the work presented in Chapter 4 was to integrate semantic and feature-based knowledge of facial appearance with the global representation afforded by the appearance model. These complementary tools include,

- **Blend tool**: Augmented 'fit' characteristics from two or more faces in a generation into a single, averaged face image (see Chapter 4, section 4.3).
- **Facial attribute manipulation tool**: A general theoretical framework for attribute enhancement was presented and implemented for the specific case of age (see chapter 4, section 4.4).
- **Local feature manipulation tool**: Individual features were translated and scaled as instructed by the witness (see Chapter 4, section 4.5).
- **Application of fine details to a composite**: The functionality for applying fine details such as wrinkles to a composite face was developed. (see Chapter 4, section 4.6).

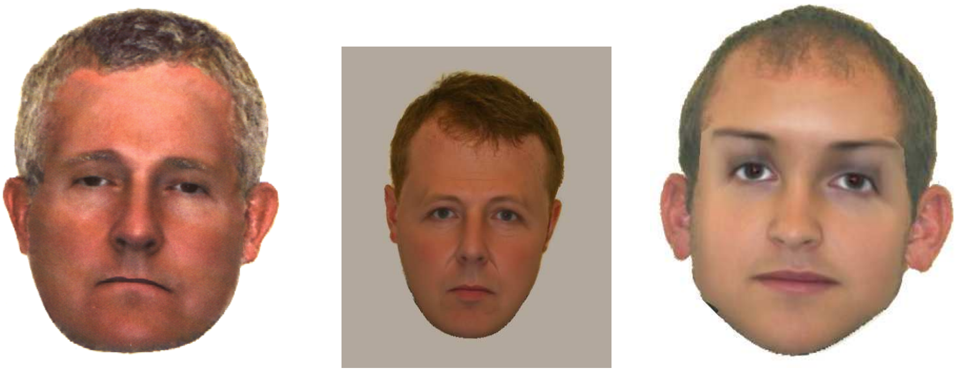
A limitation of an early proof-of-concept implementation of EigenFIT was the significant time lapse, experienced between generations. The delay was a due to the computational demands of the image warping procedure that is a necessary requirement for synthesising each of the nine faces in a new generation. A method for warping face images was suggested in Chapter 5 that may be more efficient than the commonly used reverse mapping procedure. The aim was to apply a limited number of modes of shape variation directly to the pixel coordinates, thereby avoiding the computational overheads normally associated with image warping. The chapter began with the motivation for an efficient warping method, followed by an overview of existing pixel mapping strategies. A novel forward mapping method was presented, based on the addition or superposition of pixel displacement fields. This approach was coded and compared to the standard reverse mapping procedure and a forward mapping technique referred to in the computer graphics literature as splatting.

Previous work has outlined a method for computer generated caricatures in which the difference between a subject's face and the mean (prototype) face was exaggerated by a scalar factor. However, there is no reason to suggest that this *linear scaling* is the only viable method for producing caricatures. In the final chapter of the thesis, the concept of non-linear facial caricatures was investigated using the AM framework. The chapter began with a qualitative overview of previous work in the area of automated caricature generators. The next section provided a discussion on the significance of the caricature effect on distinctiveness and identity, and the implications for facial composite synthesis. The mathematical construction for the previously established linear caricature were given, followed by the mathematical construction of novel, non-linear methods. A preliminary experiment was performed, which suggested that the standard linear approach to caricature may not be the best model for retaining identity.

7.2 Discussion

Technical viability

The technical viability of the EigenFIT facial composite system was established conclusively. Plausible, virtual faces could be synthesized using a pseudo-random sampling of appearance model parameter vectors. The parameter space relating to the AM offered sufficient facial variation, whilst being compact enough to allow a fast convergence to a target face. It was shown that fast convergence to a facial likeness was obtainable using a purposely designed evolutionary algorithm. Although the AM described was intrinsically global, deterministic changes to the shape of specific individual facial features were also shown to be possible. Common semantic descriptions such as "his face looked older" were relatively easy to implement and control in the appearance model framework, which is not the case in the commercial feature based systems.



(a) Criminal investigation by Dover Police Station. Image generated according to witness' instructions, 28th September 2005.

(b) TV presenter Bill Turnbull. Image generated for BBC Breakfast TV, broadcast on 10th February 2006.

(c) Unnamed composite produced by pupils from Hilden Grange School during a "Scene of Crime Day" at the University of Kent on 16th March 2006.

Figure 7.1: Facial composite images generated using EigenFIT.

Psychological validation

An independent psychological evaluation of an early version of the EigenFIT system was performed by Pike and Brace (unpublished). Their report included a national survey of 75 composite operators who were adept in the use of either

E-FIT or PROfit, the two leading commercial systems in the U.K.. Participants were asked to reflect upon their experiences regarding both the verbal description that witnesses provide just before composite construction commences, and the type of requests for changes to the composite that witnesses make during composite construction. A final section asked for views on new systems such as EigenFIT in which the witness is required to pick out face(s) from an array of images. A summary and discussion of Pike and Brace's survey is provided here.

The first section of the survey addressed the verbal interview that precedes established composite methods. Around 57% of operators reported that the witness found it *fairly difficult* to convey a verbal description of the perpetrator and 21% said that it was a *very or extremely difficult* task. Verbal descriptions were often 'holistic/global', pertaining to the face as a whole rather than detailed descriptions of individual features. 80% of operators reported that witnesses included information about age, approximately 50% said they included information about character and distinctiveness and just over 40% terms relating to gender and ethnicity. Only 25% reported that witnesses mentioned information regarding facial expression. Operators also reported that the witness' descriptions often decribed skin types such as freckled, smooth or wrinkled etc. The second section of the report asked participants/operators to comment on the sorts of instructions provided by the witness *during* the construction of the composite. These instructions are summarised in table 7.1.

Facial characteristics	**Extremely/very useful**
Gender	44%
Age	92%
Ethnicity	55%
Character	55%
Expression	76%
Distinctiveness	56%
Attractiveness	41%
Skin type	72%
Skin texture	80%
Skin blemishes	80%
Male characteristics	46%
Female characteristics	55%

Table 7.1: Usefulness of witness' comments during the composite construction procedure.

The final section of the survey asked operators to provide their views on the value of a composite system that presented the witness with arrays of face stimuli. Nearly 50% stated that the witness would *probably* or *definitely* be able to make comparative judgements such as ”the eyes were bigger than the first face in the array”. When asked if a witness would be able to interact directly with a composte system like EigenFIT, 40% felt that this would not be possible and 17% felt that the witness would. 32% were undecided whether such a system could be operated by the witness alone. When asked if the witness would require guidance form a trained operator when using an array based system 56% replied *definitely* and 21% replied *probably.* Operators stated that their presence would be required during the construction process due to the witness' need for emotional support or the witness' lack of familiarity with using computers.

Experiments to determine the psychological viability of array based systems were also performed by Pike et al. Unlike the survey, that focussed on the experiences of trained *composite operators*, the experiments were conducted with the witness in mind. Details of the experimental procedure and the results can be found in the report. The findings of these experiments are summarized here. The issues investigated were,

- whether witnesses are able to interact with arrays of faces.
- are they able to pick out the best match for the suspect?
- do they ever want to use any alternative selection methods?
- do they ever want to interact with the system by other means, such as by manipulating individual features?
- does the presentation of arrays overshadow memory for the suspect?

The results demonstrated that a witness *would* be able to interact with an array based system, although in many instances additional verbal descriptions were provided by the witness. The witnesses expressed a wish to interact with the system deterministically, in parallel with the simple evolutionary interface. Selecting multiple faces in a generations and elimination of the worst, was found to be desirable. No evidence was establish which suggested that viewing 30-60 arrays of faces had a detrimental effect on memory for the target face. Conclusions drawn from these experiments were pivotal in subsequent refinements of the EigenFIT system, that complemented the purely global evolutionary approach.

Home Office trial

To date, EigenFIT has been used in two criminal investigations. In one of these cases, the witness judged the composite image produced using EigenFIT to be a better likeness to the target face than an image she had previously generated using a commercially available composite system. Based upon the acknowledged potential of the system, the Home Office has provided funding for a U.K. trial in Derbyshire and Leicestershire police forces which is due to commence in July 2006. The aim of the trial is to test the system outside the laboratory environment, gaining feedback on usability and accuracy. Currently, the use of composite systems is restricted to serious crimes. However, of particular interest in the trial is the possible use of this system for volume crime. This may feasible because EigenFIT is easier to use and can generate a composite image more quickly than established methods.

7.3 Future development

The basic technical and psychological viability of the composite method described in this thesis has been proven. Suggested areas for future work to enhance the composite procedure described are:

- An improved method for the application of hairstyles, in which a seamless join can be achieved between the hairstyle and composite face.
- Independent manipulation of ear shape (in the current version of EigenFIT the ears are treated as part of the hairstyle).
- A refinement of the point model, allowing the shape of chin to be modelled more accurately.
- The incorporation of more facial attributes, including masculinity.
- An automated or advanced semi-automated method for landmarking training examples (currently a very time consuming task).
- The development of a 3D version of the composite system that may aid the recognition process thereby producing more composite images.

From a theoretical perspective, the method for encoding faces, achieved in this work by PCA, deserves further investigation. The PCA method for encoding was chosen because it enables a face to be represented compactly by a vector of parameters, that is of much smaller dimension than the original data. This representation of the face is optimal in the least squares sense, in terms of the pixel values and shape coordinates comprising the sample data. Although previous studies have shown that PCA models of the face are capable of explaining some aspects of human face recognition, this does not necessarily mean that the current PCA representation is optimal in a perceptual sense. A perceptually optimal representation of the face would require an experiment in which the response of observers to face stimuli was recorded. The results of this experiment could be used to perform a transformation on the original pixel and coordinate data, which would enable a compact representation that more closely mimics the way in which faces are encoded in the human mind. The exact nature of a superior encoding method remains unknown, and may involve PCA or other techniques such as independent component analysis (ICA) or wavelet analysis.

Appendix A

Camera settings

Canon EOS D60	
Max resolution	3072 × 2048
Image ratio w:h	3:2
Effective pixels	6.3 million
Sensor photo detectors	6.3 million
Sensor size	22.7 × 15.1 mm
Sensor type	CMOS
Colour filter array	RGB
ISO setting	400
Auto Focus	enabled
White balance override	fixed, daylight
Canon Lens EF 135mm	1:2.8 SOFTFOCUS (set to 0)
Shutter Speed	$\frac{1}{8}^{th}$ sec
Lens aperture	f/16

Table A.1: Camera settings table, reproduced by kind permission of Matthew Maylin.

Appendix B

Landmarking instructions

A description of the steps required for landmarking a face and the functionality of the landmarking tool are provided as follows:

1. User navigates to a directory of choice via the *load file* push button, then selects an image file from a listbox. The chosen image appears in the work area located on the right hand side of the interface.

2. Initially, the user is required to locate three landmarks; at the outer corner of the left eye[1], at the outer corner of the right eye and the third at the base of the nose. After the third point has been located the remaining 105 landmarks are automatically placed in their approximate positions. This is achieved by computing the transformation required to map the three landmarks, contained in the previously determined mean face shape, to the initial three landmarks located by the user. The computed transformation is then applied to mean face shape as a whole, providing an affine transform that defines the preliminary positions of all the points in the face shape.

3. The landmark points control a set of *spline curves* (also plotted) that delineate the perimeter of the internal facial features and the head itself.

4. Landmarks can subsequently be moved from their approximate positions to their correct locations using a *click and drag* technique, whereby the user selects a landmark point via the left mouse button and drags it to its correct position, holding the left button down during the procedure. When the left mouse button is released, the spline curve(s) associated

[1] The convention used here is that left refers to the left hand side of the displayed face from the perspective of the user, i.e. not the subject's left eye

with the translated point are redrawn, updating the face shape in response to the users action (see figure 3.6). Selected or 'active' landmarks and their associated spline curves are plotted as red graphics objects, whereas the inactive landmarks and spline curves are plotted in blue. When a spline curve becomes active its description, e.g. 'chin', appears in a frame labelled *instructions* on the left hand side of the interface. This is particularly helpful when adjusting landmarks around the mouth, where there are many spline curves that could otherwise become confused. Landmarks that define the end of one spline curve and the beginning of another connected curve are referred to as *base landmarks* and are plotted as magenta circles, distinguishing them from the ordinary landmarks plotted in either red or blue.

5. For landmarking purposes the images are displayed at full resolution (2048×3072 pixels, 300dpi). Regions of the face can be enlarged using the *zoom mode* push button located under the image. When the zoom mode is set to on, placing the mouse cursor over a region of interest in the face image and clicking the left mouse button will enlarge that area, making it easier to place landmarks/spline curves accurately. With the zoom mode set to on, horizontal and vertical sliders appear at the left side and bottom edge of the image respectively. These may be used to pan around the image when in zoom mode. Selecting the zoom mode button again allows the user to exit zoom mode and return to the standard landmarking mode.

6. Sets of landmarks corresponding to whole features can be translated using by selecting the *lock* pushbutton, located near the bottom left hand corner of the interface. When this button is selected the click and drag operation results in a translation of the nearest feature to the cursor position at the time when the left mouse button is depressed. This option can save time in situations where the approximate landmark positions, generated by the affine transformation, are a poor fit to the actual features within the image itself.

7. Once the user is satisfied that all the spline curves are correctly located, the face shape can be saved using the *save file* pushbutton on the left hand side of the interface. Depressing the save file button starts an interpolation process whereby *pseudo-landmarks* are generated that lie at equidistant positions along a spline curve. Spacing of the pseudo-landmarks for each curve section is predetermined on an empirical basis according to the

likely curvature of the section. For instance, the perimeter of the mouth exhibits a higher degree of curvature than the boundary of the head, hence the densities of pseudo-landmarks in curve sections delineating the mouth are relatively high. Pseudo-landmarks are saved to a MATLAB '.mat' file as are the original landmarks which are required if the saved shape is to be reloaded for modification in the future.

Appendix C

Operating instructions

C.1 EigenFIT - EasyFIT operation overview

1. On start-up, a form is displayed into which details that identify the composite are entered. Fields are provided for the witness' forenames, surname, date of birth and also the operator's rank and number. This information is combined with the current date to generate a unique reference number that can be used to identify the composite in the future.

2. In keeping with the graphical approach, prototypical faces are used to initialise the EigenFIT system. This involves the user selecting the appropriate sex and race for the target face. By selecting prototypical faces the user is effectively seeding the search algorithm. Hence, a suitable choice of prototypical faces leads to a set of appearance model parameters that relate to an approximate 'face type'. This provides a more informed starting point for the search procedure than the global mean, which lies at the origin of appearance space.

3. Prior to the evolutionary process, a hairstyle is chosen using the *hair tool*. From a perceptual point of view it is sensible to ask the witness to select an appropriate hairstyle first as the external features are more salient in unfamiliar faces and there is evidence to suggest that facial features should be selected in order of decreasing significance . The user can scroll through the available hairstyles using a slider, with each increment in slider position displaying nine more hairstyles in the familiar three × three configuration. Hairstyles are mapped onto the mean face so that the witness may view the hair in its usual with-face context. Since the mean face is unlikely to be very similar to the target, there may be issues associated with this approach concerning visual overshadowing. The effect, if

it exists, is unlikely to be significant and has not been investigated in this case. A filter is provided to sort the hairstyles into categories describing the length and colour. Thus the number of candidate hairstyles that the witness must examine can be greatly reduced by marking the appropriate checkboxes provided in the filter.

4. At this stage in the process the setup has been completed and a face may now be evolved using a simple *one touch* interface. Nine randomly sampled virtual faces are drawn from a normal distribution that is determined by the initial choice of prototypes. These faces, collectively known as the initial population, are presented against a grey backdrop on the left hand side of the user interface. The witness is required to choose one of the nine faces on the basis of its similarity to the target face. Virtual faces that exhibit a particularly poor likeness to the target may be removed from view using the *remove face* icon positioned near the lower left hand corner of the face image. This de-clutters the backdrop, making it easier for the witness to form an opinion on the suitability of the remaining visible faces. Once the witness has made a choice, the initial population is replaced by nine more faces comprising the first generation. The process of selecting a face and synthesising new faces is repeated until a good likeness is obtained. An approximate likeness is often obtained in as few as twenty generations or less.

5. If the witness is especially unhappy with the initial population or subsequent generations, more randomly generated faces can be synthesised using the *Generate More* pushbutton located on the right hand side of the interface. This feature of the interface should be used sparingly and not as a means for searching the appearance space. It produces images on a totally random basis, and given the extent of the appearance space, would offer an inefficient 'brute force' method for finding a likeness to the target. If used wisely, it can be a valuable tool for preventing the search algorithm getting stuck in a local minimum in the later stages of the evolutionary process.

6. An *Undo* pushbutton is also provided that can be used if the witness feels that a poor choice of face has been made, or a face has been selected unintentionally and another image is preferred. Although both the Undo and Generate More options allow a user to avoid local minima, Undo does not introduce new genetic information into the current generation, it is merely a means of retracing ones steps.

7. In EasyFIT mode the user has the option to *lock* the shape of particular facial feature that exhibits a good likeness to the corresponding feature in the target face. This achieved by choosing a region of the *iconic face* located on the right hand side of the interface and then selecting one of the nine virtual faces. Placing the cursor over a feature in the iconic face and using a single click of the left mouse button turns that region from grey to blue, indicating that the shape of the chosen feature will be fixed through subsequent generations. Deselecting a region of the iconic face reintroduces shape variation in the previously locked feature. If required, more than one feature may be fixed at any given instant. The available choice of features that can be locked are the eyebrows (left/right pair), eyes (left/right pair), nose, mouth and face shape or any combination of these.

8. Once an acceptable likeness has been obtained using the tools provided, the current stallion image (best likeness) can be saved both as graphics file and as an EigenFIT composite file. The graphics file is intended to be reproduced for use in a criminal investigation. The EigenFIT composite file may be reloaded for modification by the witness in the future. The icon for saving the composite is labelled *finish* and is located at the top right hand corner of the interface. This icon also allows the composite image to be exported to a graphics package for a post processing. This enables the operator to draw on tattoos or embellish the composite image in other ways under instruction by the witness.

Appendix D

Examples generated for television appearances

D.1 Aging example

An aging example using attribute manipulation and a wrinkle-map, provided for the Beyond International Ltd production company (used in a programme shown on Australian television).

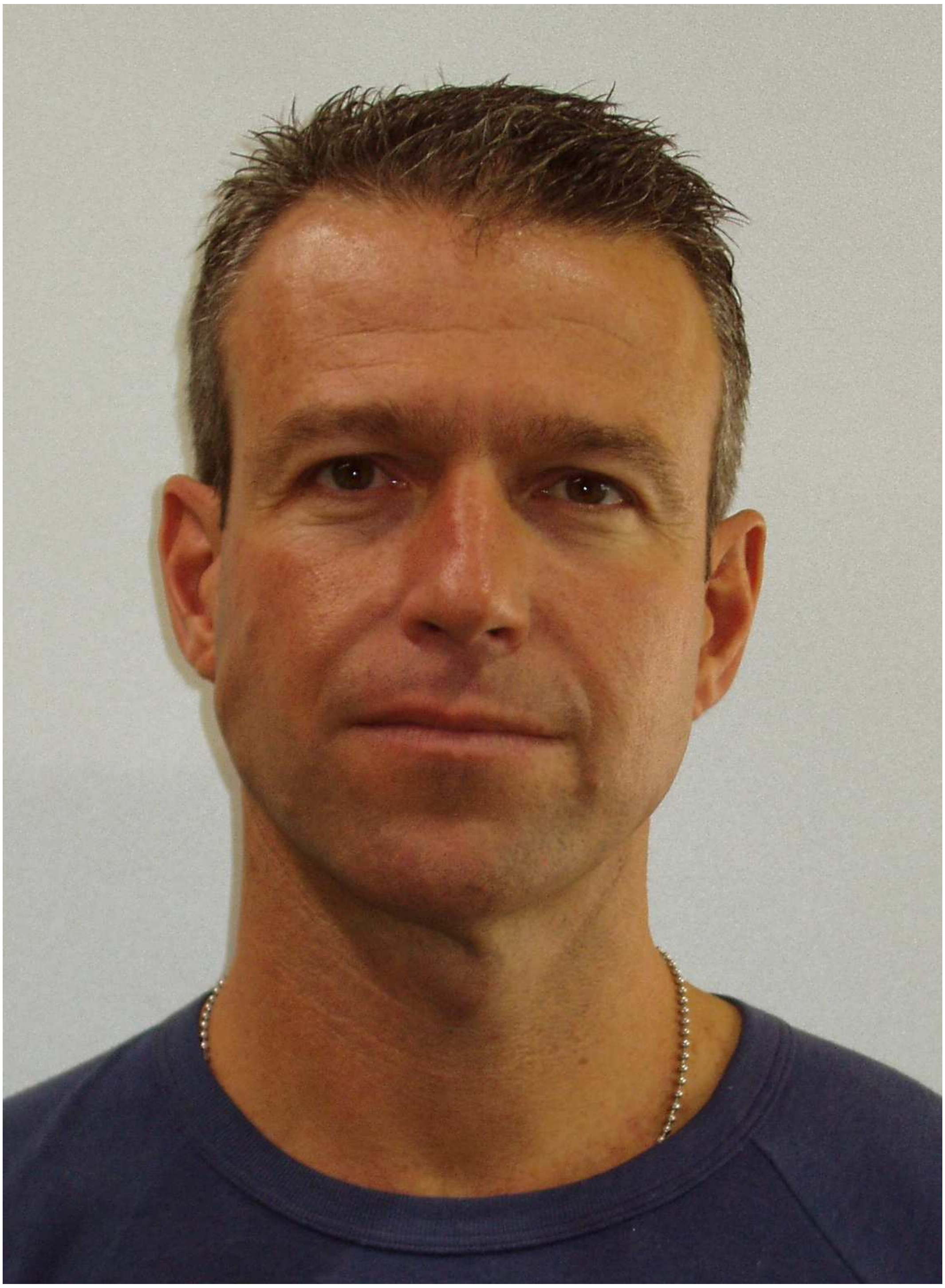


Figure D.1: Hayden Turner, "Beyond Tomorrow" presenter (Beyond International Ltd)

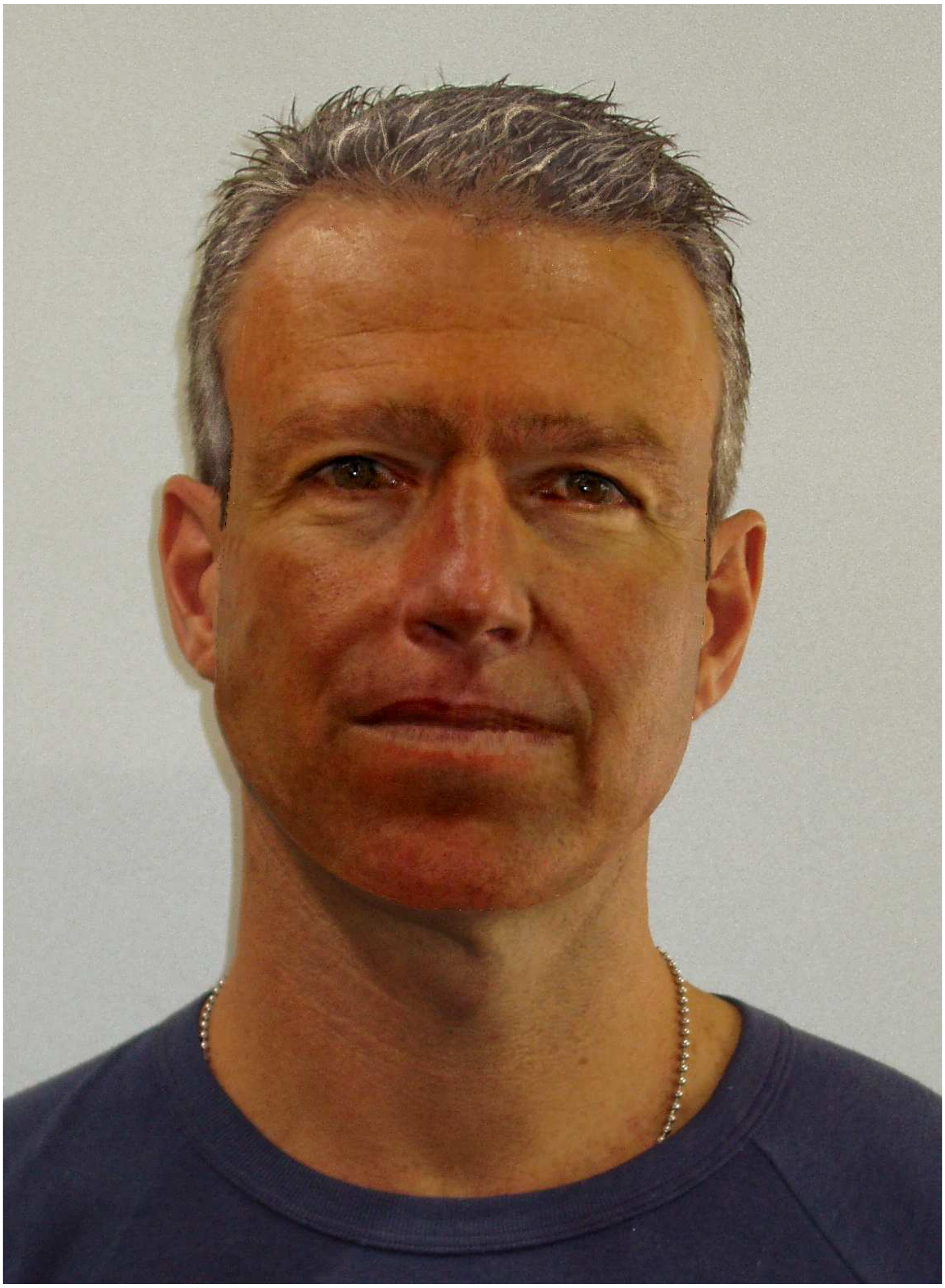


Figure D.2: Hayden Turner, "Beyond Tomorrow" presenter (Beyond International Ltd) aged

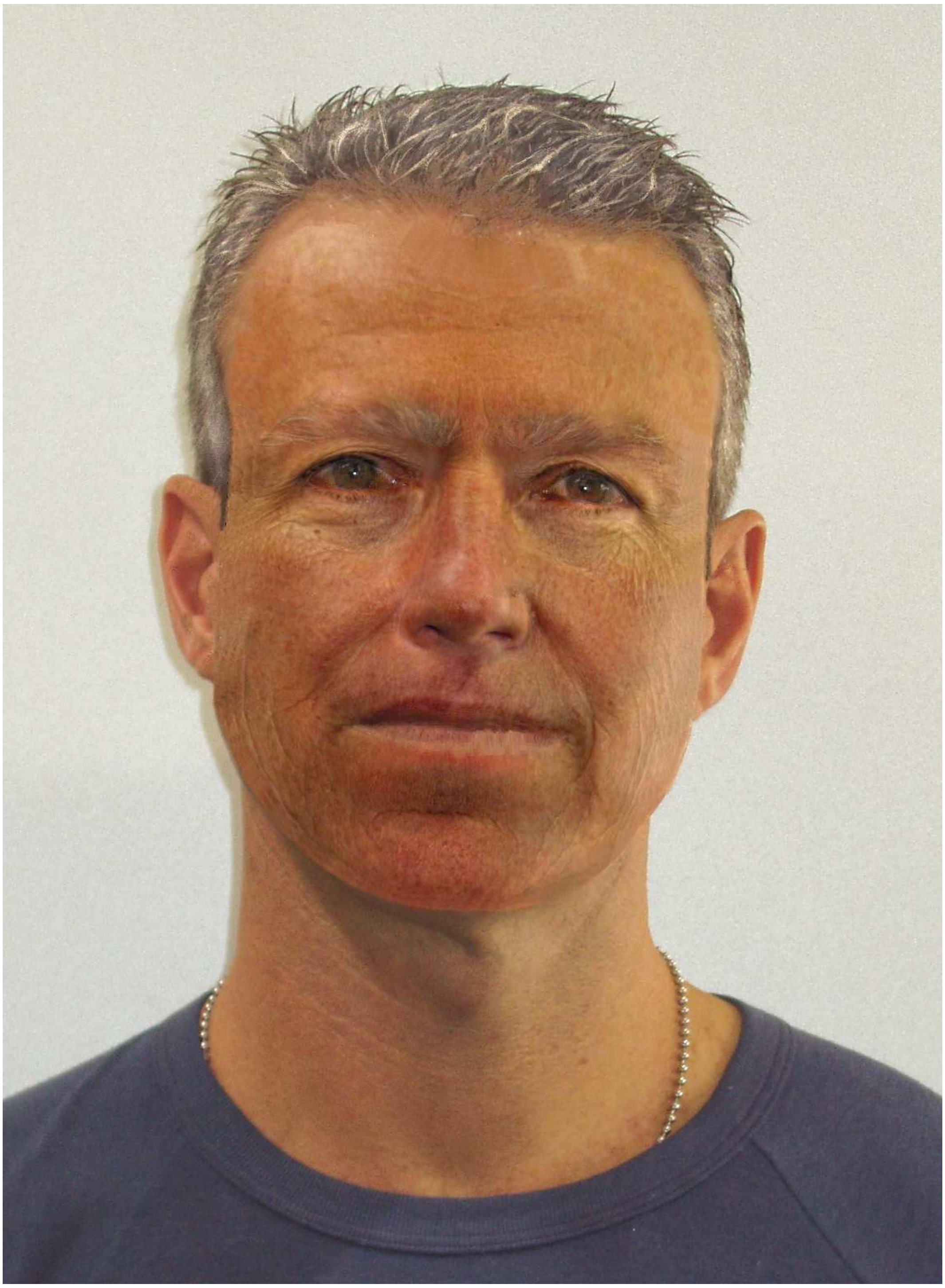
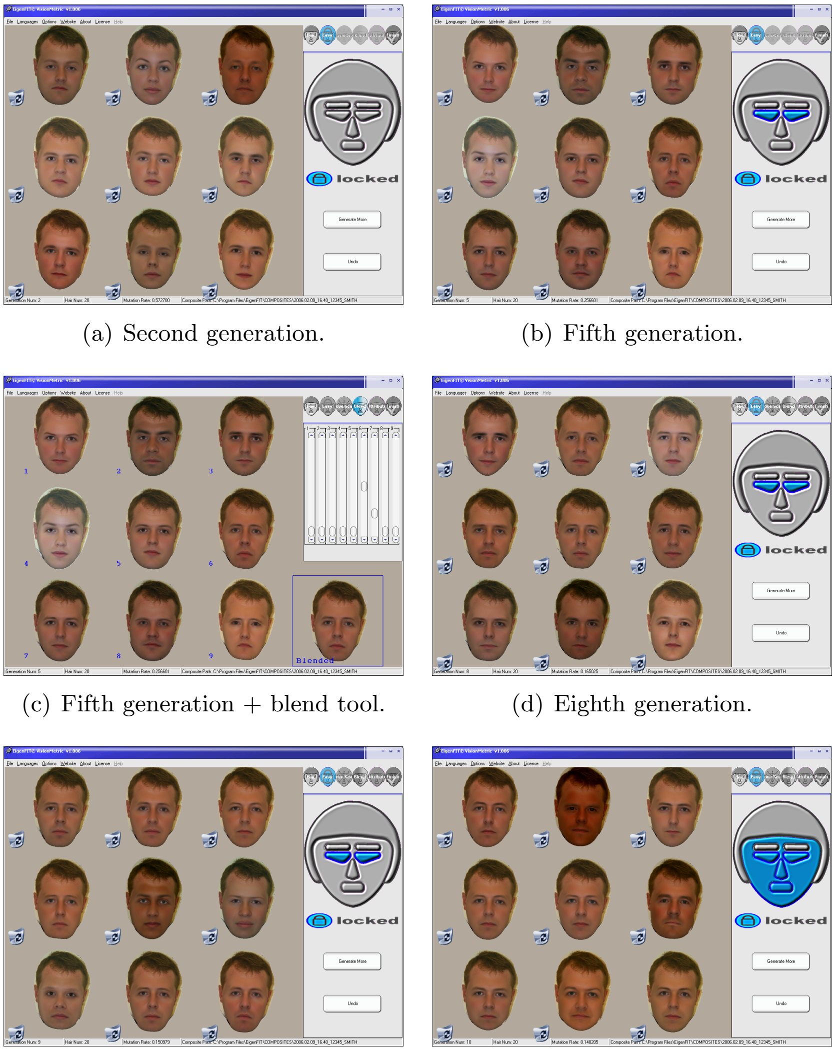


Figure D.3: Hayden Turner aged and wrinkle-map overlaid. Image produced for Beyond International Ltd's "Beyond tomorrow" programme

D.2 EigenFIT composite example

An example of a composite sequence leading to a likeness of BBC TV presenter Bill Turnbull (used on national breakfast television). For brevity some of the steps in the sequence have been removed.



(a) Second generation.

(b) Fifth generation.

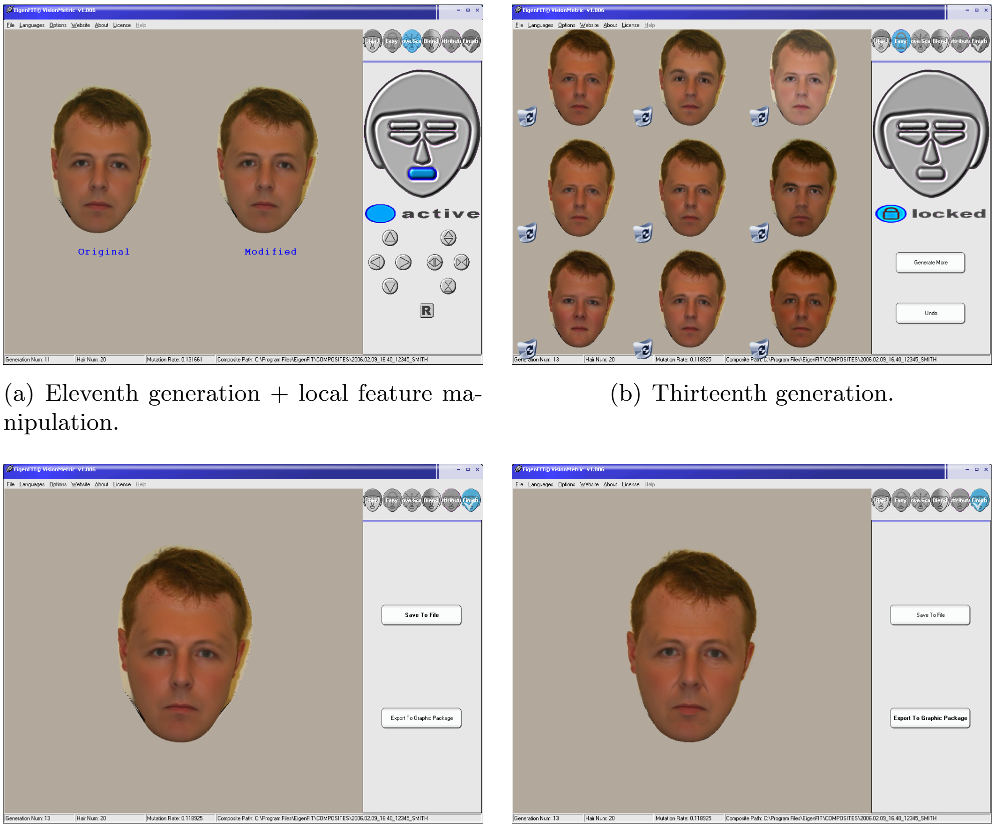
(c) Fifth generation + blend tool.

(d) Eighth generation.

(e) Ninth generation.

(f) Tenth generation.

Figure D.4: Example composite sequence (part I).


EigenFIT© VisionMetric v1.006
File Languages Options Website About License Help
Original
Modified
active
R
locked
Generate More
Undo
Save To File
Export To Graphic Package

Generation Num: 11 | Hair Num: 20 | Mutation Rate: 0.131661 | Composite Path: C:\Program Files\EigenFIT\COMPOSITES\2006.02.09_16.40_12345_SMITH

(a) Eleventh generation + local feature manipulation.

Generation Num: 13 | Hair Num: 20 | Mutation Rate: 0.118925 | Composite Path: C:\Program Files\EigenFIT\COMPOSITES\2006.02.09_16.40_12345_SMITH

(b) Thirteenth generation.

Generation Num: 13 | Hair Num: 20 | Mutation Rate: 0.118925 | Composite Path: C:\Program Files\EigenFIT\COMPOSITES\2006.02.09_16.40_12345_SMITH

(c) Thirteenth generation + aging attribute manipulation.

Generation Num: 13 | Hair Num: 20 | Mutation Rate: 0.118925 | Composite Path: C:\Program Files\EigenFIT\COMPOSITES\2006.02.09_16.40_12345_SMITH

(d) Final composite image including graphics package enhancement.

Figure D.5: Example composite sequence (part II).

Appendix E

Two pass splatting algorithm implementation

A texture mapping (forward pixel mapping) method for achieving image warps was previously proposed by Heckbert [45]. A naive two pass splatting algorithm based on Heckbert's original approach is described here.

E.1 Overview of basic approach

To prevent holes occuring in the destination image a one to one mapping from source to destination was avoided. Instead line segments were inserted into the destination image using the two pass splatting method. This approach shares similarities to the splatting techniques previously described [45, 40, 102] for projecting textures belonging to 3D objects into the plane of a monitor. Defining u, v as the interior coordinates of the source triangle and x, y as the forward mapped interior coordinates in the destination triangle, a method for assigning pixel intensity values in the destination image is outlined in algorithm figure 4. In the first pass of this two pass algorithm, vertical line segments were 'splatted' into the destination image. At each iteration of the loop, the pixel value at position u, v in the source image are sampled and mapped to the integer coordinates in the destination image that lie in the range $round(x), y - \frac{s_y}{2} < y_{int} \leq y + \frac{s_y}{2}$. Hence, the columns for which $round(x)$ is defined were filled and an *intermediate* destination image ($destination_{ypass}$) is formed as illustrated in figure E.1. In the second pass, a horizontal line segment is placed at the location of each pixel that was defined in the first pass. This horizontal splatting fills the remaining holes in the destination image ($destination_{xpass}$) as shown in figure E.2

Algorithm 4 Two pass pixel mapping algorithm

```
for loop over $v$ do
  determine integer pixel coordinates $\{y_{int}\}$ in the destination image that lie in the range $y - \frac{s_y}{2} < y_{int} \leq y + \frac{s_y}{2}$
  for loop over integer coordinates do
    $destination_{ypass}\left(round\left(x\right), y_{int}\right) = source\left(u, v\right)$
  end for
end for
for loop over $y_{int}$ do
  determine integer pixel coordinates $\{x_{int}\}$ in the destination image that lie in the range $x - \frac{s_x}{2} < x_{int} \leq x + \frac{s_x}{2}$
  for loop over integer coordinates do
    $destination_{xpass}\left(x_{int}, y_{int}\right) = destination_{ypass}\left(round\left(x\right), y_{int}\right)$
  end for
end for
```

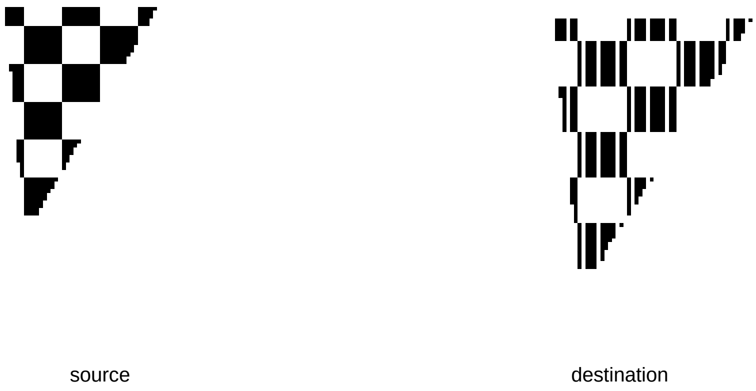


Figure E.1: Scaling transformation with vertical splatting. Pixel intensity values are spread in the vertical direction, thereby filling holes in the destination image (compare with figure 5.3(b) in which no splatting was used)

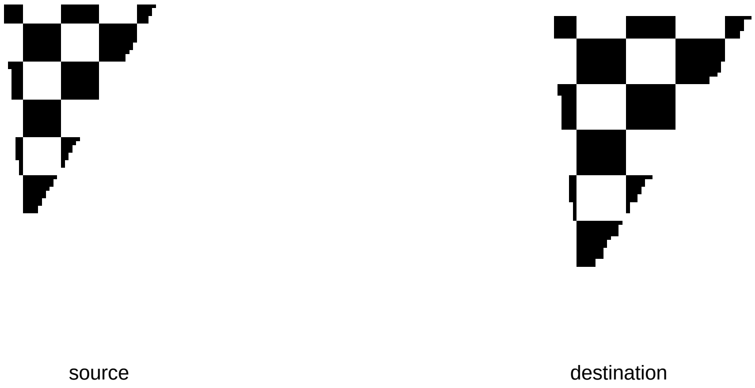


Figure E.2: Transformation with vertical and horizontal splatting. Pixel intensity values were initially spread in the vertical direction and then in the horizontal direction, thereby filling all the holes in the destination image (compare with figure 5.3(b) in which no splatting was used and figure E.1 in which only vertical splatting was used).

E.1.1 Determining the length of each line segment (splat)

To prevent holes occuring in the destination image the length of the vertical, s_y, and horizontal, s_x, line segments must be determined. Here the transformation is assumed to be piece-wise affine. For each triangular (piece) the corresponding line segment lengths are constant and determined from the transformation matrix as follows,

$$M(x,y) = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ 0 & 0 & 1 \end{bmatrix} \tag{E.1}$$

Defining the horizontal and vertical scaling parameters as s_x and s_y as $s_x = \frac{\partial x}{\partial u}$ and $s_y = \frac{\partial y}{\partial v}$ respectively, leads to the following equations.

$$s_x = \frac{\partial x}{\partial u} = \frac{\partial(m_{11}u + m_{12}v + m_{13})}{\partial u} = m_{11} \tag{E.2}$$

$$s_y = \frac{\partial y}{\partial v} = \frac{\partial(m_{21}u + m_{22}y + m_{23})}{\partial v} = m_{22} \tag{E.3}$$

Bibliography

[1] ABM. (profit composite software). http://www.abm-uk.com/.

[2] James Arvo, editor. *Graphics gems II.* Academic press inc, 1991.

[3] Aspley. (e-fit composite software). http://www.efit.co.uk/.

[4] P.J. Benson and D.I. Perrett. Perception and recognition of photographic quality facial caricatures: implications for the recognition of natural images. *Eur J Cogn Psychol*, 3:105–135, 1991.

[5] P.J. Benson and D.I. Perrett. Extracting prototypical facial images from exemplars. *Perception*, 22:257–262, 1993.

[6] F.L. Bookstein. Principal warps: Thin plate splines and the decomposition of deformations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11:567–585, 1989.

[7] F.L. Bookstein. Landmark methods for forms without landmarks: localizing group differences in outline shape. *Medical Image Analysis*, 1(3):225–244, 1997.

[8] G.E.P. Box. Evolutionary operation: a method of increasing industrial productivity. *Applied Statistics*, 6:81–101, 1957.

[9] N. Brace, G. Pike, and R. Kemp. Investigating e-fit using famous faces. In A. Czerederecka, T. Jaskiewicz-Obydzinska, and J. Wjcikiewicz, editors, *Traditional Questions and New Ideas.* Krakw: Institute of Forensic Research Publishers., 2000.

[10] H.J. Brenermann. *Optimization through evolution and recombination.* Spartan Books, 1962.

[11] S.E. Brennen. Caricature generator: The dynamic exaggeration of faces by computer. *Leonardo*, pages 170–178, 1985.

[12] V. Bruce, P. Hancock, and A.M. Burton. *Human Face Perception and Identification, in Face Recognition - From Theory to Applications*, pages 51–72. Springer Verlag, 1998. ISBN3-540-64410-5.

[13] R. Brunelli and O. Mich. SpotIt! an interactive identikit system. *Graphical models and image processing: GMIP*, 58(5):399–404, 1996.

[14] D.M. Burt and D.I. Perrett. Perception of age in adult caucasian male faces: computer graphic manipulation of shape and colour information. *Royal Society Proceedings B*, 259:137–143, 1995.

[15] P. Burt and E.H. Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, 2(4):217–236, 1983.

[16] C. Caldwell and V. S. Johnston. Tracking a criminal suspect through face-space with a genetic algorithm. In *Proc. Fourth Int. Conf. On Genetic Algorithms*, pages 416–421. Morgan Kaufmann, 1991.

[17] Kenneth R Castleman. *Digital Image Processing.* Prentice Hall, 1979.

[18] D. Christie and H. Ellis. Photofit constructions versus verbal descriptions of faces. *Journal of Applied Psychology*, 66(3):358–363, 1981.

[19] T. Cootes, D. Cooper, C.Taylor C, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[20] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. In H.Burkhardt and B. Neumann, editors, *Proc. European Conference on Computer Vision*, volume 2, pages 484–498. Springer, 1998.

[21] T.F. Cootes and C.J. Taylor. Statistical models of appearance for computer vision. Technical report, Imaging Science and Biomedical Engineering, University of Manchester, Manchester M13 9PT, U.K., September 2001.

[22] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Training models of shape from sets of examples. In *British Machine Vision Conference*, page 918, 1992.

[23] I. Craw and P. Cameron. Parameterising images for recognition and reconstruction. In Peter Mowforth, editor, *British Machine Vision Conference*, pages 367–370, London, 1991. Springer Verlag.

[24] B.L. Cutler, C.J. Stockleim, and S.D. Penrod. An empirical examination of computerized facial composite production system. *Forensic reports*, 1:207–218, 1988.

[25] Charles Darwin. *The Origin of Species by Means of Natural Selection: The Preservation of Favored Races in the Struggle for Life.* London: Penguin Books, 1985 (Originally published in 1859).

[26] L. Davis. Adapting operator probabilities in genetic algorithms. In *International Conference on Genetic Algorithms ICGA89*, pages 61–69, 1989.

[27] S. DiPaolo. Investigating face space. SIGGRAPH presented paper (sketch), 2002.

[28] I.L. Dryden and K.V. Mardia. *Statistical Shape Analysis.* John Wiley, 1998.

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

[30] D.H. Enlow and M.G. Hans. *Essentials of Facial Growth.* W.B. Saunders Company, 1996.

[31] L.J. Fogel, A.J. Owen, and M.J. Walsh. *Artificial intelligence through simulated evolution.* McGraw-Hill, New York, 1966.

[32] B.R. Frieden. *Probability, Statistical Optics, and Data Testing.* Springer, September 2001.

[33] G.J. Friedman. Digital simulation of an evolutionary process. *General Systems Yearbook*, 4:171–184, 1959.

[34] C. Frowd, V. Bruce, D. Ross, A.McIntyre, and P. Hancock. An application of caricature: how to improve the recognition of facial composites. *Visual Cognition*, In Press.

[35] C.D. Frowd. *EvoFIT: A Holistic, Evolutionary Facial Imaging System.* PhD thesis, Department of Psychology, University of Stirling, 2001.

[36] S.J. Gibson, C.J. Solomon, and A. Pallares Bejarano. Synthesis of photographic quality facial composites using evolutionary algorithms. In Richard Harvey and J.Andrew Bangham, editors, *British Machine Vision Conference 2003*, volume 1, pages 221–230, 2003.

[37] C.A. Glasbey and K.V. Mardia. A review of image-warping methods. *Journal of Applied Statistics*, 25(2):155–172, 1998.

[38] Rafael C. Gonzalez and Richard E. Woods. *Digital image processing.* Prentice Hall, second edition, 2002.

[39] F. Gray. Pulse code communication. U.S. patent no. 2,632,058., March 1953.

[40] N. Greene and P. Heckbert. Creating raster omnimax images from multiple perspective views using the elliptical weighted average filter. *IEEE Computer Graphics & Applications*, 6(6):21–27, June 1986.

[41] J. J. Grefenstette. Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 16:122–128, 1986.

[42] N.D. Haig. Exploring recognition with interchanged facial features. *Perception*, 15:235–247, 1986.

[43] P.J.B Hancock. Evolving faces from principal components. *Behaviour Research Methods, Instruments and Computers*, 32(2):327–333, 2000.

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

[45] P. Heckbert. Fundamentals of texture mapping and image warping. Master's thesis, University of California at Berkeley, Department of Electrical Engineering and Computer Science, June 1989.

[46] B. Heisele, P. Ho, J. Wu, and T. Poggio. Face recognition: component-based versus global approaches. *Computer Vision and Image Understanding*, 91(1):6–12, 2003.

[47] A. Hill, C.J. Taylor, and T.Cootes. Object recognition by flexible template matching using genetic algorithms. In G. Sandini, editor, *European Conference on Computer Vision*, pages 852–856. Springer-Verlag, 1992.

[48] C.M. Hill, C.J. Solomon, and S.J. Gibson. Aging the human face - a statistically rigorous approach. In *The IEE International Symposium on Imaging for Crime Detection and Prevention*, pages 89–94. IEE, June 2005.

[49] J.H. Holland. *Adaptation in Natural and Artificial Systems.* Ann Arbor: University of Michigan Press, 1975.

[50] J.H. Holland. *Adaptation in natural and artificial systems.* Ann Arbor: University of Michigan Press, 1975.

[51] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.

[52] T.J. Hutton, B.F. Buxton, P. Hammond, and H.W.W. Potts. Estimating average growth trajectories in shape-space using kernel smoothing. *IEEE Transactions on Medical Imaging*, 22(6):47–753, 2003.

[53] C. Janikow. Inductive learning of decision rules from attribute-based examples: A knowledge-intensive genetic algorithm approach. TR 91-030, The University of North Carolina, Dept. of Computer Science, Chapel Hill, NC., 1991.

[54] I. Jolliffe. *Principal component analysis.* Springer-Verlag, 1986.

[55] J. T. Jolliffe. *Principal components analysis.* Springer-Verlag, second edition, 2002.

[56] K.A. De Jong. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems.* PhD thesis, University of Michigan, 1975.

[57] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Internation Journal of Computer Vision*, pages 321–331, 1988.

[58] C. Koehn and R.P. Fisher. Constructing facial composites with the mac-amug pro system. *Pyschology, crime & law*, 3:209–218, 1997.

[59] A. Lanitis, C. Taylor, and T. Cootes. Toward automatic simulation of aging effects on face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):442–455, 2002.

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

[61] D.C. Lay. *Linear Algebra and its Applications.* Addison Wesley, 1994.

[62] K.J. Lee and D.I. Perrett. Presentation time measures of the effects of manipulation in colour space on discrimination of famous faces. *Perception*, 26:733–752, 1997.

[63] H. Y. Mark Liao, C. C. Han, G. J. Yu, M. C. Chen, H. R. Tyan, and L. H. Chen. Face recognition using a face-only database: A new approach. In *3rd Asian Conference on Computer Vision*, pages 742–749, 1998.

[64] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164, 2004.

[65] B. Moghaddam and M.-H. Yang. Sex with support vector machines. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 960–966. MIT Press, 2001.

[66] C.M. Scandrett (nee Hill), C.J. Solomon, and S.J. Gibson. A person-specific, rigorous aging model of the human face. *Special Issue Pattern Recognition Letters: Vision for Crime Detection and Prevention*, 27(15):1776–1787, 2006.

[67] A.J. OToole and S. Edelman. Structural aspects of face recognition and the other race effect. *Memory and Cognition*, 22:208–224, 1994.

[68] A.J. OToole, T. Vetter, H. Volz, and E.M. Salter. Three-dimensional caricatures of human heads: distinctiveness and the perception of facial age. *Perception*, 26:719–732, 1997.

[69] A. Pallares-Bejarano. *Evolutionary Algorithms for Facial Composite Synthesis*. PhD thesis, University of Kent, 2006 (in press).

[70] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(559-572), 1901.

[71] J. Penry. *Looking at faces and remembering them: A guide to facial identification*. Blek Books, London, 1971.

[72] D.I. Perrett, K.A. May, and S. Yoshikawa. Facial shapes and judgments of female attractiveness. *Nature*, 368:239–242, 1994.

[73] S. Rakover and B. Teucher. Facial inversion effects: Parts and whole relationship. *Perception and Psychophysics*, 59(5):752–761, 1997.

[74] N. Ramanathan and R. Chellappa. Face verification across age progression. *IEEE Computer Vision and Pattern Recognition*, 2:462–469, June 2005.

[75] I. Rechenberg. *Evolution strategy: Optimization of technical systems by means of biological evolution*. Fromman-Holzboog, 1973.

[76] G. Rhodes. *Superportraits: Caricatures and recognition.* Hove: The Psychology Press, 1996.

[77] Y. Rosenthal, G. de Jager, and J. Greene. A computerised face recall system using eigenfaces. University of Cape Town., 1998.

[78] H.P. Schwefel. *Evolution and Optimum Seeking.* John Wiley, New York, 1995.

[79] J. Sergent. An investigation into component and configural processes underlying face perception. *British Journal of Psychology*, 75(2):221–242, 1984.

[80] K. Sims. Artificial evolution for computer graphics. *Computer Graphics*, 25(4):319–328, 1991.

[81] Sirchie. (comphotofit composite software). http://www.sirchie.com.

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

[83] C.J. Solomon, S.J. Gibson, and A. Pallares-Bejarano. Eigenfit - the generation of photographic-quality facial composites. One Day BMVA symposium at the Royal Statistical Society, March 2002.

[84] W.M. Spears. A formal analysis of the role of multi-point crossover in genetic algorithms. *Annals of Mathematics and Artificial Intelligence*, 5(1):1–26, 1992.

[85] M. Stegmann. Active appearance models: theory, extensions and cases. Master's thesis, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 2000.

[86] S. V. Stevenage. Can caricatures really produce distinctiveness effects? *British Journal of Psychology*, 86:127–146, 1995.

[87] G. Strang. *Linear Algebra and its Applications.* Harcourt Brace Jovanovich Inc, Orlando, Florida 32887, third edition, 1988.

[88] G. Syswerda. Uniform crossover in genetic algorithms. In *3rd International Conference on Genetic Algorithms*, 1989.

[89] J.W Tanaka and M.J. Farah. Parts and wholes in face recognition. *Quarterly Journal of Experimental Psychology*, 46A:225–245, 1993.

[90] K.T Taylor. *Forensic Art & Illustration.* CRC Press, 2000.

[91] B.P. Tiddeman, D.I. Perrett, and D.M Burt. A wavelet based method for prototyping and transforming the textural detail of images. Preprint available from http://psy.st-and.ac.uk/people.personal/bpt/publications.html.

[92] C. Tredoux, J. Rosenthal, L. Da Costa, and D. Nunez. Reconstructing faces with an eigenface composite system. SARMAC III, Boulder, Colorado, July 1999.

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

[94] J. Turner, G. Pike, N. Towell, R. Kemp, and P. Bennett. Making faces: Comparing e-fit construction techniques. *Proceedings of The British Psychological Society*, 7(1):78, 1999.

[95] J.A.J. Turner. *Applying Psychological Research to Facial Compositing: Featural and Configural Approaches, Schematic Faces, and Feature Saliency in E-FIT Construction.* PhD thesis, University of Westminster, 2005.

[96] T. Valentine. A unified account of the effects of distinctiveness, inversion, and race in face recognition. *The Quarterly Journal of Experimental Psychology*, 43A:161–204, 1991.

[97] T. Valentine and M. Endo. Towards an exemplar model of face processing: the effects of race and distinctiveness. *Quarterly Journal of Experimental Psychology*, 44A:671–703, 1992.

[98] A. Webb. *Statistical Pattern Recognition.* John Wiley & Sons, 2002.

[99] A.W. Young, D.C. Hay, K.H. McWeeny, B.M Flude, and A.W. Ellis. Matching familiar and unfamiliar faces on internal and external features. *Perception*, 14:737–746, 1985.

[100] A.W. Young, D. Hellawell, and D.C. Hay. Configurational information in face perception. *Perception*, 16:747–749, 1987.

[101] G. Zamora. (sketch artist). http://www.sketch-artist.com/.

[102] M. Zwicker, H. Pfister, J. van Baar, and M. Gross. Ewa splatting. *IEEE TVCG*, 8(3):223–238, 2002.