Imperial College London

Department Of Computing

Constructing 3D Morphable Facial Models capable of expressing emotion

by

James Booth (jab08)

Submitted in partial fulfilment of the requirements for the MSc Degree in Computing Science of Imperial College London

September 2012

Abstract

A framework for constructing a 3D Morphable Facial Model (3DMM) capable of expressing emotion is developed, including tools for landmarking input textured facial meshes, aligning and simplifying meshes, establishing dense correspondences between faces through the Thin Plate Spline warp, and subsequent Principal Component Analysis of shape and texture data. As a use case, a set of 3DMM's are constructed from faces taken from the BU-4DFE database. The expressive performance of these models is measured by a number of metrics, including how well the Lucas-Kanade algorithm fits our 3DMM to a range of facial images when compared against the popular Basel Face Model.

Acknowledgements

Warm thanks to my supervisor Dr. Stefanos Zafieriou for his constant guidance and support throughout this project. Special thanks also to Joan Alabort Medina, not only for his invaluable assistance throughout, but in particular for allowing me to utilize his implementation of the Lucas-Kanade image matching algorithm to test the models produced in this thesis.

Contents

Chapter 1

Introduction

The concept of a 3D Morphable Facial Model (3DMM) was first proposed in the pioneering paper of Blanz and Vetter [5]. A 3DMM is a statistical model of the human face, which allows for the creation of realistic novel faces by taking linear superpositions of a set of *eigenfaces* developed using Principal Component Analysis. All faces in the model are parameterised into shape and texture components that can be combined independently, allowing the model to reproduce a wide gamut of human faces.

The key challenge in constructing a morphable model is establishing dense correspondence between the input face meshes. In the initial work [5], an optical flow algorithm allowed for this to be done automatically, albeit with some drifting causing erroneous matchings. Patel and Smith [3] have since demonstrated that a superior mapping can be found by applying a TPS warp on a sparse set of facial landmarks, using this as an interpolation function to find dense correspondences. This has even been expanded to work on a dynamic (time varying) set of input meshes, as demonstrated recently by Cosker [8]. In this paper, we adapt Patel's TPS warp approach, prescribing a detailed account of all steps needed to construct a model from raw 3D data.

Due to the great deal of flexibility provided, 3DMMs have found a wide range of practical uses. One example is in the special effects industry, where models are trained on frames of video to produce realistic 3D heads that can be repositioned or animated in the scene at the director's discretion [4].

1.1 Contributions

The main contribution of this thesis is a detailed analysis of the techniques that can be used to build a 3D Morphable Face Model from raw 3D facial scans. A full pipeline is provided resulting in a statistical model which is then tested in it's ability to represent novel faces using an implementation

[13] of the well known Lucas-Kanade algorithm [14].

The most novel aspect of this work is that faces displaying a mixture of different emotions are used in the construction of the models. The effect this has on the principal components of the models produced is analysed in some detail.

The second contribution is a suite of software tools which prepare a morphable model based on the pipeline outlined in this thesis. This includes Matlab code for importing and organising raw 3D meshes, a tool to assist in the landmarking of faces, and the subsequent processing which results in a set of models in dense correspondence and analysed in terms of their principal components. All rasterization in the pipeline is done using a bespoke OpenGL renderer, which can also be used for interactive visualization of any face mesh.

1.2 Structure of this report

Chapter 2 mathematically defines faces in terms of their shape and texture, as well as defining what exactly a 3DMM is. It also includes a full overview of the pipeline that will be explained subsequently in later chapters. Chapter 3 explains the importance of accurately landmarking faces - that is, picking out salient points such as *nose-tip* or *chin*. Chapter 4 covers how landmarked faces can then be aligned in a shape space, removing arbitrary pose effects. Aligned faces can then be brought into dense correspondence by projecting, warping, and resampling the meshes, a process described in chapter 5.

Chapter 6 takes a set of faces in dense correspondence and finds the key Principal Components or *eigenfaces* present in the data. An understanding of the statistical distribution present in the data is developed which allows for the construction of new faces by taking linear combinations of these eigenfaces.

Finally, a number of experiments are performed in chapter 7 to quantitatively and qualitatively analyse the expressive performance of a set of morphable models constructed using the pipeline outlined in this thesis, with conclusions drawn and directions for future work considered in chapter 8.

Chapter 2

3D Morphable Models

2.1 Face definition

In this thesis, a face $\mathbf{F}$ is defined by four objects. The shape, $\mathbf{S}$, is a collection of n_c coordinate points $\boldsymbol{c} = (x, y, z) \in \mathbb{R}^3$, and is stored as a $3 \times n_c$ matrix

$$\mathbf{S} = \begin{pmatrix} x_1 & x_2 & \dots & x_{n_c} \\ y_1 & y_2 & \dots & y_{n_c} \\ z_1 & z_2 & \dots & z_{n_c} \end{pmatrix} \tag{2.1}$$

as this lends itself to manipulation by transformation matrices. The texture image is a rectangular colour image of dimensions $w \times h$, which defines the entire region of colour values that may be sampled from to texture the face. It is represented as an $3 \times n_t$ RGB matrix

$$\mathbf{T} = \begin{pmatrix} r_1 & r_2 & \dots & r_{n_t} \\ g_1 & g_2 & \dots & g_{n_t} \\ b_1 & b_2 & \dots & b_{n_t} \end{pmatrix} \tag{2.2}$$

where $n_t = w \times h$.

Coordinates are arranged into n_t triangles via a $3 \times n_t$ triangle list matrix **tri**, and each coordinate has associated with it a texture coordinate $c^t = (s, t)$ where $s, t \in [0, 1]$. c^t indexes into the texture image with the length and height normalised to 1 (e.g. $(0, 0)$ = *bottom left*, $(1, 1)$ = *top right*).

Figure 2.1 shows how all four objects define part of a face. As the triangle list and texture coordinates are rarely changed in this paper, a face is defined using the notation $\mathbf{F}(\mathbf{S}, \mathbf{T})$.

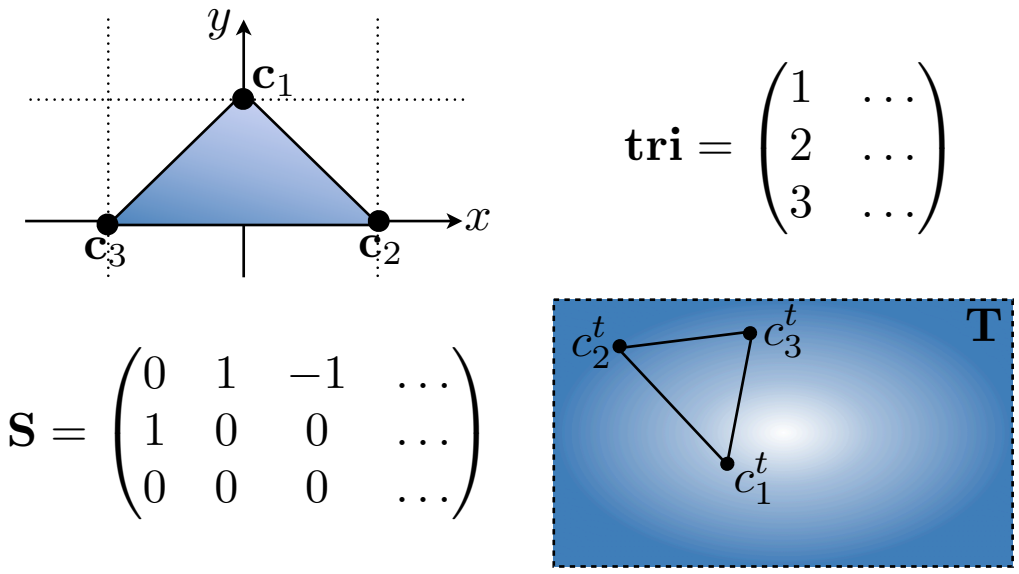

Figure 2.1: Demonstration of how the first triangle in the model (top left) is defined by the first column in the triangle list (top right). Its shape is thus given by column entries $(1, 2, 3)$ in the shape matrix (bottom left), while the texture is mapped from the triangular region of the texture image defined by the corresponding three texture coordinates (bottom right).

2.2 Morphable Model definition

A 3D Morphable Model, **3DMM**, is a collection of k faces $\{\mathbf{F}_1, \mathbf{F}_2, \dots, \mathbf{F}_k\}$ that possess the property that new novel faces can be constructed by taking linear combinations of the faces in the collection

$$\mathbf{F}_{novel} = \sum_{i=1}^{k} \alpha_i \mathbf{F}_i \qquad \forall i, \alpha_i \in \mathbb{R} \tag{2.3}$$

In general, facial meshes cannot be 'added' to one another. Much of the work in this thesis (chapters 3-5) revolves about manipulating the k faces used to build the model to generate a sensible notion of addition. Furthermore, to be of value the model needs to quantify the likelihood of a given generated face existing, which means understanding and qualifying the range of acceptable values the coefficients of each face can take. This is the second part of the work, tackled in chapter 6.

2.3 Addition and dense correspondence

Assume two faces are composed of the same number of points, and have an identical connectivity (that is, they share the same triangle list). Even in such a case, it is not possible to add the two meshes together, as there is no sense of which coordinate on one face should be added to which on the other. Only if given such an bijective mapping, between every coordinate in one face to another, can addition take place. Such a mapping is termed *dense correspondence*, and establishing dense

correspondence between all the faces used in building the model is the key challenge in creating a 3DMM.

2.4 3DMM construction pipeline

A terse overview of the entire pipeline is as follows:

1. A set of raw 3D facial meshes with corresponding textures are the inputs to the pipeline.

2. All faces in the set are landmarked, meaning that a sparse set of points on each mesh with a specific semantic meaning (such as *nose tip* or *chin*) are identified. This is firstly done with a state of the art algorithm, before manual human improvements are made. All points across the set of faces sharing the same landmark are said to be in correspondence. As only a tiny subset of each mesh is in correspondence, meshes on the whole are said to be a state of sparse correspondence.

3. Faces are brought into a consistent shape space by performing a Generalised Procrustes Analysis.

4. The face meshes are flattened by representing them in a cylindrical coordinate basis.

5. A warping function is generated for each flatted face. The warping function guarantees a map of the face's landmarked coordinates to the mean landmark coordinates of the set of faces, and in doing so also deforms all other coordinates around the landmarks. We will analyse two such warping functions, the familiar Piecewise Affine warp, and the Thin-Plane Spline warp.

6. Post warping, the flattened faces are postulated to be in alignment. A resampling of the meshes from this state yields a set of meshes in dense correspondence.

7. Principal Component Analysis is performed on the resulting faces, generating an orthonormal basis on which linear combinations can be taken to build novel faces. The statistical likelihood of each principal component appearing in a generated faces is calculated, and from this the likelihood of the overall generated face existing is quantised.

Chapter 3

Landmarking Faces

The representational ability of a 3DMM is largely governed by two factors - the quality and diversity of the input faces, and how effectively dense correspondence can be established between them. Our approach to establishing dense correspondence is by first landmarking a set of predefined sparse correspondences on every face, before using warping techniques to interpolate these into a dense matching across the set. As is to be expected with such a strategy, the quality of the interpolated output is highly sensitive to the accuracy of the input landmarks, and as such, a manual approach is taken to minimise landmarking errors.

3.1 Landmark definition

A landmark is a salient coordinate with a unique semantic meaning that can be readily identified on every face in the set. Expanding the notation developed earlier, each face has associated with it a set of n_l landmark coordinates, $\mathbf{A} = \{ \boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_{n_l} \}$, where $\boldsymbol{a}_i \in \mathbb{R}^3$ and $\mathbf{A} \subset \mathbf{S}$. The i'th landmark $\boldsymbol{a}_i$ is defined for each face in **3DMM**, and represents the position of a certain feature on each face. As such, the landmark coordinates are in correspondence across the set of faces.

3.2 Landmarking approach

Previous approaches to landmarking fall into one of two categories. Some, like in the original work by Blanz and Vetter [5] start with the assumption that every coordinate on each face is a landmark, and then make use of optical flow algorithms run between different faces to infer how landmarks on different faces correspond. This has the major advantage that the entire process is automated, yet conversely, the process is nebulous, with no easy way to correct mistaken assignments.

Others [3] invest the time to manually landmark key points before using warping techniques to

establish dense correspondences for all coordinates. As already suggested, our approach aligns more closely with the latter, with the distinction that our technique combines algorithmic and manual strategies to reduce the amount of time spent landmarking.

Annotating facial images is critical to a number of problems in computer vision, like facial recognition and manipulation, and as such, much work [15] has been done on developing robust image landmarking algorithms, even ones that can landmark faces displaying emotions with a high degree of success. Our approach is to use such a cutting edge facial landmarker to coarsely landmark each faces texture image. A tool was developed which allows for rapid assessment of the quality of these automated landmarks, and for subsequent corrections to be made by hand if necessary. In a final automated stage, these texture landmarks are mapped onto coordinate landmarks consistent with the definition given in section 3.1. We feel this strategy combines the benefits of an automatic approach (consistency of positioning, speed) with the accuracy that currently only the human eye can provide.

3.3 Landmark results

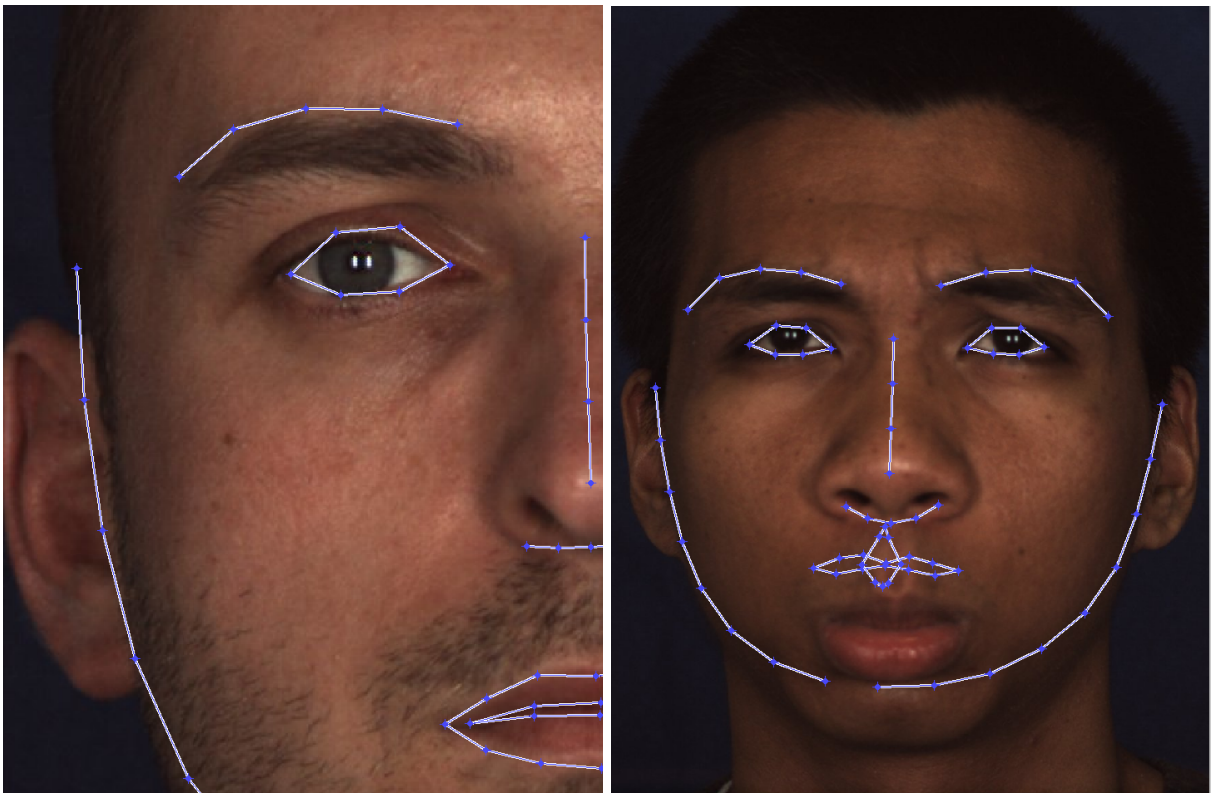

Figure 3.1: Two example results from the landmarking algorithm, highlighting how the manual correction required varied from minor (left) to significant (right).

The algorithm in question defines a set of 68 landmarks as pixel coordinates on the texture

images. Figure 3.1 shows two outputs from the automatic landmarking algorithm. As can be seen the quality of the automatic landmarking varies based on a variety of factors - the lighting on the face, expression, pose variation, and skin tone all influence the output.

It was found that the number of landmarks around the eyes and mouth, areas both critical to identity and ones that undergo considerable change when displaying emotions, were insufficient to fully capture the shape of the features for later correspondence interpolation. As such an extra 44 manual landmarks are added in these areas once the initial points are set. An example of a completely landmarked face is given in figure 3.2.

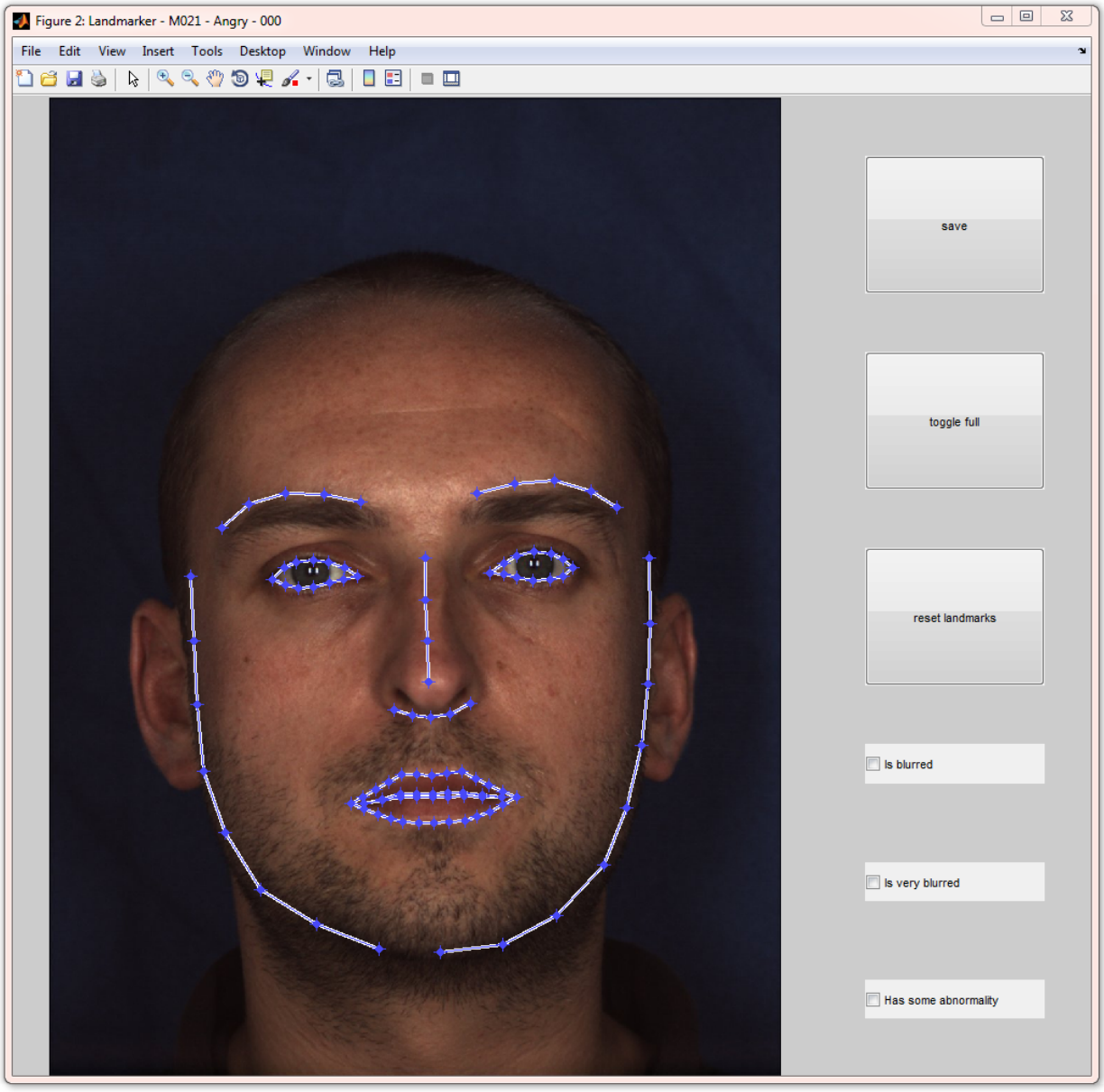

Figure 3.2: Landmarker application showing finalised texture landmarks for a face. Extra landmarks are added around the eyes and mouth to capture as much emotional detail as possible (c.f. figure 3.1).

3.4 Coordinate landmarks from texture landmarks

Given the final set of 100 pixel landmarks the actual coordinate landmarks **A** need to be derived. A connection between every coordinate and it's associated pixel location on the texture image is encoded in the texture coordinate matrix. The approach taken is to find the least squares distance between these associated pixel location's and the landmark pixel in question, with the corresponding coordinate being assigned.

In the case of a tie, the distances between the two conflicting pixel landmarks and their *second* nearest neighbours is evaluated, with the landmark with the nearer second neighbour being reassigned to that coordinate. This greedy algorithm ensures the net assignment error is minimised.

Chapter 4

Generalised Procrustes Analysis

Faces at this stage of the pipeline have been landmarked in a consistent manner, setting the stage for a group-wide processing of the meshes to establish dense correspondence. However, each face shape is currently expressed in it's own coordinate space, inhibiting such operations. Generalised Procrustes Analysis is employed to alleviate this issue, expressing each mesh in a single consistent shape space.

4.1 Definition of a Shape Space

Consider the case of a simple test shape (perhaps a small wooden cube) being captured by two different, but theoretically perfect 3D capture devices. Although the important information (the shape) is unchanged, the data captured will vary significantly based on on the nature of the devices and the position of the subject. What is always true, however, is that there exists a particular set of transforms (specifically, a translation, scale, and rotation) that will map one device's data to the other. This fact is reflected in the widely used definition of shape given by Kendall [11]: *the geometrical information that remains after scale, rotation, and translation effects are removed.*

Of course this definition can equally well be applied to different shapes of the same classification, such as landmarked face meshes. If all effects of rotation, translation, and scale that exist between the different face landmarks could be filtered out, what would remain would be only the physical differences separating each of the subject's face from each other. In this representation, the faces are said to be in a shared *shape space* [11].

4.2 The Procrustes mean and Procrustes distance

Continuing with the notation developed in chapter 4, $\bar{\boldsymbol{a}}_i$ is defined as the i'th Procrustes mean landmark across all faces

$$\bar{\boldsymbol{a}}_i = \frac{1}{n_f} \sum_{j=1}^{n_f} \boldsymbol{a}_i^j \tag{4.1}$$

where $\boldsymbol{a}_i^j$ is the i'th landmark coordinate on the j'th face. The Procrustes distance E_p is then defined to be the summed squared distance between the each of the corresponding landmark coordinates and the Procrustes mean across all n_f faces in the set

$$E_p = \sum_{i=1}^{n_l} \sum_{j=1}^{n_f} |\bar{\boldsymbol{a}}_i - \boldsymbol{a}_i^j|^2 \tag{4.2}$$

Generalised Procrustes Analysis [11] is an iterative algorithm which finds the optimum rotation, scale, and translation operations for each face in order to minimise E_p. If at the end of the process E_P is zero, then the input landmarks were of the same shape. If not, the value of E_p is a measure of the dissimilarity in the shape class. The algorithm is now prescribed in full.

4.3 Generalised Procrustes Analysis

1. For each face the centroid of it's landmarks is calculated

$$\boldsymbol{a}_{cent}^j = \sum_{i=1}^{n_l} \boldsymbol{a}_i^j \tag{4.3}$$

   with each faces coordinates subsequently translated so that the centroid of the landmarks is now at the origin

$$\mathbf{s}_j \leftarrow \mathbf{s}_j - \begin{bmatrix} \boldsymbol{a}_{cent}^j & \boldsymbol{a}_{cent}^j & \dots & \boldsymbol{a}_{cent}^j \end{bmatrix} \tag{4.4}$$

2. A landmark shape metric is defined

$$m = \sum_{i}^{n_l} |\boldsymbol{a}_i - \boldsymbol{a}_{cent}|^2 \tag{4.5}$$

   and the metric calculated for each of the faces. $\mathbf{m}_j$ is applied to the j'th face shape

$$\mathbf{s}_j \leftarrow \mathbf{m}_j \mathbf{s}_j \tag{4.6}$$

where $\mathbf{m_j}$ is a scale matrix that when applied to the j'th face normalises it's metric length

$$\mathbf{m}_j = \begin{pmatrix} 1/m_j & 0 & 0 \\ 0 & 1/m_j & 0 \\ 0 & 0 & 1/m_j \end{pmatrix} \tag{4.7}$$

3. The Procrustes mean of the landmarks is calculated as in equation 4.1. Let $\bar{\mathbf{A}}$ be the set of all mean landmarks. For each face, the Singular Value Decomposition of the covariance of the shape landmarks with the mean landmarks is calculated

$$\text{cov}(\mathbf{A}^j, \bar{\mathbf{A}}) = \mathbf{U}_j \mathbf{\Sigma}_j \mathbf{V}_j^T \tag{4.8}$$

where $\mathbf{V}_j \mathbf{U}_j^T$ is the optimal rotation matrix to align the j'th faces landmarks as closely as possible to the mean landmarks.

4. Each face's shape matrix is once again updated

$$\mathbf{s}_j \leftarrow \mathbf{V}_j \mathbf{U}_j^T \mathbf{s}_j \tag{4.9}$$

before the mean Procrustes landmarks are recalculated. If the change in the mean Procrustes landmarks between the two iterations is sufficiently small, stop. Else, the algorithm is repeated from step 1.

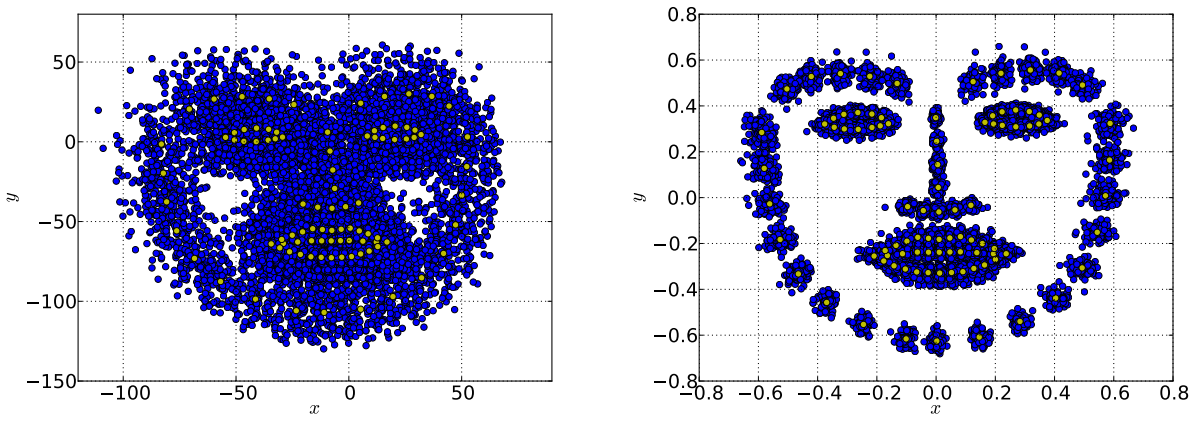

Figure 4.1: Left: Landmark coordinates pre-alignment (yellow: mean landmarks). Right: Faces post alignment. Note the meshes have also been straightened to face down the z axis (see section 4.4).

4.4 Post alignment

Figure 4.1 shows the position of 110 faces before and after Generalised Procrustes Analysis. Although all faces are now in the same shape space, it's numerically convenient to align the faces to face down the z axis. To do so, a facing vector is determined by constructing a vector from the centroid of the mean landmarks to a specially assigned nose tip mean landmark

$$d = \bar{\boldsymbol{a}}_{nose} - \bar{\boldsymbol{a}}_{cent} \tag{4.10}$$

With an assuption that y always faces up, this allows for a trivial rotation to be made to all faces to straighten them, removing any pose (see figure 4.1).

Chapter 5

Achieving Dense Correspondence: Thin-Plate Splines

The purpose of this stage of the pipeline is to establish dense correspondence between all the facial meshes. Intuitively, this means that each face is built from the same number of coordinates connected in the same way, with each coordinate having the same meaning across all faces. Formally, we seek a set of facial meshes where all members are of the same topology, with identical connectively, with a bijective mapping between any coordinate in a given face to another coordinate in all other members of the set.

The starting point at this stage of the pipeline is a set of faces with differing topologies and connectivities, and with only a sparse set mesh coordinates in correspondence. The former statement necessitates some form of mesh resampling to enforce a consistent topology across the set, while the latter suggests that some form of interpolation needs to be employed in order to find a likely dense correspondence of the entire mesh from the sparsely landmarked coordinates. The technique outlined here achieves both of these requirements by borrowing warping techniques common in 2D image manipulation and applying them to flattened meshes.

5.1 Warp Functions on Facial Meshes

The approach taken for establishing dense correspondence is borrowed from techniques commonly applied in 2D image warping. There, a set of n source landmarks $S_1, S_2, \ldots, S_n$ where $S_i = (x_i, y_i)$ are used to drive a warping function which transforms a source image to fit against a set of n corresponding target landmarks $T_1, T_2, \ldots, T_n$ with $T_i = (x'_i, y'_i)$. The warped image will have it's landmarks coincident with the reference points, while the manner in which the rest of the image

is disturbed varies depending upon the warping function used. We first explore two such warping functions in the context of image warping, before then showing how facial meshes can be flattened into an appropriate representation for the application of these ideas.

5.1.1 Piecewise Affine warp

Perhaps the most straightforward warping function is the Piecewise Affine warp. Firstly the Delaunay Triangulation [9] of the source and target landmarks is computed. For a given triangle built from the coordinates **a**, **b**, & **c**, any internal point **r** can be represented by Barycentric coordinates $(\alpha, \beta, \gamma) \in [0, 1]$, where

$$\mathbf{r} = \alpha\mathbf{a} + \beta\mathbf{b} + \gamma\mathbf{c} \tag{5.1}$$

under the constraint that $\alpha + \beta + \gamma = 1$. Each coordinate is encoded in such a way with regards to its containing triangle. Using the same Barycentric coordinates and triangulation with regard to the target landmarks results in an output image where each triangular region is linearly deformed, as demonstrated in figure 5.1.

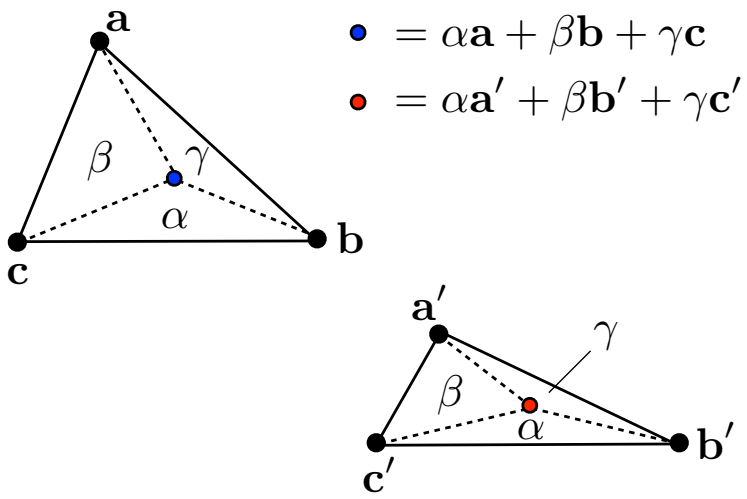

Figure 5.1: The contents of a warped triangle can be linearly deformed by ensuring the Barycentric coordinates of a given contained coordinate remain the same post warp. Note the geometrical interpretation that the relative areas of the triangle are given by the coefficients α, β, and γ, and these relative areas are maintained throughout the warp.

5.1.2 Thin Plate Splines

Motivation

Due to the definition of Barycentric coordinates, a Piecewise Affine warp is continuous along the boundaries of the triangles, however it is not first order smooth, which leads to unphysical sharp

boundary artefacts. Thin Plate Splines provide a superior warp which is everywhere smooth and continuous.

At the heart of the TPS warp is the biharmonic equation in 2 dimensions

$$\nabla^4 \phi(x, y) = 0 \tag{5.2}$$

an equation which occurs frequently in continuum mechanics when considering the bending behaviour of physical systems under load. The fundamental solution to equation 5.2 is

$$\begin{aligned} U(x, y) &= (x^2 + y^2)^2 \log(x^2 + y^2)^2 \\ U(r) &= r^2 \log r^2 \end{aligned} \tag{5.3}$$

where $r = \sqrt{x^2 + y^2}$ is the distance from the origin, and we use the notation $U(r)$ to distinguish the fundamental solution from a general one.

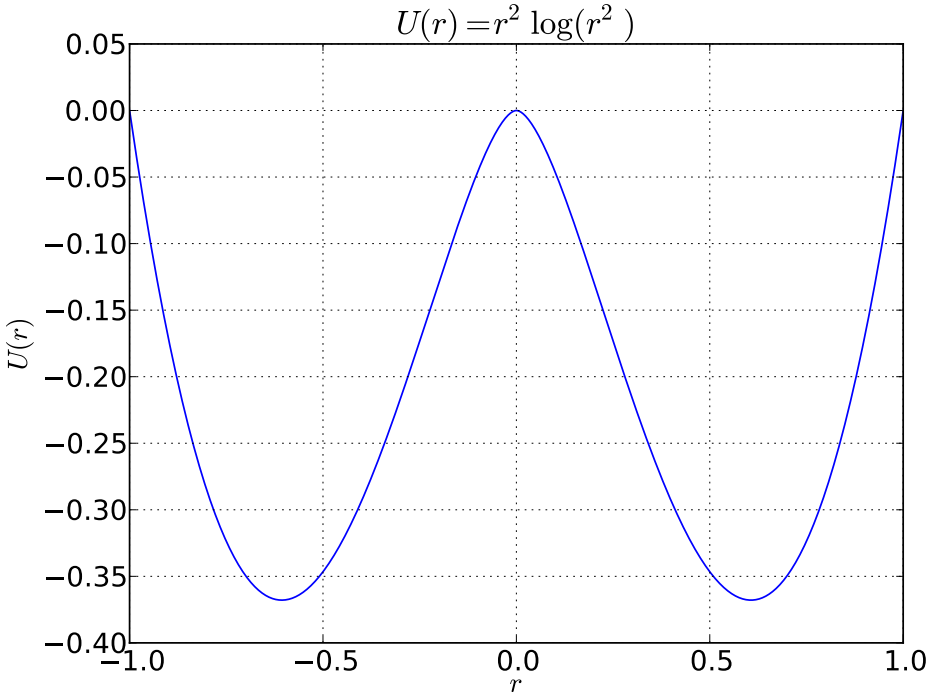

Figure 5.2: The radial form of the fundamental solution to the biharmonic equation

As will be demonstrated, if a set of n constraints of the form $\phi(x_i, y_i) = t_i$ are imposed, equation 5.2 can be used for calculating interpolations. If we further stipulate that the undisturbed function lies in the plane $\phi = 0$ by demanding that in polar coordinates

$$\lim_{r \to \infty} \phi(r, \theta) = 0 \qquad \theta = \tan^{-1}(y/x) \tag{5.4}$$

then it can be shown [6] that any solution to equation 5.2 under these constraints can be expressed in the form

$$\phi(x,y) = \sum_{i=0}^{n} w_i \, U(|S_i - (x,y)|) \qquad\qquad w_i \in \mathbb{R} \tag{5.5}$$

which is a weighted mixture of the fundamental solution evaluated at the absolute distance from the point in question to each of the landmarks. This form is suggestive that equation 5.3 can be thought of as the two dimensional analogue to the one dimensional cubic spline basis function $|x|^3$ used in the construction of the ubiquitous Bezier curve, an intuition which it transpires is correct [6].

As an example, consider the case where

$$\begin{aligned} \phi(+1,0) &= -1/2 \qquad & \phi(0,+1) &= +1/2 \\ \phi(-1,0) &= -1/2 \qquad & \phi(0,-1) &= +1/2 \end{aligned} \tag{5.6}$$

The numerical solution to equation 5.2 for such a set of constraints is graphically shown in figure 5.3. This shape has a physical realisation - it is the shape an infinite flat sheet of steel $\phi(x,y) = 0$ would take if forced to pass through the given four constraints at right angles to the flat surface. As such, it has a further physical interpretation, which is that this function minimises an energy associated with the bending between the landmarks. This bending energy E_B, is proportional to

$$E_B \propto \iint_A \nabla^4 \phi(x,y) \, \mathrm{d}x \, \mathrm{d}y \tag{5.7}$$

and as such TPS acts to minimise squared second order derivatives in the solution.

TPS as a warping function - in-plane landmarks

A slightly different interpretation of ϕ is used in TPS warping. Instead of visualising ϕ as a surface orthogonal to x and y, we now consider it as a disturbance in either the x or y direction, yielding a warped ordinate. As such, a TPS warp is a vector-valued function $\Phi(x,y) \to (x',y')$ described by the composition two independent functions

$$\begin{aligned} x' &= x + \phi_x(x,y) \\ y' &= y + \phi_y(x,y) \end{aligned} \tag{5.8}$$

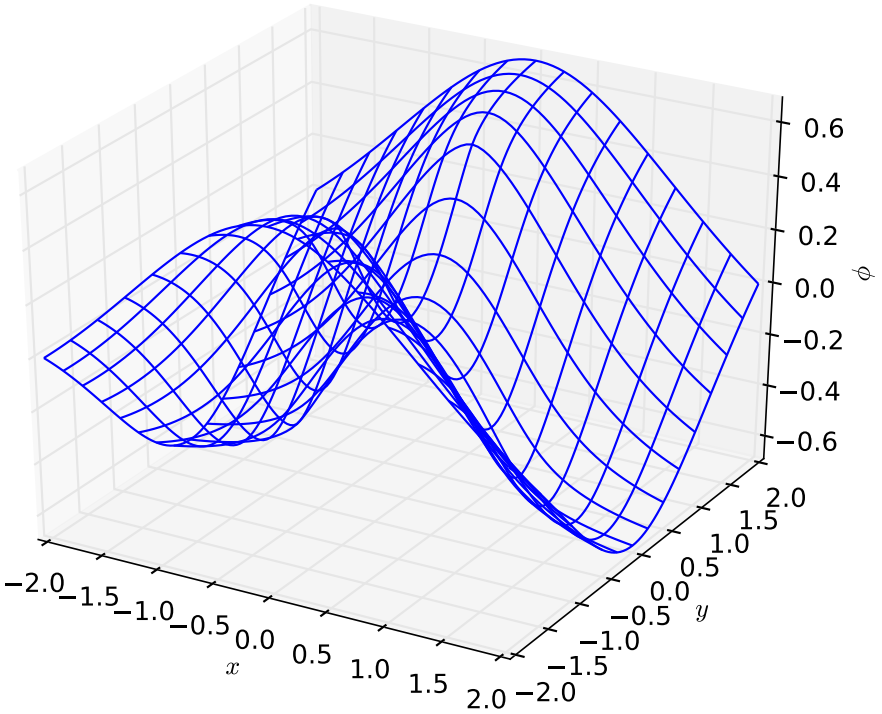

Figure 5.3: Solution to the biharmonic equation $\phi(x, y)$ for the constraints given in equation 5.6.

A simple rearrangement of equation 5.8 evaluated at S_i reveals the interpretation of the value of the two biharmonic functions involved in the warp

$$\begin{aligned} \phi_x(x_i, y_i) &= x'_i - x_i \\ \phi_y(x_i, y_i) &= y'_i - y_i \end{aligned} \tag{5.9}$$

in other words, the value of $\phi_x(S_i)$ is the displacement of the target landmark from the source landmark in the x dimension, $\phi_y(S_i)$ the displacement in the y dimension.

5.1.3 Comparison of TPS and Piecewise Affine warps

Figure 5.4 shows a comparison of TPS and Piecewise Affine warp functions. The y values of the pixels in the left image are generated from equation 5.8, with a $\phi_y(x, y)$ similar in form to figure 5.3, only with a set of constraints particular to how the landmarks change in this warp (x is unaltered).

As a comparison, a near identical warp configuration calculated with a Piecewise Affine transform is given alongside, the only difference being the addition of a set of fixed landmarks at the four corners of the image. Due to the localised nature of the Piecewise Affine these extra points do not influence the warp of the shape in the centre, but do allow for a visualization of how the triangle boundaries are continuous by how the grid lines are warped.

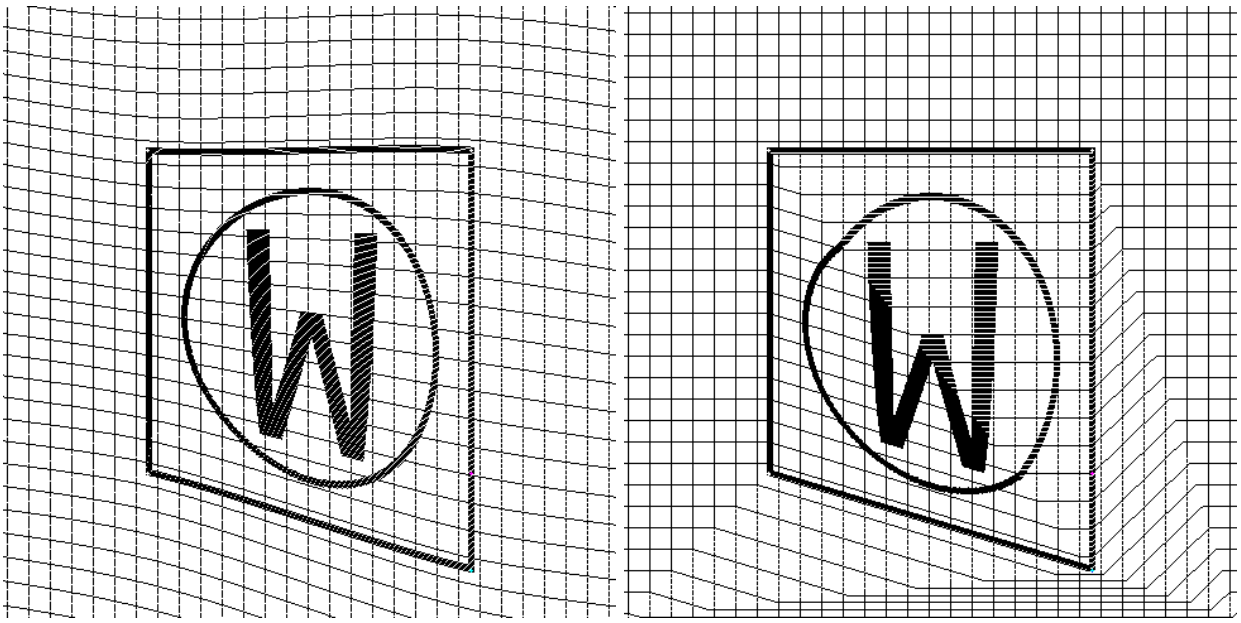

Figure 5.4: Comparison of warp functions. Left: TPS. Right: Piecewise Affine Warp. Note while both are continuous, only TPS is smooth.

Algebra of calculation

A more succinct form for the TPS warp can be attained by relaxing the condition given in equation 5.4. Now the general solution ϕ^w includes an affine component

$$\phi^w(x,y) = a_c + a_x x + a_y y + \sum_{i=0}^{n} w_i \, U(|S_i - (x,y)|) \tag{5.10}$$

dictating the behaviour of ϕ at infinity. This allows for equation 5.8 to be rewritten as

$$\begin{aligned} x' &= \phi_x^w(x,y) \\ y' &= \phi_y^w(x,y) \end{aligned} \tag{5.11}$$

or, as a single vector valued function

$$\begin{aligned} \Phi(x,y) &= \begin{bmatrix} \phi_x^w(x,y) & \phi_y^w(x,y) \end{bmatrix} \\ \Phi(x,y) &= (x',y') \end{aligned} \tag{5.12}$$

as the addition of the relevant source coordinate (x or y) in equation 5.8 is subsumed in the affine component.

Referring back to the form of ϕ^w given in 5.10 we define a vector of weightings

$$W = \begin{pmatrix} w_1 & w_2 & \dots & w_n & a_c & a_x & a_y \end{pmatrix}^T \tag{5.13}$$

and note that a given configuration of W defines $\phi^w(x, y)$ for all input points.

As stated in equation 5.11 however, a TPS warp is defined by *two* such biharmonic functions, one for each dimension. The full state of the warp can thus be captured in a $n \times 2$ warping matrix $\mathbf{W}$, where

$$\mathbf{W} = \begin{bmatrix} W_x & W_y \end{bmatrix} \tag{5.14}$$

with W_x and W_y being the weighting vectors for ϕ_x^w and ϕ_y^w respectively. For the special case of source landmark inputs we note that $\mathbf{W}$ must map to their respective target landmarks. It is this closed form solution that can be exploited numerically to find the global warping function given by ϕ_x^w and ϕ_y^w.

Following the approach taken by Bookstien [6] we define, using

$$r_{ij} = |S_i - S_j| \tag{5.15}$$

a kernel matrix $\mathbf{K}$

$$\mathbf{K} = \begin{bmatrix} 0 & U(r_{12}) & \dots & U(r_{1n}) \\ U(r_{21}) & 0 & \dots & U(r_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ U(r_{n1}) & U(r_{n2}) & \dots & 0 \end{bmatrix} \tag{5.16}$$

and slightly augmented source and target matrices, $\mathbf{S}$ and $\mathbf{T}$ respectively

$$\mathbf{S} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \\ y_1 & y_2 & \dots & y_n \end{bmatrix}^T \tag{5.17}$$

$$\mathbf{T} = \begin{bmatrix} x'_1 & x'_2 & \dots & x'_n & 0 & 0 & 0 \\ y'_1 & y'_2 & \dots & y'_n & 0 & 0 & 0 \end{bmatrix}^T \tag{5.18}$$

For ease of computation, the composite $(n+3) \times (n+3)$ matrix $\mathbf{L}$ is also constructed

$$\mathbf{L} = \begin{bmatrix} \mathbf{K} & \mathbf{S} \\ \mathbf{S}^T & \mathbf{O} \end{bmatrix} \tag{5.19}$$

where $\mathbf{O}$ is a 3×3 matrix of zeros, and we note $\mathbf{L}$ is solely defined by the position of the source landmarks.

It can now be seen by careful inspection that the entire problem can be restated elegantly in this vector notation as

$$\mathbf{T} = \mathbf{LW} \tag{5.20}$$

revealing that a given TPS warp is calculated by solving the linear problem

$$\mathbf{W} = \mathbf{L}^{-1}\mathbf{T} \tag{5.21}$$

which is well defined as long as no source landmarks are coincident.

5.2 Flatting Facial Meshes

In the present application we have facial manifolds embedded in 3D space, so before any warping function can be applied the face meshes need to be flattened into a 2D representation. A crude but effective approximation is to consider the human face as a cylinder - expressing the mesh in a cylindrical coordinate representation about the centre of the head $(x, y, z) \rightarrow (r, \theta, z)$ then allows for a view of the entire face mesh in (θ, z), with the radial distance r acting as a depth map at each point.

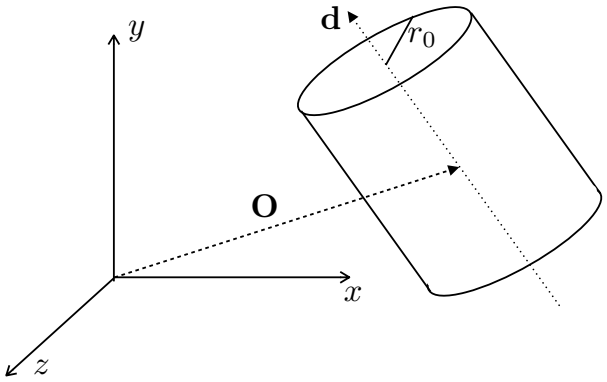

Figure 5.5: The parameters which govern the shape of a cylinder

As shown in figure 5.5, any cylinder can be parameterised by a choice of origin $\mathbf{O} = (\mathrm{O}_x, \mathrm{O}_y, \mathrm{O}_z)$,

a radial distance r_0, and the major axis of symmetry, $\boldsymbol{d}$. In the present application, $\boldsymbol{d}$ is fixed to $\boldsymbol{y}$, with the choice of $\mathbf{O}$ and r_0 being dictated by minimising the sum of squared perpendicular distances from the mean landmark points to the surface of the cylinder. This is an identical problem to finding an optimal circle fit to the landmark points after they have been projected down into the (x, z) plane. This is a non-trivial problem, and we adopt Taubin's method [1] to find a solution. Once found, we transform all faces to a frame of reference where the origin is zero

$$\begin{aligned} x_c &= x - \mathrm{O}_x \\ y_c &= y - \mathrm{O}_y \\ z_c &= z - \mathrm{O}_z \end{aligned} \tag{5.22}$$

so the cylindrical mapping is given by

$$\begin{aligned} r &= \sqrt{x_c^2 + z_c^2} - r_c \\ \theta &= -\tan^{-1}(z_c/x_c) \\ z &= z_c \end{aligned} \tag{5.23}$$

with $\theta \in \,]-\pi, \pi]$ so the discontinuity in the range occurs at the back of the head. Figure 5.6 demonstrates the unwrapping found based on the Procrutes mean landmarks.

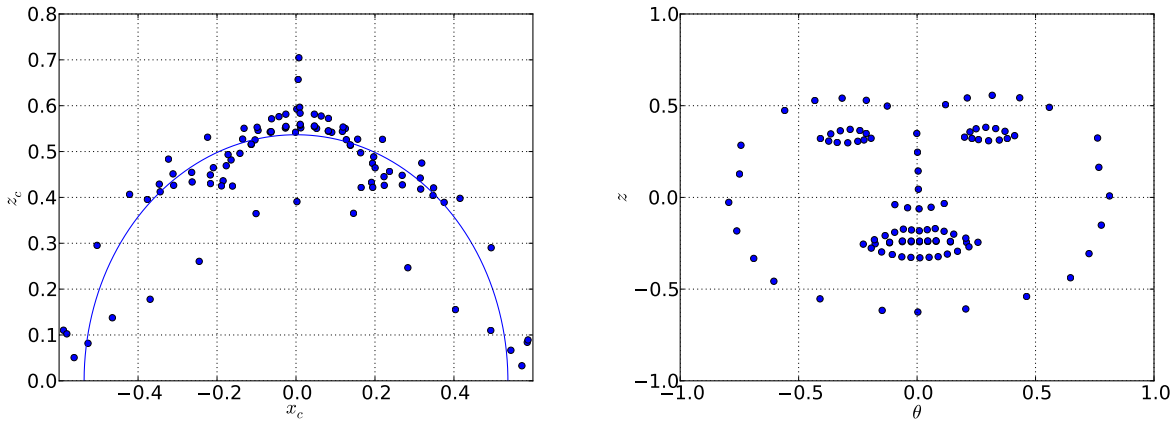

Figure 5.6: Example of the optimal cylindrical unwrapping found by Taubin's method. Left: Optimal cylindrical fit found using Taubin's method. Right: The mean landmarks in the cylindrical representation (θ, z)

An example of a face in it's cylindrical coordinate representation is given in figure 5.7. We note that there is no loss of data as the mapping is bijective, and, as no elements of the typical human face are self obscuring in (θ, z), we have a full view of the 3D mesh in this 2D representation, thus

the location of a point with a certain semantic meaning on the face can be described as a solely in terms of a (θ, z) coordinate.

Figure 5.7: An example face unwrapped into the cylindrical coordinates shown in figure 5.6.

5.3 Sampling in Dense Correspondence

Once all faces are represented in the same (θ, z) space, a per-face TPS Warp $\Phi(\theta, z)$ is constructed mapping the face landmarks to the mean landmarks found in the Procrustes Alignment. The warp is used to displace each of the n_c coordinates that make up the face in turn, eg

$$\boldsymbol{c}_j^{c\prime} = \Phi(\boldsymbol{c}_j^c) \tag{5.24}$$

where $\boldsymbol{c}_j^c = (\theta_j, z_j)$ is the j'th coordinate $\boldsymbol{c}$ in the cylindrical representation. Once warped, the original per face triangle list and texture map is still perfectly valid, and can thus be used to construct the TPS warped texture, an example of which is given in figure 5.8.

We have now forced the alignment of all landmark points across all faces. We further postulate that the TPS warp acts as a correspondence interpolant, shifting all inter-landmark face elements into dense correspondence. However, this is only true of the texture data, up to this point we have neglected how to establish dense correspondence between the original coordinates in each face, $\boldsymbol{c}_j$.

5.3.1 Use of a coordinate framebuffer

Our solution is to attach to each warped coordinate an extra piece of information - it's *unwarped non-cylindrical* coordinate value $\boldsymbol{c}_j = (x_j, y_j, z_j)$. These values are stored in the usual RGB color channels but as floating point numbers. Using OpenGL's ability to render to multiple targets, a

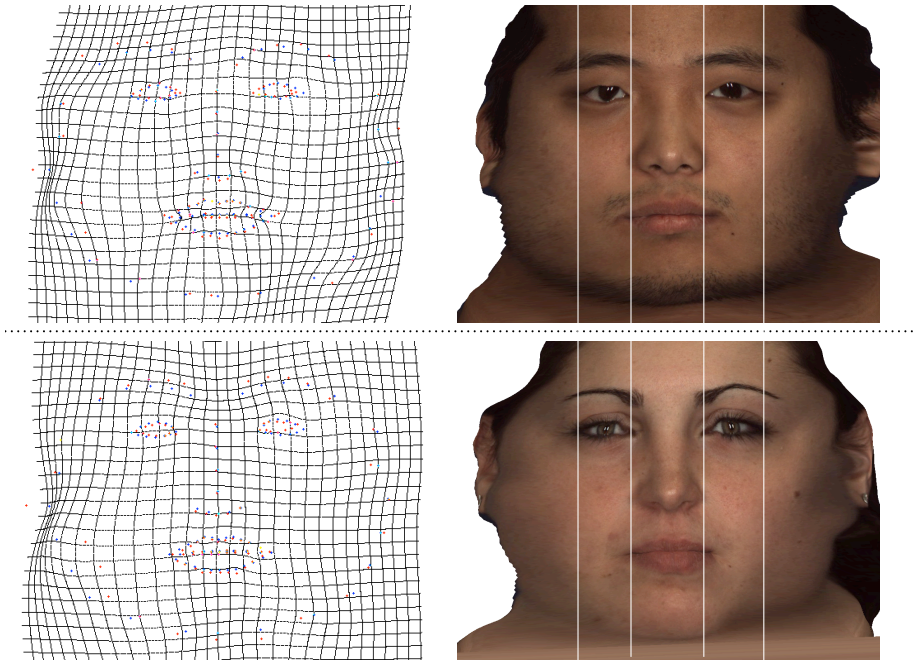

Figure 5.8: Right: two face textures viewed in cylindrical coordinates that have been placed in dense correspondence by a Thin-Plate Spline warp. Left: visualisations of the effect the TPS warp had on coordinate values for each face.

second image is procured, a *coordinate buffer*. Points between each coordinate undergo the same linear interpolation that usual colour values do, yielding a 3D analogue of the widely used depth buffer. This, in combination with the TPS Warped texture allow for the full construction of a face in correspondence (figure 5.9). This approach has the benefit of allowing for a primitive but effective form of mesh optimisation - to build a low polygon morphable model one only needs to sample from the coordinate buffer with less density. To capture the full shape detail, the sampling can be done at every pixel value. In either scenario, the full texture data can be retained.

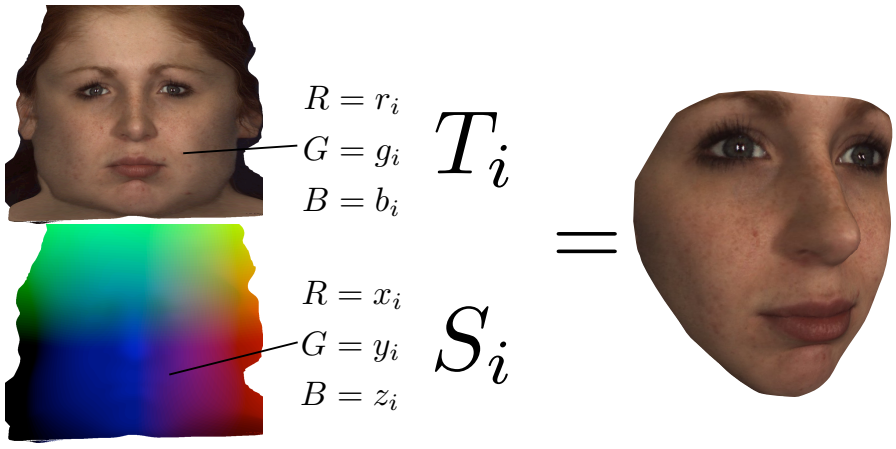

Figure 5.9: Top: TPS frame buffer output, bottom: coordinate buffer output. Sampling the same location on both buffers yields texture and (unwarped) shape information respectively, the latter of which can be triangulated to construct a version of the face in dense correspondence (right).

Chapter 6

Extracting Eigenfaces: Principal Component Analysis

Proceeding chapters have led to the production of a set of faces in dense correspondence. Now a statistical framework is developed for the processing of these meshes and textures into a morphable model. Although techniques explored are generally well understood, an attempt is made to explain the concepts in a manner that focusses on maintaining a geometric understanding throughout. Bra-ket notation ubiquitous in the field of Quantum Mechanics is used as it is an ideal fit for exploring concepts based on linear vector spaces. For a primer on Bra-ket notation, see the appendix.

6.1 Motivation

The purpose of a 3DMM is to construct novel plausible faces that were not in the original training set. Naturally, this lends itself to the concept of treating the faces as a linear statistical model, where a novel face is constructed from taking a linear superposition of the existing faces. Perhaps the most natural superposition of faces is the mean face, $\bar{\mathbf{F}} = \mathbf{F}(\bar{\mathbf{S}}, \bar{\mathbf{T}})$, where

$$\bar{\mathbf{S}} = \frac{1}{k}\sum_{i=1}^{k}\mathbf{S}_i \qquad \bar{\mathbf{T}} = \frac{1}{k}\sum_{i=1}^{k}\mathbf{T}_i \tag{6.1}$$

Intuition would suggest that an arbitrary face can be constructed in a similar manner by generalising 6.1 into the form $\mathbf{F}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = (\mathbf{s}(\boldsymbol{\alpha}), \mathbf{t}(\boldsymbol{\beta}))$, where

$$\mathbf{S}(\boldsymbol{\alpha}) = \sum_{i=1}^{k}\alpha_i\mathbf{S}_i \qquad \mathbf{T}(\boldsymbol{\beta}) = \sum_{i=1}^{k}\beta_i\mathbf{T}_i \tag{6.2}$$

with $\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_k \end{bmatrix}^T$, $\boldsymbol{\beta} = \begin{bmatrix} \beta_1 & \beta_2 & \dots & \beta_k \end{bmatrix}^T \in \mathbb{R}^k$ and the constraint that

$$\sum_{i=1}^{k} \alpha_i = \sum_{i=1}^{k} \beta_i = 1 \tag{6.3}$$

holds.

This naive approach can be used to build sensible faces, however it necessitates that any new face has to be explicitly described in terms of a mixture of the arbitrary training faces used to build the model. A face is built up out of a mixture of features, some common to all (iris size and shape) others more distinctive (the presence of a goatee beard). This spectrum of features, from common to rare, is present in all faces in the training set, which makes building novel faces with specific attributes tricky. Ideally, the faces in the model would be ordered in terms of commonality, the first containing only the most statistically likely features of a face, the last containing the attributes most rarely encountered. Keeping these features independent greatly simplifies the process of constructing novel faces. Principal Component Analysis is a mathematical technique for extracting these independent *eigenfaces* from the arbitrary set of faces used to build the model.

6.2 The representation of a face

Further analysis is simplified by adopting a different approach to the representation of a face. A face is still described by $\mathbf{S}$ and $\mathbf{T}$, however, as these two objects are processed in exactly the same manner for the remainder of this chapter, a generalisation is made where both shape and texture are simply considered two examples of facial *attributes*. An n dimensional attribute is defined as a vector of *parameter values*

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \tag{6.4}$$

To give a concrete example, the n dimensional texture attribute vector is constructed from RGB pixel parameter values, in a manner such as

$$\boldsymbol{x}_{texture} = \begin{bmatrix} r_1 & g_1 & b_1 & r_2 & g_2 & b_2 & \dots & r_{n_t} & g_{b_t} & b_{n_t} \end{bmatrix}^T \tag{6.5}$$

where $n = 3 \times n_t$. The choice of how to vectorize the shape and texture matrices is arbitrary, so long as it is performed consistently across the set of training faces (so as to maintain dense

correspondence). The reader should bare in mind that in practice the following process is done for both shape and texture independently, and a complete face is still described by the composition of the two attributes.

6.3 Faces as vectors in a shape space

We define an n dimensional Euclidean linear vector space $\mathbb{R}^n$ with a set of n basis vectors $\mathbf{X} = \{\ |x_1\rangle, |x_2\rangle, \ldots |x_n\rangle\ \}$, where, using Dirac's Bra-ket notation [10]

$$|x_1\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \qquad |x_2\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \qquad \ldots \qquad |x_n\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \tag{6.6}$$

A face attribute[1] can then be represented as

$$|F\rangle = x_1 |x_1\rangle + x_2 |x_2\rangle + \ldots + x_k |x_k\rangle = \sum_{i=1}^{k} x_i |x_i\rangle = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{pmatrix} \tag{6.7}$$

with the constraint that each face vector is of unit length

$$\langle F|F\rangle = \begin{pmatrix} x_1 & x_2 & \ldots & x_k \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{pmatrix} = \sum_{i=1}^{k} (x_i)^2 = 1 \tag{6.8}$$

so the set of all possible face objects is a hypersphere of unit radius around the origin. For a discussion of why attributes can be normalized without loss of generality, see section 4.1.

This vector space is a mathematical desription of the shape space defined earlier. Of course, many of these possible objects will be pathological, and in general the space of sensible faces will be a small subset of the hypersphere. The aim is to quantify how to explore this space. As a starting point, we postulate that a set of k sample faces, $\mathbf{F} = \{\ |F_1\rangle, |F_2\rangle, \ldots, |F_k\rangle\ \}$ will provide a (not

[1] Each n dimensional attribute has it's own n dimensional space. For our purposes, this means two vector spaces of differing dimensions; one for shape, the other for texture.

necessarily orthogonal) basis for the sensible face space. This would mean that a sensible novel face $|F_{nov}\rangle$ can be expressed in the basis as

$$|F_{nov}\rangle = c_1 |F_1\rangle + c_2 |F_2\rangle + \ldots + c_k |F_k\rangle \tag{6.9}$$

which is precisely what was proposed in equation 6.2. Although this basis can be used, an orthonormal basis $\mathbf{P}$ defined by

$$\langle P_i | P_j \rangle = \delta_{ij} \tag{6.10}$$

where δ_{ij} is the Kronecker delta, would be a more natural choice. A method for generating an orthonormal basis from a non-orthonormal basis is Principal Component Analysis.

6.4 Principal Component Analysis

6.4.1 Linearity of PCA and the tangent space

We first note that the basis can only span a linear vector space, not a curved one as we have over a region of a hypersphere. More properly then, we are seeking a basis that spans a *tangent* to the space of faces, and naturally we choose this tangent to be coincident on the hypersphere with the location of the mean face, given by

$$\exp(\mathbf{F}) \coloneqq |F_m\rangle = \frac{1}{k} \sum_{i=1}^{k} |F_i\rangle \tag{6.11}$$

All faces need now to be projected into the tangent space; a straightforward case of changing the magnitude of each vector. Define the tangent projection of a face as

$$|F_i^{tang}\rangle = \gamma_i |F_i\rangle \tag{6.12}$$

where $\gamma_i \in \mathbb{R}$. In the tangent space it must be true that

$$\begin{aligned} \langle F_m | F_i^{tang} \rangle &= 1 \\ \gamma_i \langle F_m | F_i \rangle &= 1 \end{aligned} \tag{6.13}$$

which reveals the appropriate scaling factor as

$$\gamma_i = \frac{1}{\langle F_m | F_i \rangle} \tag{6.14}$$

Figure 6.1 gives a geometrical view of the relationship between the attribute space, the tangent space, and the mean face in a low dimensional example. From now on, each face referred to has been projected to the tangent space, yet to keep notation compact, the superscript *tang* is dropped.

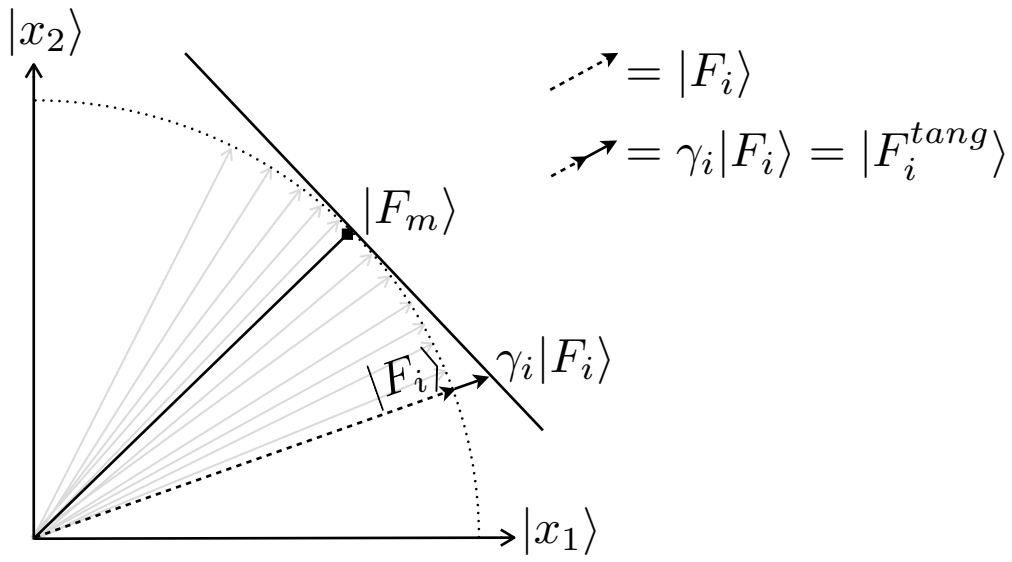

Figure 6.1: A geometrical view of the attribute and tangent space for the case of $n = 2$. The hypersphere in this space is a circle, the tangent space simply being the tangent to the circle evaluated at the mean face. Projecting all faces (grey arrows) into the tangent space is a simple scaling operation, as shown for the highlighted i'th face.

6.4.2 Deriving the Principal Component Basis

Define a new basis $\mathbf{P} = \{ \mid P_1\rangle, \mid P_2\rangle, \ldots, \mid P_{k-1}\rangle \}$ where

$$|P_i\rangle = \sum_{j=1}^{n} p_i^j \, |x_j\rangle \tag{6.15}$$

and $p_i^j \in \mathbb{R}$. Enforce that this basis is both orthonormal

$$\langle P_i | P_j \rangle = \delta_{ij} \tag{6.16}$$

and spans the tangent space. As such, each face can be expressed in terms of $\mathbf{P}$, as

$$|F_i\rangle = |F_m\rangle + \sum_{j=1}^{k-1} y_i^j \, |P_j\rangle \tag{6.17}$$

where $y_i^j \in \mathbb{R}$. We also note that this term has a geometrical meaning

$$\langle F_i - F_m | P_t \rangle = \sum_{j=1}^{k-1} y_i^j \langle P_j | P_t \rangle \tag{6.18}$$

$$\langle F_i - F_m | P_t \rangle = \sum_{j=1}^{k-1} y_i^j \delta_{jt} \tag{6.19}$$

$$\langle F_i - F_m | P_t \rangle = y_i^t \tag{6.20}$$

that is, y_i^j is the inner product between the i'th mean subtracted face and the j'th principal component.

Construct $\mathbf{Y}$, the matrix of y_i^j terms,

$$\mathbf{Y} = \begin{pmatrix} y_1^1 & y_1^2 & \dots & y_1^{k-1} \\ y_2^1 & y_2^2 & \dots & y_2^{k-1} \\ \vdots & \vdots & \ddots & \vdots \\ y_k^1 & y_1^2 & \dots & y_k^{k-1} \end{pmatrix} \tag{6.21}$$

noting that the i'th *column* of $\mathbf{Y}$ dictates the weightings of the i'th principal component for all k faces.

By defining the mean-centred face matrix $\mathbf{F_{mc}}$

$$\mathbf{F_{mc}} = \begin{pmatrix} \uparrow & \uparrow & \dots & \uparrow \\ |F_1 - F_m\rangle & |F_2 - F_m\rangle & \dots & |F_k - F_m\rangle \\ \downarrow & \downarrow & \dots & \downarrow \end{pmatrix} \tag{6.22}$$

equation 6.17 can be summarised over all principal components and all faces as

$$\mathbf{F_{mc}} = \mathbf{P}\mathbf{Y}^T \tag{6.23}$$

which can be rearranged (remembering that $\mathbf{P}^{-1} \equiv \mathbf{P}^T$ as $\mathbf{P}$ is an orthogonal matrix)

$$\mathbf{P}^T\mathbf{F_{mc}} = \mathbf{P}^T\mathbf{P}\mathbf{Y}^T$$

$$\mathbf{P}^T\mathbf{F_{mc}} = \mathbf{Y}^T \tag{6.24}$$

$$\tag{6.25}$$

Enforce that (mean centred) faces represented in the Principal Component basis are uncorre-

lated. Mathematically, this is

$$\text{cov}(\mathbf{Y}) \coloneqq \mathbf{\Lambda} = \text{diag}\begin{pmatrix}\sigma_1^2 & \sigma_2^2 & \dots & \sigma_k^2\end{pmatrix} \tag{6.26}$$
$$\mathbf{\Lambda} = \frac{1}{k-1}\mathbf{Y}^T\mathbf{Y}$$

as the columns of $\mathbf{Y}$ are already mean centred, being defined about $|F_m\rangle$. A final condition states that

$$\sigma_i > \sigma_j, \quad \forall i < j \tag{6.27}$$

that is, the variances along the principal component axes are ordered, with the highest variance along $|P_1\rangle$, also known as the first principal component.

Starting with 6.26

$$\mathbf{\Lambda} = \frac{1}{k-1}\mathbf{Y}^T\mathbf{Y} \tag{6.28}$$
$$\mathbf{\Lambda} = \frac{1}{k-1}(\mathbf{P}^T\mathbf{F}_{mc})(\mathbf{P}^T\mathbf{F}_{mc})^T \tag{6.29}$$
$$\mathbf{\Lambda} = \frac{1}{k-1}\mathbf{P}^T\mathbf{F}_{mc}\mathbf{F}_{mc}^T\mathbf{P} \tag{6.30}$$
$$\mathbf{\Lambda} = \mathbf{P}^T(\frac{1}{k-1}\mathbf{F}_{mc}\mathbf{F}_{mc}^T)\mathbf{P} \tag{6.31}$$
$$\mathbf{\Lambda} = \mathbf{P}^T\text{cov}(\mathbf{F}_{mc}^T)\mathbf{P} \tag{6.32}$$

and rearranging for $\text{cov}(\mathbf{F}_{mc}^T)$

$$\text{cov}(\mathbf{F}_{mc}^T) = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^T \tag{6.33}$$
$$\text{cov}(\mathbf{F}_{mc}^T) = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{-1} \tag{6.34}$$

which is the standard form of the eigendecomposition of $\text{cov}(\mathbf{F}_{mc}^T)$. Thus, by direct comparison, the eigenvectors of $\text{cov}(\mathbf{F}_{mc}^T)$ define an orthonormal basis (known as the Principal Component basis) in which the variance is completely maximised to lie along the basis, with no covariance between different basis vectors. The corresponding eigenvalue captures the variance captured along a given basis vector. All that is required is a sorting of the variances from large to small to define the Principal Component basis in descending order of significance. An arbitrary face can now be expressed in the principal component basis as

$$|F_{nov}(\boldsymbol{\alpha})\rangle = |F_{mean}\rangle + \sum_{i=1}^{k-1}\alpha_i\,|P_i\rangle \tag{6.35}$$

the remaining question being what constraints should be placed on the $\boldsymbol{\alpha} = \{ \alpha_1, \alpha_2, \alpha_{k-1} \} | \alpha_i \in \mathbb{R}$ coefficients to yield sensible novel faces.

6.5 Limiting coefficients - the sensible face subspace

Surprisingly, there is no concrete interpretation in the literature for what constraints should be placed on the shape and texture principal component coefficients. Blanz and Vetter [5] argued that the fact that Principal Component Analysis can be applied successfully to the mean centred faces at all infers that the distribution of faces in the hyperplane is a multivariate Gaussian distribution, specifically one with zero mean and covariances given by $\sigma_1^2, \sigma_2^2, \ldots, \sigma_{k-1}^2$. Under this interpretation, the probability of a given attribute vector existing is proportional to the standard Gaussian multivariate probability distribution

$$p(|F_{nov}(\boldsymbol{\alpha})\rangle) \propto e^{-\frac{1}{2}\sum_i \frac{\alpha_i^2}{\sigma_i^2}} \tag{6.36}$$

This interpretation, however suggests that that mean face is statistically the most likely face to be found, something which is trivial to experimentally verify as false. The inability to reconcile these two theorems is known as the *The Face-Space Typicality Paradox* [7]. Patel [3] has contributed to the discussion by instead suggesting that the important constraint is actually on the Mahalanobis length of the mean subtracted face vector

$$| \, |F_{nov} - F_m\rangle \, |_M = \sum_{i=1}^{k-1} \frac{\alpha_i^2}{\sigma_i^2} \tag{6.37}$$

which he shows follows a Chi-squared distribution. The expected value of such a distribution is actually $k-1$, and the probability of a given face being the mean (equation 6.37 being zero) is vanishingly small. This however does not provide guidance on the individual values of the coefficients for each principal component.

As an extension to this thesis, it would be interesting to further develop the statistical framework governing the space of sensible faces. For present applications, however, we stick to the Blanz view of the matter given in equation 6.36, as this has immediate utility in providing a sensible scope limit on significance the i'th component should have in building a sensible face. In particular, we note that weightings of the order σ_i would nicely exemplify the features of the i'th component on the model whilst providing a sensible result.

Chapter 7

Results

In order to assess performance of the techniques presented in this thesis, four separate morphable models are constructed using our pipeline. All input data is taken from the BU-4DFE [17] database. In this dataset, each subject performs an emotion over a short period of time - the faces used in this thesis are snapshots or frames from this temporal emotion data. The first three models, *happy*, *sad*, and *neutral*, are each built from a set of approximately 70 subjects all displaying the same emotion. The fourth model, *mixed*, is built from all 203 input faces, with an approximately even split of happy, sad, and neutral faces present.

For each of these four models, we examine the model's intrinsic properties - namely the mean shape and texture, along with the three most significant principal components. We then project an out of sample face onto the principal components, to gauge in a controlled way the expressiveness of the model. Finally, each model is utilized in an image matching algorithm to test it's performance in a real world application. The widely used Basel Face Model [12] is also used in the same matching algorithm as a point of comparison.

7.1 Mean and most significant principal components

Figures 7.1-7.4 show for each of the four models the mean and first three shape and texture principal components. To visualize the i'th principal component, faces of the form

$$|F\rangle = |F_m\rangle \pm 3\sigma_i |P_i\rangle \tag{7.1}$$

are constructed for texture and shape respectfully, where σ_i is the relevant standard deviation given by equation 6.26.

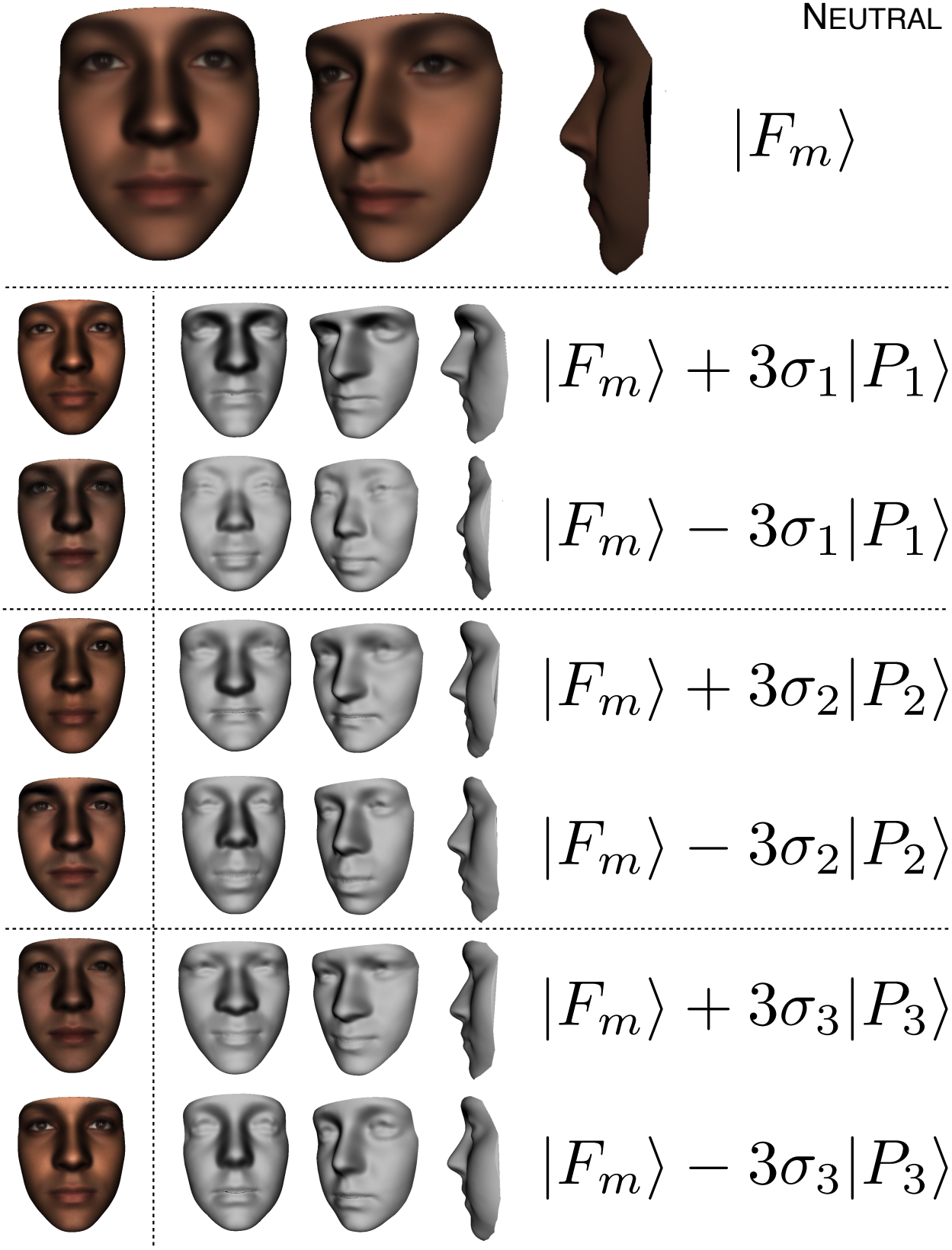

Figure 7.1: Mean and principal components of the neutral model. Top row: three views of the total mean face, constructed from the mean shape and the mean texture. Subsequent three pairs of rows: The first three shape and texture principal components, in each case added and subtracted from the mean as expressed on the far right. For each expression, a frontal rendering of the texture attribute mapped onto the mean shape is given (left of the dotted line) along with a texture-free rendering of the shape attribute from a frontal, posed, and side view (right of dotted line).

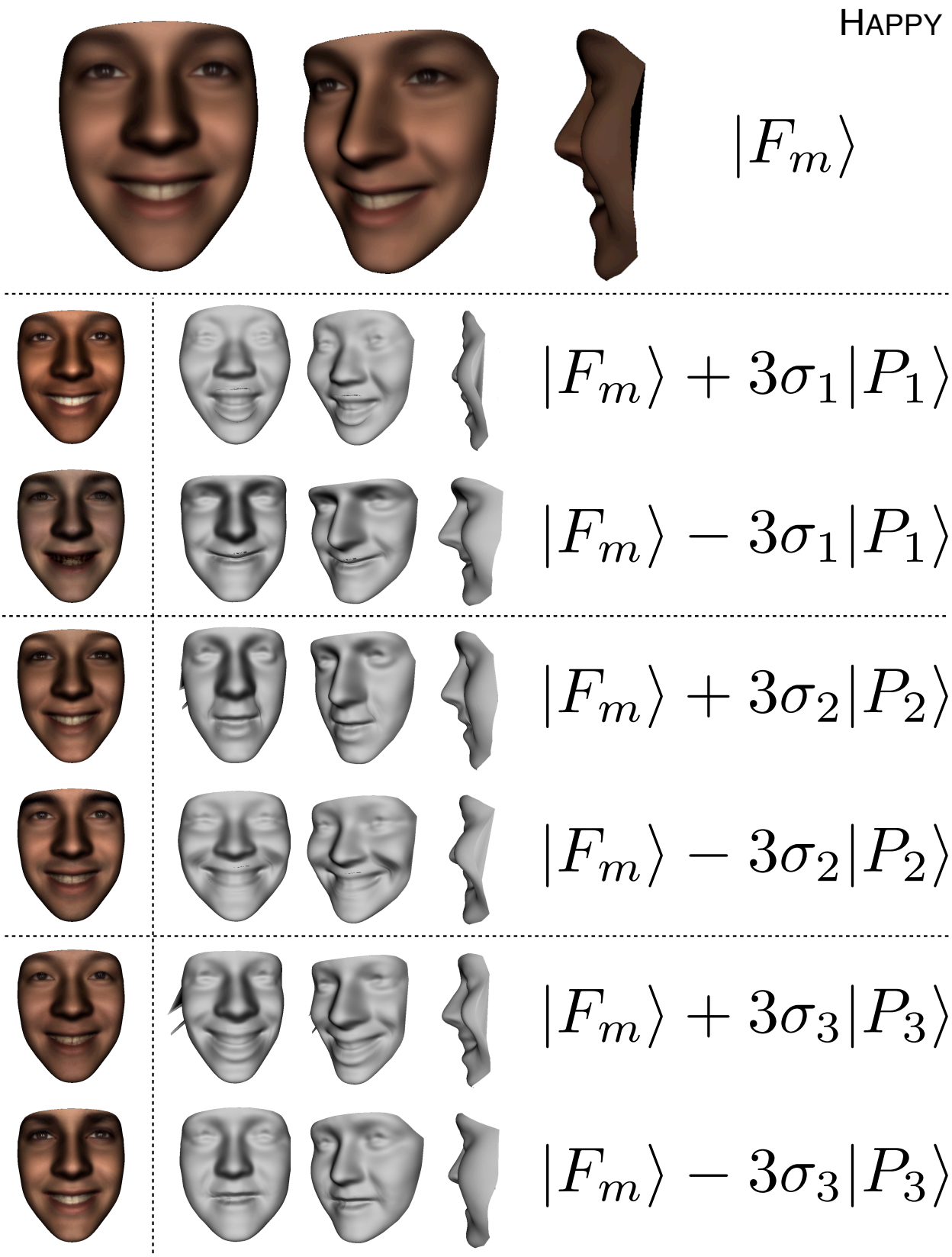

Figure 7.2: As figure 7.1, but for the happy model. Note the strong similarity in the identity of the shape and texture principal components to their counterparts from the neutral model, as exemplified in figure 7.7. Also note the erroneous strong features present in the second and third shape principal components near the cheek region.

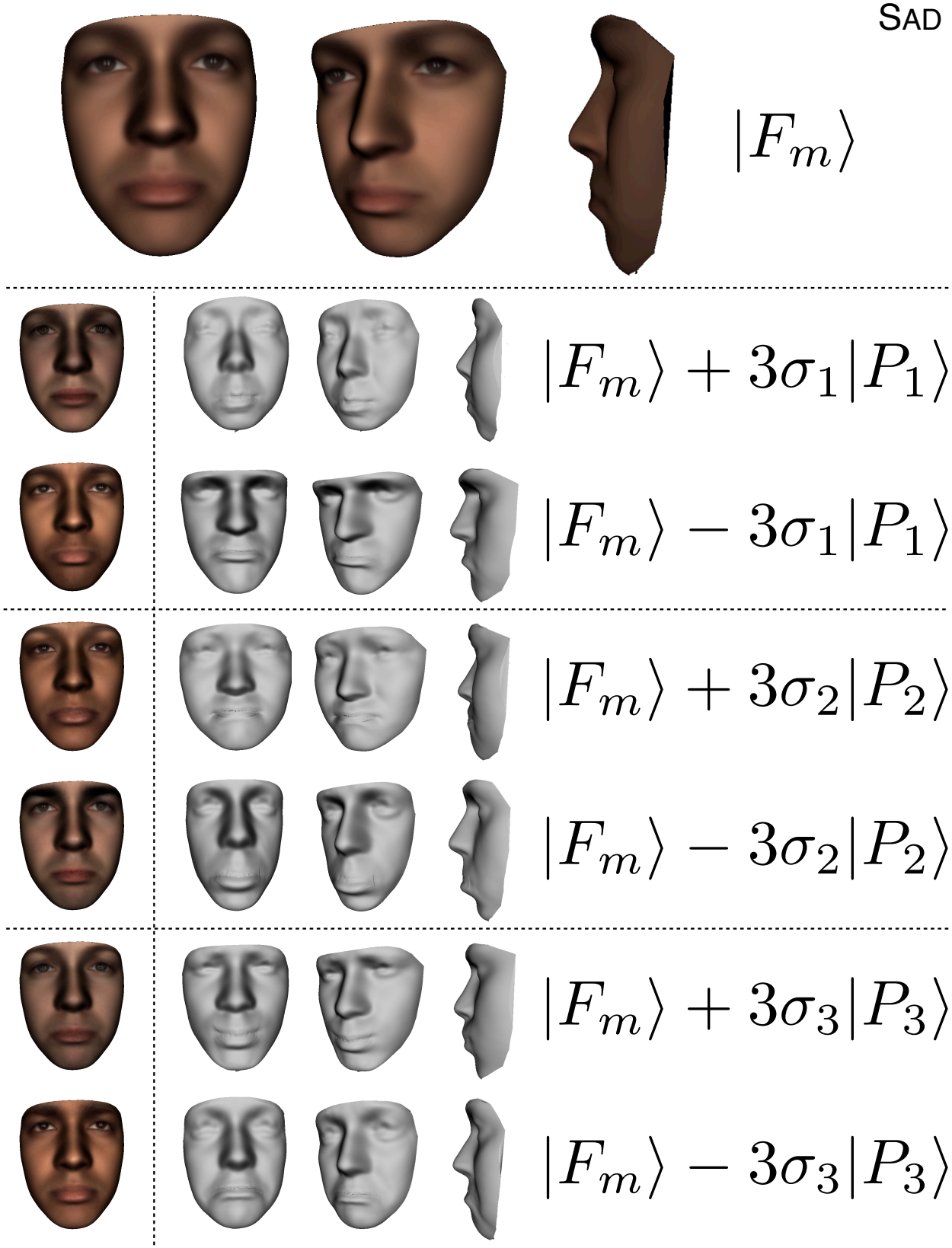

Figure 7.3: As figure 7.1, but for the sad model. Again each component's identity is consistent with previous models, with only the emotion changing. Interestingly, the addition of the third shape principal component results in a week smile, suggesting a duality between happy and sad emotions.

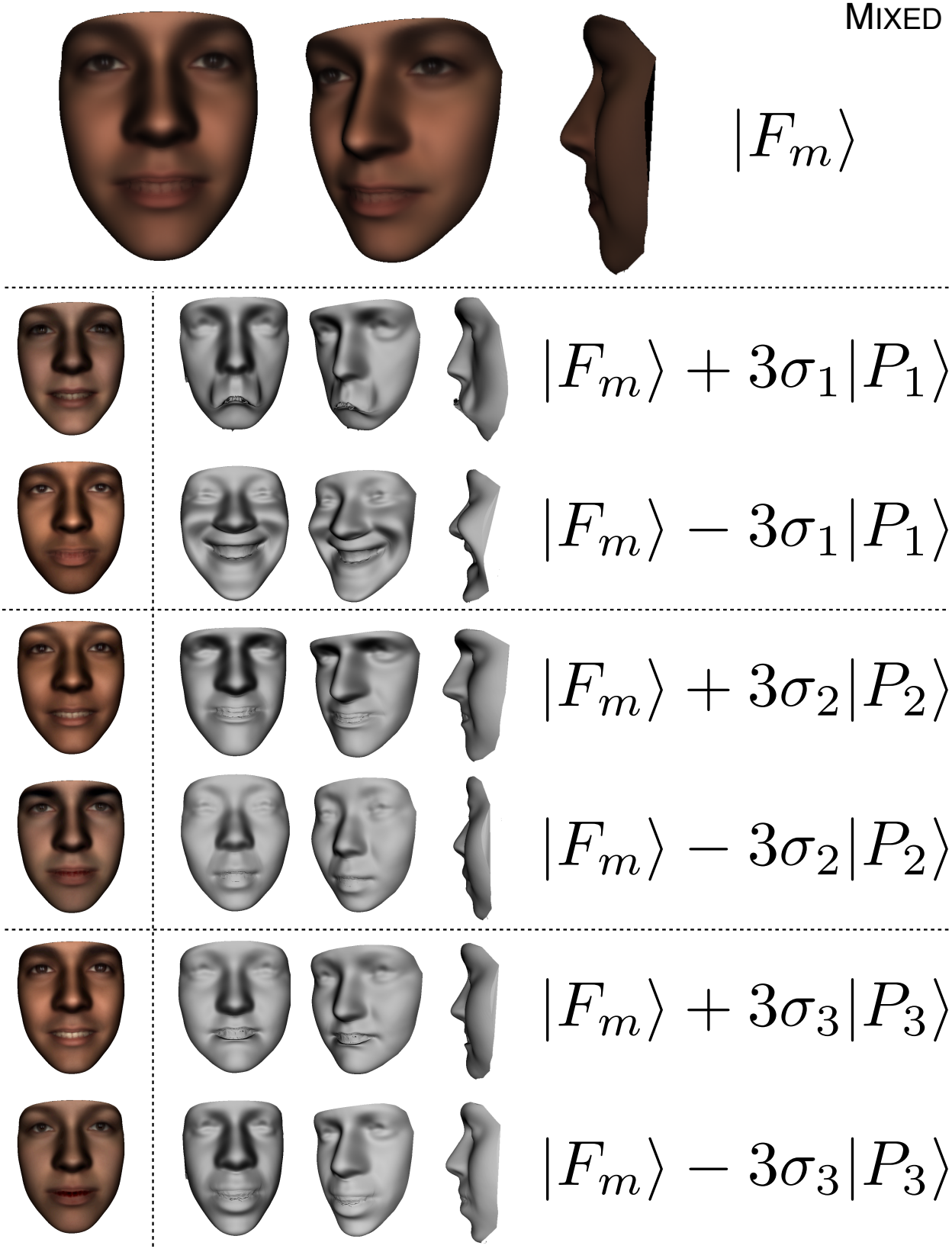

Figure 7.4: As figure 7.1, but for the model constructed from a mixture of all neutral, happy, and sad faces. The mean face appears close to neutral, with principal component one seemingly a blend of happiness and sadness depending on addition or subtraction. The second and third components do not express either emotion, yet do bear a strong resemblance to the identities seen in the first two principal components from the previous discrete models.

7.1.1 Discussion

Broadly, it is encouraging that the first three eigenvectors for all four models display with clarity an identity and emotion, however some noise artefacts can be seen on the second and third shape components of the happy model around the cheek area. This is likely explained by the limited resolution of model data used. Mesh data in the BU-4DFE database is only pseudo-3D (constructed from a depth reading from a single camera facing the front of the subject) so surfaces near-parallel to the capture device have extremely low shape and texture information density. Indeed, areas like the nostrils are entirely devoid of model detail due to this shortcoming. This is further compromised by the landmarking approach taken, as finding the nearest mesh coordinate from points along the flanks of the face are highly susceptible to large variations in position.

Another unphysical artefact in the data is the presence of some black regions at the edge of the mesh, most easily seen at the cheek boundary in the side view. This is a result of the way the TPS warping is performed - in the current pipeline only coordinates within the landmarked perimeter of the face are warped. The cut-out which is sampled from the TPS image to create the faces in dense correspondence is a polygon bound by the perimeter landmarks. This straightforward linear crop means that, in regions of strong warping, points in-between landmarks on the edge can have no texture value. This effect was much stronger than anticipated on the cheek region due to the heavy warping caused by the large variation in cheek landmarks discussed above. Importantly, it is a minor flaw due to the current specific implementation, not the technique of TPS warping itself, and could be easily fixed in the future.

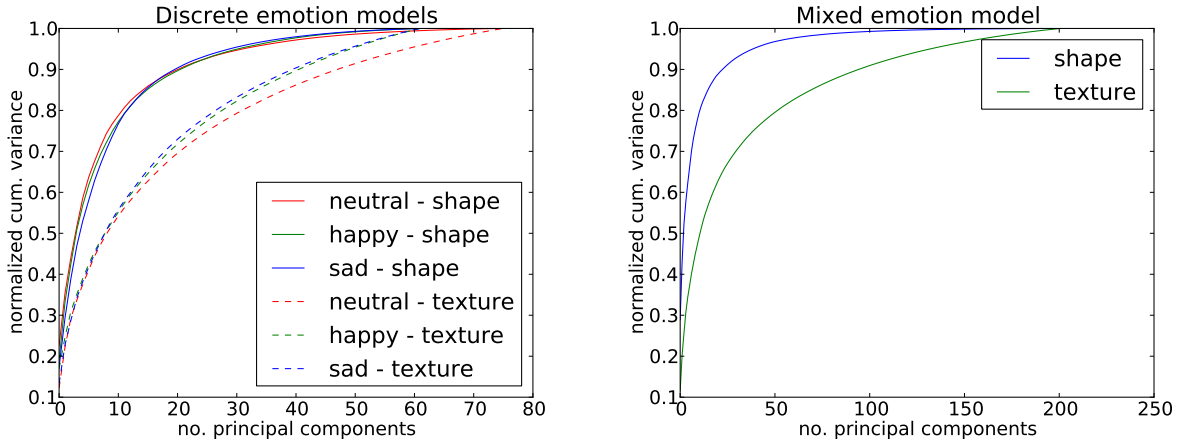

Figure 7.5: Cumulative variances of the shape and texture principal components for the four models built using our pipeline. Each cumulative variance is normalized to allow comparison between shape and texture for each model.

A useful measure of the quality of the principal components found is by looking at the cumulative

variance accounted for by a subset of the components. Figure 7.5 shows this for both shape and texture components across all models. The shape and smoothness of the curves is in line with exceptions, as can be seen by direct comparison with the equivalent graph calculated for the Basel Face Model (figure 7.6).

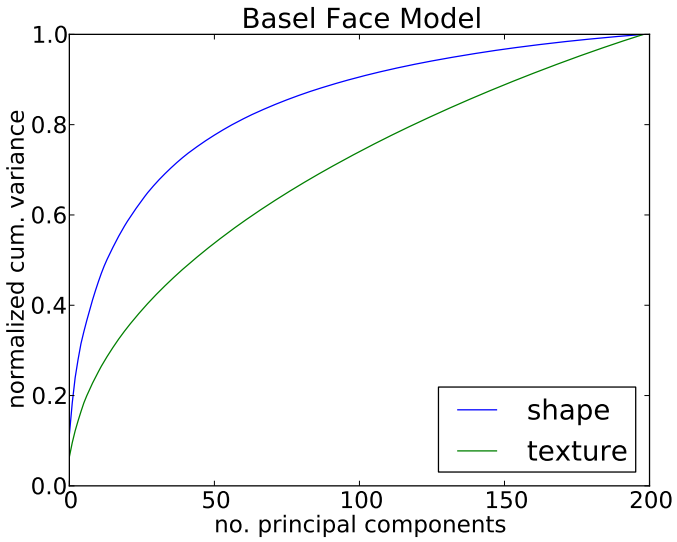

Figure 7.6: Normalized cumulative variance for the shape and texture principal components of the Basel Face Model.

The role of emotion and identity in a mixed model's principal components

The most novel aspect of this thesis is the native inclusion of emotive faces in 3D Morphable Model construction. By native, we mean without special consideration - happy, sad, and neutral faces are processed in an identical manor in the model's construction. 3DMM's capable of expressing emotion have been developed before, namely by Amberg in his PhD thesis [2], but the approach is to build a model from purely neutral faces first, before mapping emotions to subjects by capturing secondary emotive faces which are physically landmarked with a special paint pattern. Naturally, this means that the emotive textures are unusable, but the physical markings allow for a consistent automated alignment with the corresponding neutral face.

The open question is how identity and emotion are represented in the principal components of a natively mixed-emotion morphable model.

As stated previously, there is a consistent set of identity eigenfaces which are present across all the three discrete emotion models. What is somewhat surprising though is the presence of same identities in the mixed model principal components, all of which display the same emotion seen on the mean face (something close to neutral).

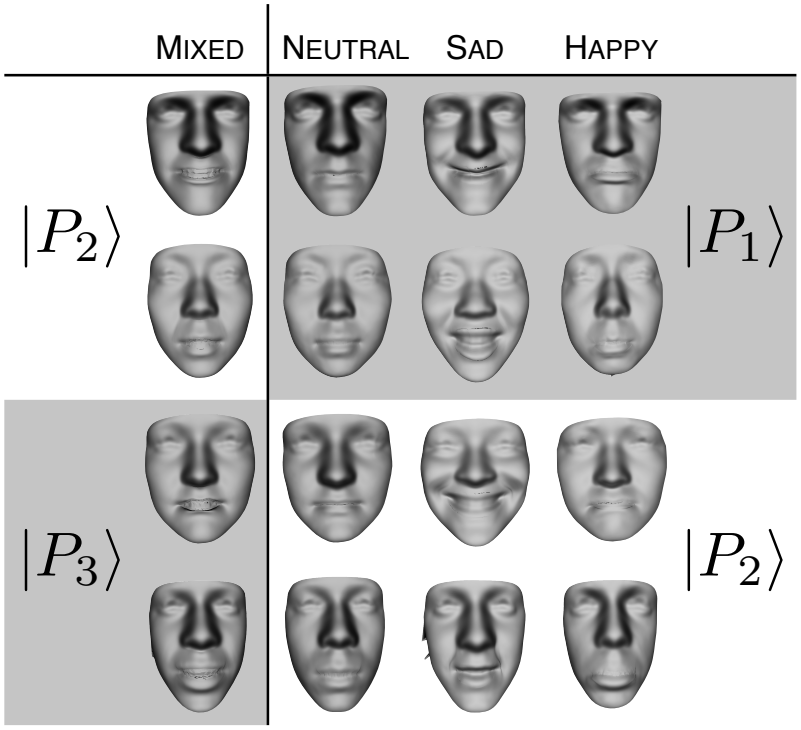

Figure 7.7: Direct comparison of the second and third shape components from the combined model against the first and second from the separate emotion models. The consistency of identity across the rows is clear. The unchanging emotion on the faces from the mixed model is somewhat suggestive that the principal components of a combined emotion model naturally separate into emotion and identity.

The consistency of these identities present in the models is demonstrated in figure 7.7, which presents the identities seen across the different model's principal components side by side. Note that whether an identity appears as a 'plus' or a 'minus' of a certain principal component and the mean is meaningless, as no care is taken to ensure a consistent handedness in the calculation of the eigenvectors.

The fact that the same key identities occur across all models is not surprising, as there is a large overlap in the identities of the subjects used for each model. (To be clear, the discrete emotion models are not constructed from the exact same people, but many individuals appear in two or three of the models due to the limited scope of subjects that could be chosen from the BU-4DFE database. This naturally means that the mixed database, being a superset of the discrete models, contains repetitions of the same identities found in the other models).

What is of interest, is that the identities appear in the mixed database principal components *all showing the same* mean emotion. The fact that this mean emotion is close to neutral is not surprising given the number of neutral faces used in building the model is slightly higher than the number of happy and sad faces (77 neutral, 63 happy/sad). More important is that the key identities (consistently the first and second principal components in the discrete models) are

demoted in statistical significance, with the first component being a previously unseen 'emotion' mode, a face purely representing the happy and sad emotions.

Future work

Unfortunately, much of this is based on a subjective analysis of the models, but there is some evidence that a clean separation of emotion and identity might naturally occur in mixed emotion morphable models. Future work could explore this more fully by altering the balance of emotions present in the mixed model and seeing how it impacts the mean and principal components seen.

7.2 Projection onto out of sample face

In this experiment, each model is constructed as normal from the k relevant input faces up to the point of projecting the faces in dense correspondence into the tangent space, however only a random subset of $k-3$ faces are then included in the calculation of the principal components. Each of the three random (mean subtracted) out of sample faces are then *projected* onto the $k-4$ principal components, yielding a set of $k-4$ reconstruction coefficients, for shape and texture respectfully.

Motivation

Mathematically, for both shape and texture we have three out of three out of sample faces $|F_{out}\rangle$.

A reconstruction of this out of sample face can be formed as

$$|F^r_{out}\rangle = |F_m\rangle + \sum_{i=1}^{k-4} \alpha_i |P_i\rangle \tag{7.2}$$

where as before $|F_m\rangle$ is the mean face attribute vector, and $\alpha_i \in \mathbb{R}$. Ultimately, the principal components only span a subspace of the attribute space. Any face which does not lie in the subspace cannot be exactly reproduced by a pure mixture of the eigenfaces. Very commonly, we seek to minimise the deficit between the reproduction and the original. Bearing this in mind, consider $|error\rangle$, the vector difference between the reconstructed face and the original out of sample face

$$|error\rangle = |F^r_{out}\rangle - |F_{out}\rangle \tag{7.3}$$

$$|error\rangle = |F^r_{out} - F_m\rangle - |F_{out} - F_m\rangle \tag{7.4}$$

where the addition and subtraction of the mean face will be convenient shortly.

This error vector starts at the out of sample face and points into the hyperplane spanned by the eigenvectors. If we remind ourselves that each face vector can be expressed in it's attribute basis

$$|F_{out}\rangle = \sum_{i=1}^{n} x_i \, |x_i\rangle \tag{7.5}$$

$$|F^r_{out}\rangle = \sum_{i=1}^{n} x^r_i \, |x_i\rangle \tag{7.6}$$

then we seek the set of coefficients $\{ \alpha_1, \alpha_2, \ldots, \alpha_{k-4} \}$ which minimises the norm of the error vector

$$\min \langle error|error\rangle \tag{7.7}$$

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} (x_i - x^r_i) \langle x_i|x_j\rangle (x_j - x^r_j) \tag{7.8}$$

$$\min \sum_{i=1}^{n} (x_i - x^r_i)^2 \tag{7.9}$$

As a concrete example, for the shape attribute equation 7.9 would be a minimisation of the sum of least squares differences between all corresponding coordinate values.

From simple geometry, it can be stated that the vector of minimal magnitude between a point and a hyperplane is one which is orthogonal to the plane, or equivalently, a vector who's inner product with any vector in the hyperplane is zero

$$\langle P_i|error\rangle = \langle P_i|F_{out} - F_m\rangle - \langle P_i|F^r_{out} - F_m\rangle \tag{7.10}$$

$$= \langle P_i|F_{out} - F_m\rangle - \sum_{i=1}^{k-4} \alpha_j \langle P_i|P_j\rangle \tag{7.11}$$

$$= \langle P_i|F_{out} - F_m\rangle - \sum_{i=1}^{k-4} \alpha_j \delta_{ij} \tag{7.12}$$

$$= \langle P_i|F_{out} - F_m\rangle - \alpha_i \tag{7.13}$$

$$= 0 \tag{7.14}$$

from which we find

$$\langle P_i|F_{out} - F_m\rangle = \alpha_i \tag{7.15}$$

Thus the problem of finding the closest reconstruction of a out of sample face is simply a case of calculating the inner product or projection between the mean subtracted out of sample face and each principal component, and then using the resulting coefficients and equation 7.2 to construct

a face. As such, this projection test provides a good consistent benchmark of the reproductive performance of the model.

Results

Figures 7.8 to 7.11 show the result of an out of sample face being projected onto the principal components. To provide a quantitative measure of performance, the normalised error, given by

$$e = \frac{\langle error|error\rangle}{\langle F_{out}|F_{out}\rangle} \tag{7.16}$$

is provided for both shape and texture (e_s and e_t respectively).

As a baseline, the neutral model/neutral face (neutral-neural) face projection is give in figure 7.8. Subjectively, the model has done a good job of reproducing the out of sample face, with the general features being present as expected. The weakest areas of reproduction are around the eyes and mouth, where a small amount of noise is seen. Generally, high frequency features fair worst, with no reproduction of skin freckles or blemishes in the texture. This is in line with expectations for a statistical model based on an orthogonal basis, much as high frequency details are the hardest to reproduce in a Fourier Transform, so higher frequency features on out of sample vectors here require more and more principal components to recover faithfully.

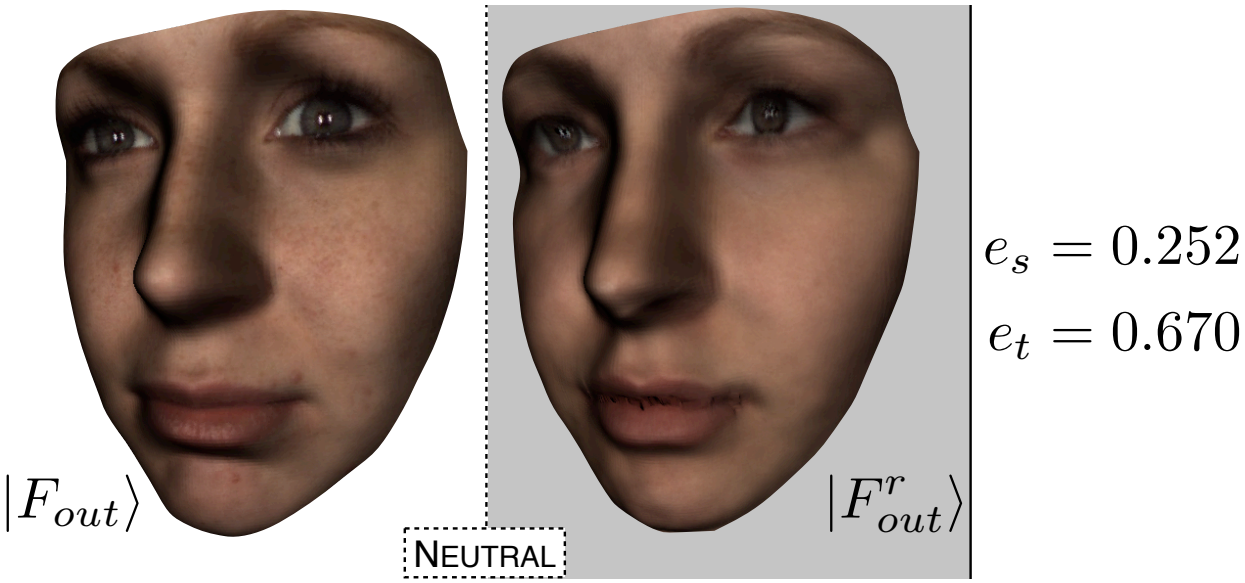

Figure 7.8: Left: an out of sample neutral face. Right: the result of rebuilding the out of sample face from the principal components of the neutral model, as by equation 7.2.

Figure 7.9 shows the sad-sad projection, and is of a similar quality to the neutral-neutral projection both subjectively, and in terms of the error metrics. Figure 7.10 is the result of the happy-happy projection. This has markedly higher error rates than the previous two projections,

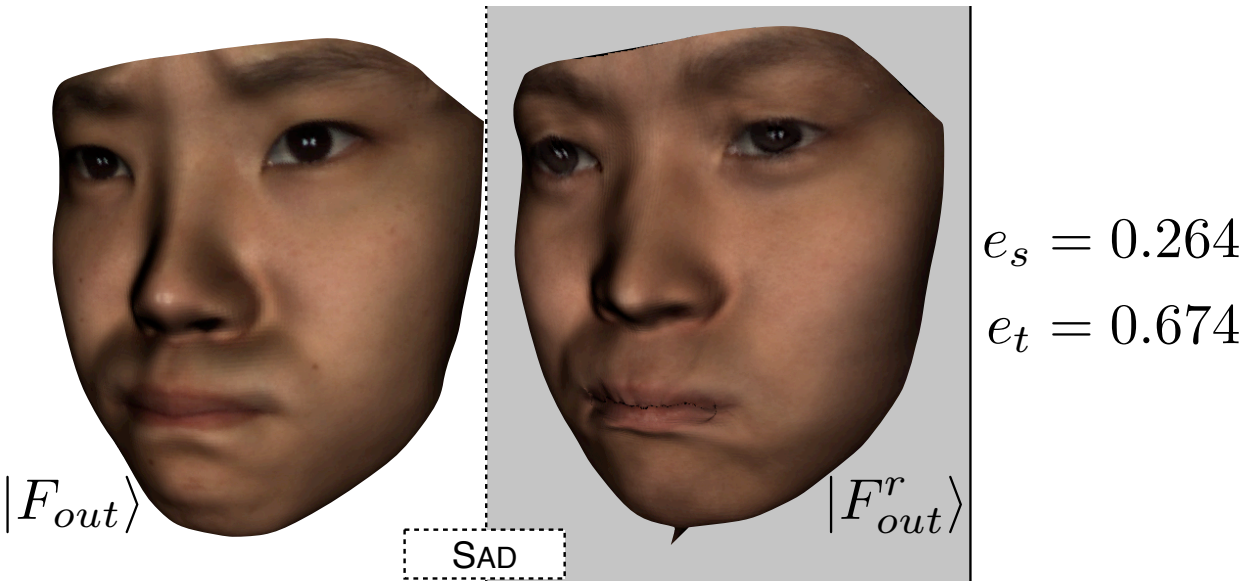

Figure 7.9: As figure 7.8, but for a sad out of sample face projected onto the sad model.

and certainly the exposed teeth seem to pose difficulties in reproduction. This is not so surprising, as unlike the rest of the face the teeth do not deform as a smile is shown, which is the base assumption behind why the TPS warp is successful in establishing dense correspondence. Potential improvements in this area will be discussed shortly.

Figure 7.10: As figure 7.8, but for a happy out of sample face projected onto the happy model.

Finally, figure 7.11 shows three projections from the mixed model. Across the board, error rates are lower than in the individual emotion cases. This is simply a by-product of the mixed model containing more principal components. The hyperplane spanned by this model is a superset of all the individual models hyperplanes, which can only act to reduce projection error.

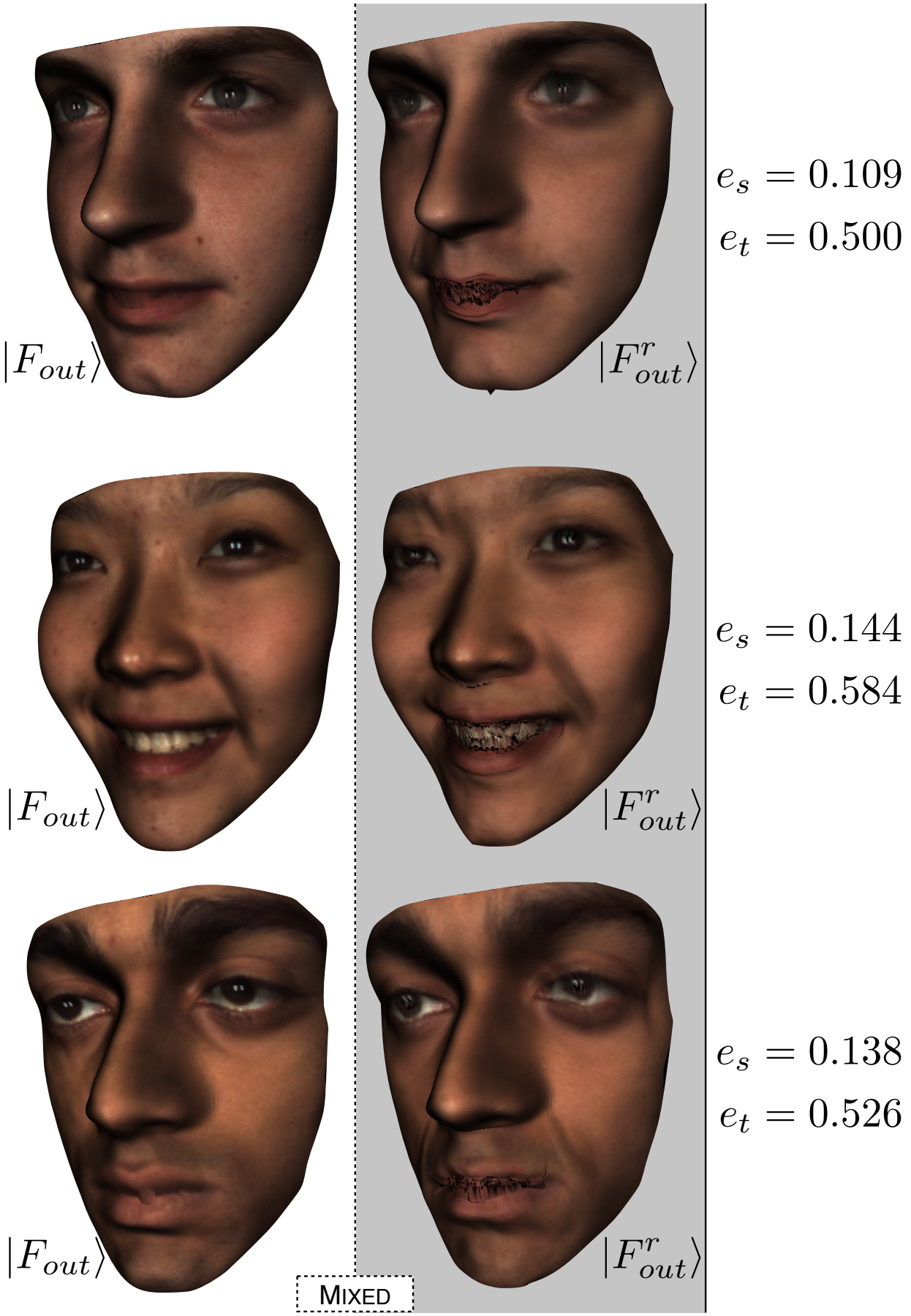

Figure 7.11: A range of faces projected onto the mixed model. Top: A neutral out of sample face projection. Middle: Happy face projection. Bottom: Sad face projection.

7.2.1 Image matching

The final experiment involves matching the mixed model through an implementation [13] of the Lucas Kanade algorithm [14].

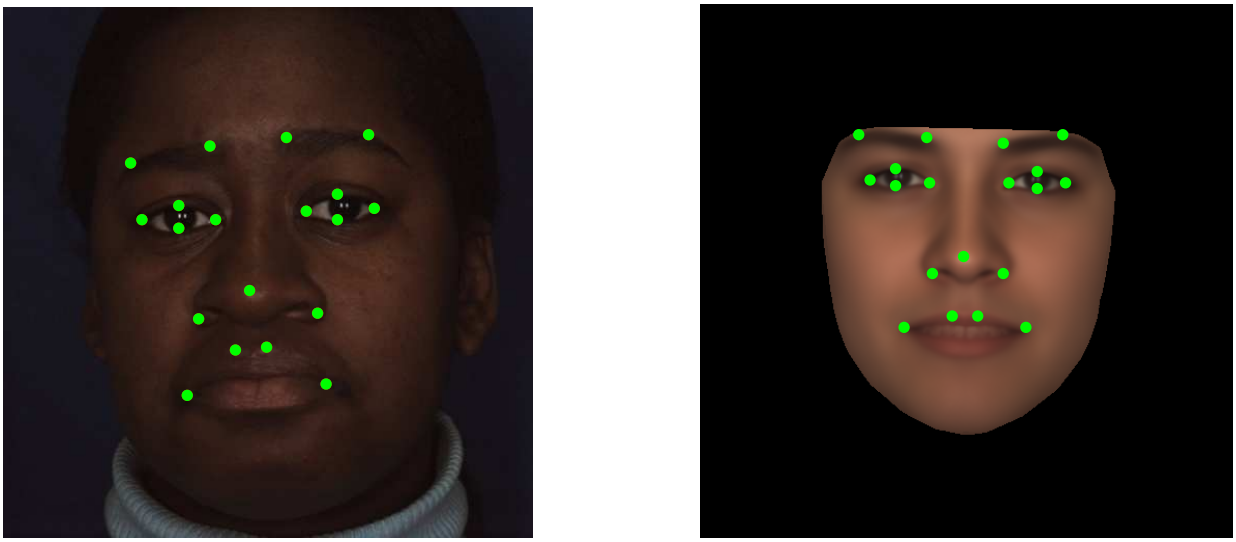

Figure 7.12: Example of the landmarking program used for the matching algorithm. Left: target image. Right: rendering of mean face from the 3DMM, the starting point of the matching algorithm. This figure shows 19 of the 20 landmarks marked, the final being situated on the bottom lip.

Firstly, 20 points are landmarked on both the model and the target image (figure 7.12). Then, a five stage fitting is performed. In each stage the algorithm builds a face from some mixture of the shape and texture mean and principal components, positions the face in the image, and (optionally) applies a lighting model, before using a set of cost functions to iteratively improve the generated face towards a good match. Each stage of the pipeline focuses on a different part of the process to try and ensure a stable solution.

In the first stage, the mean model face is Procrustes aligned as closely as possible to the target image by matching the positions of the relevant annotations. An example of the state of the algorithm after this first stage is given at the top of figure 7.13. The second stage is still solely about shape alignment, only now weightings of the first few shape principal components are allowed to vary in order to better align the face. The third stage calculates a lighting model, improving it iteratively until the mean face is matched as closely to the target image as possible. In the fourth stage the first few principal components of the shape and texture are now allowed to vary, and iteratively the algorithm starts to truly match the source image. In the fifth and final stage, the number of components used increases to try and capture the most detail possible from the image.

Figures 7.16 to 7.17 show the matching results against a range of faces.

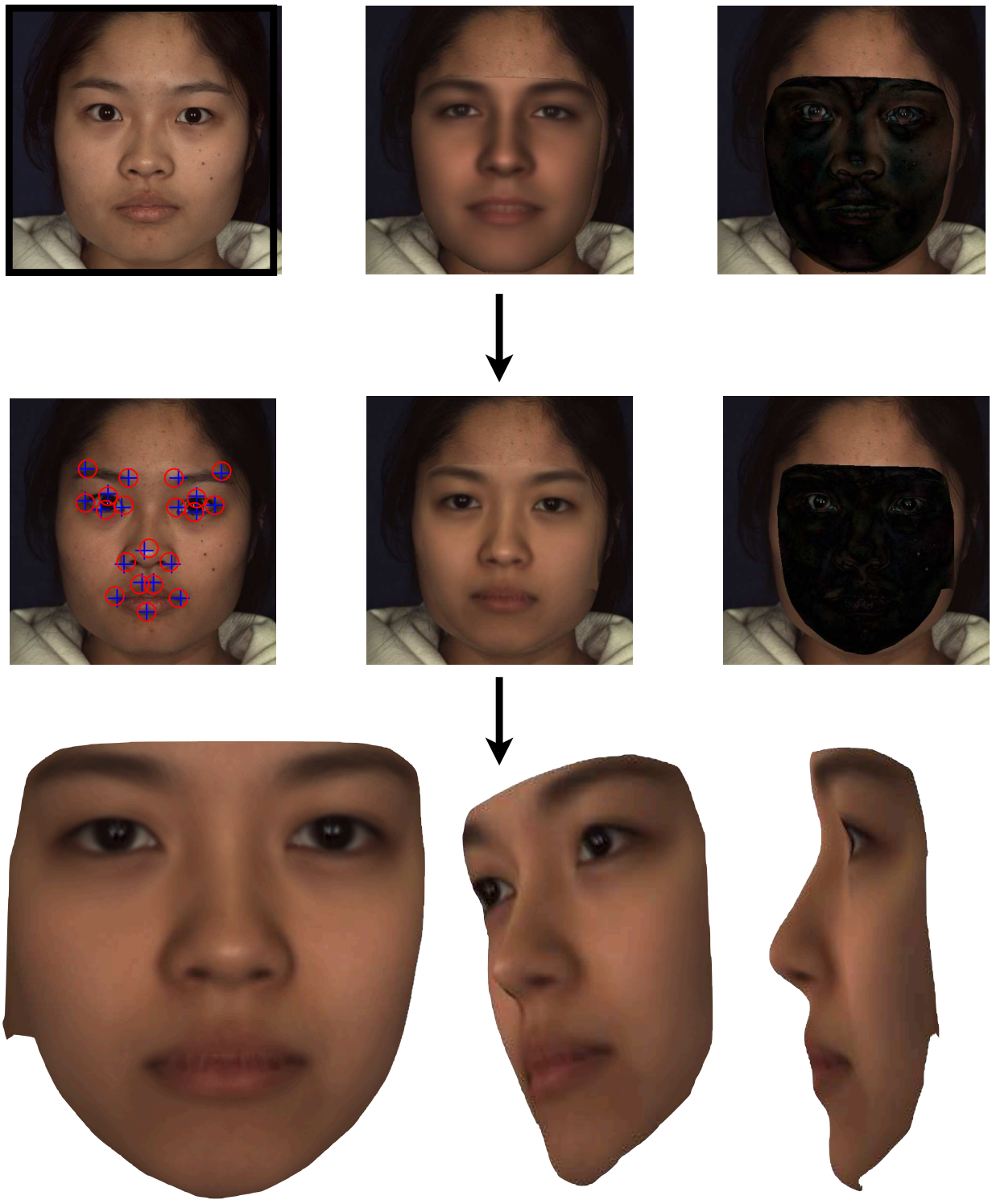

Figure 7.13: Top row: left: the input face matched against. Middle: the state of the algorithm after the first stage of alignment. Right: The per-pixel error between the model and the image. A black pixel means the respective model pixel color value perfectly matches the original. Middle row: the state of the algorithm at the end of the matching process. Bottom row: renderings of the output of the algorithm.

Figure 7.14: As figure 7.13, but for the Basel Face Model matched against the same image, and under the same algorithm settings. The middle row once again depicts the state of the algorithm post matching. While it is certainly the case that the large region of the face modelled leads to a more challenging matching environment, it is encouraging how well our model performs in comparison.

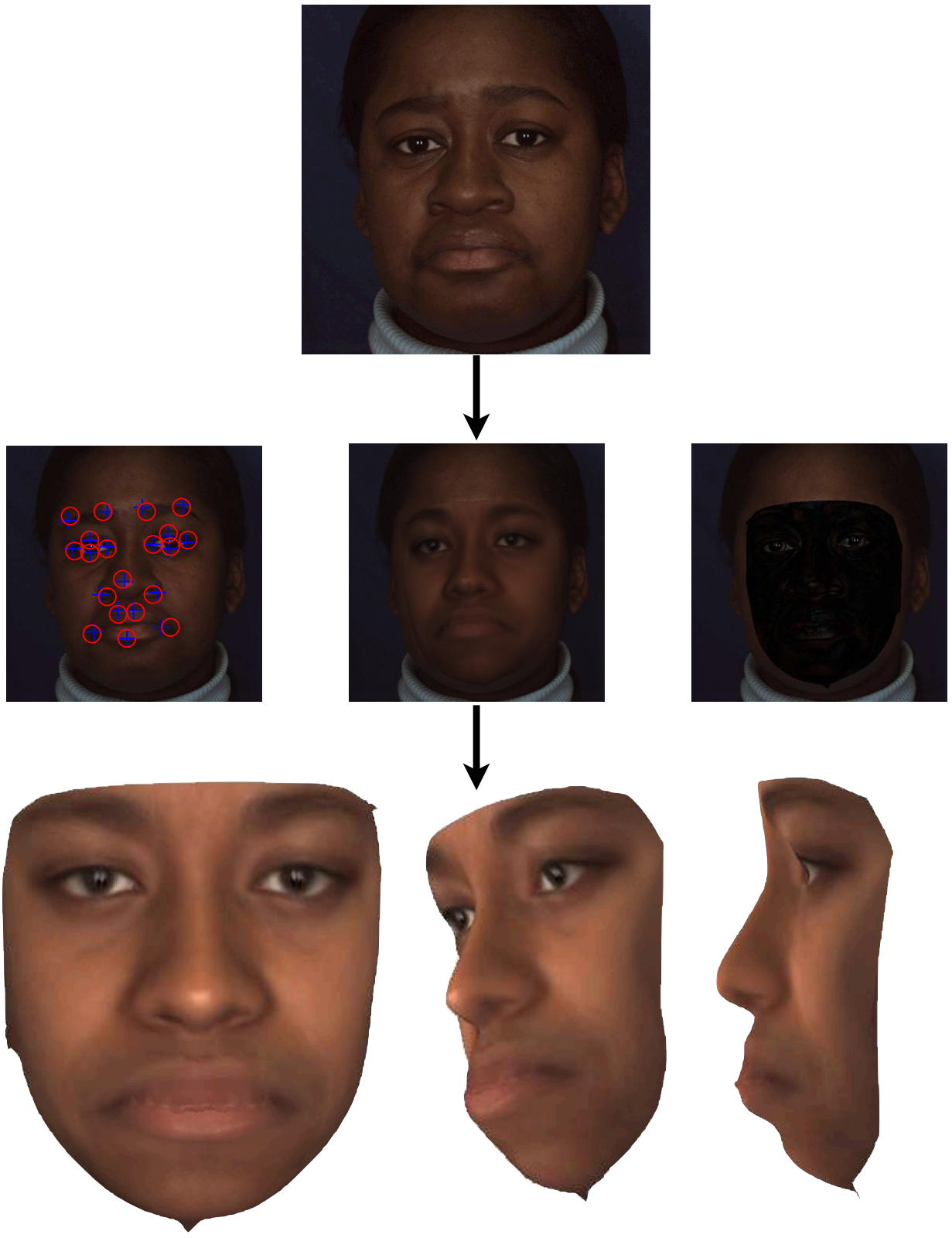

Figure 7.15: As figure 7.13, but for a matching between our model and a sad target image. The model is able to match both identity and emotion of the target image well, dealing with the low lighting and dark complexion of the subject.

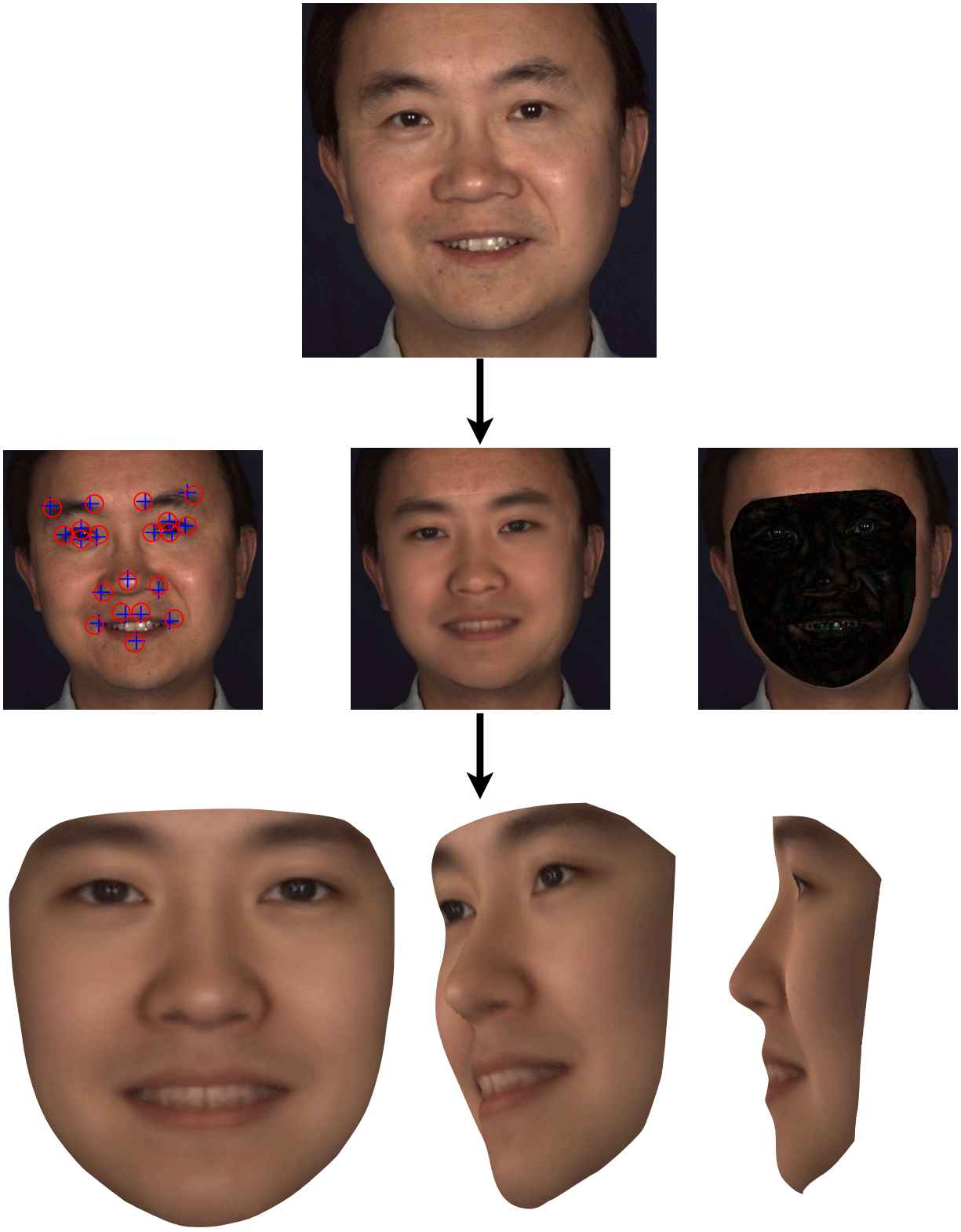

Figure 7.16: As figure 7.13, but for a matching between our model and a happy target image. The error results from the projection test suggest that this is the most difficult emotinon to reproduce, but the model produces a relatively faithful result.

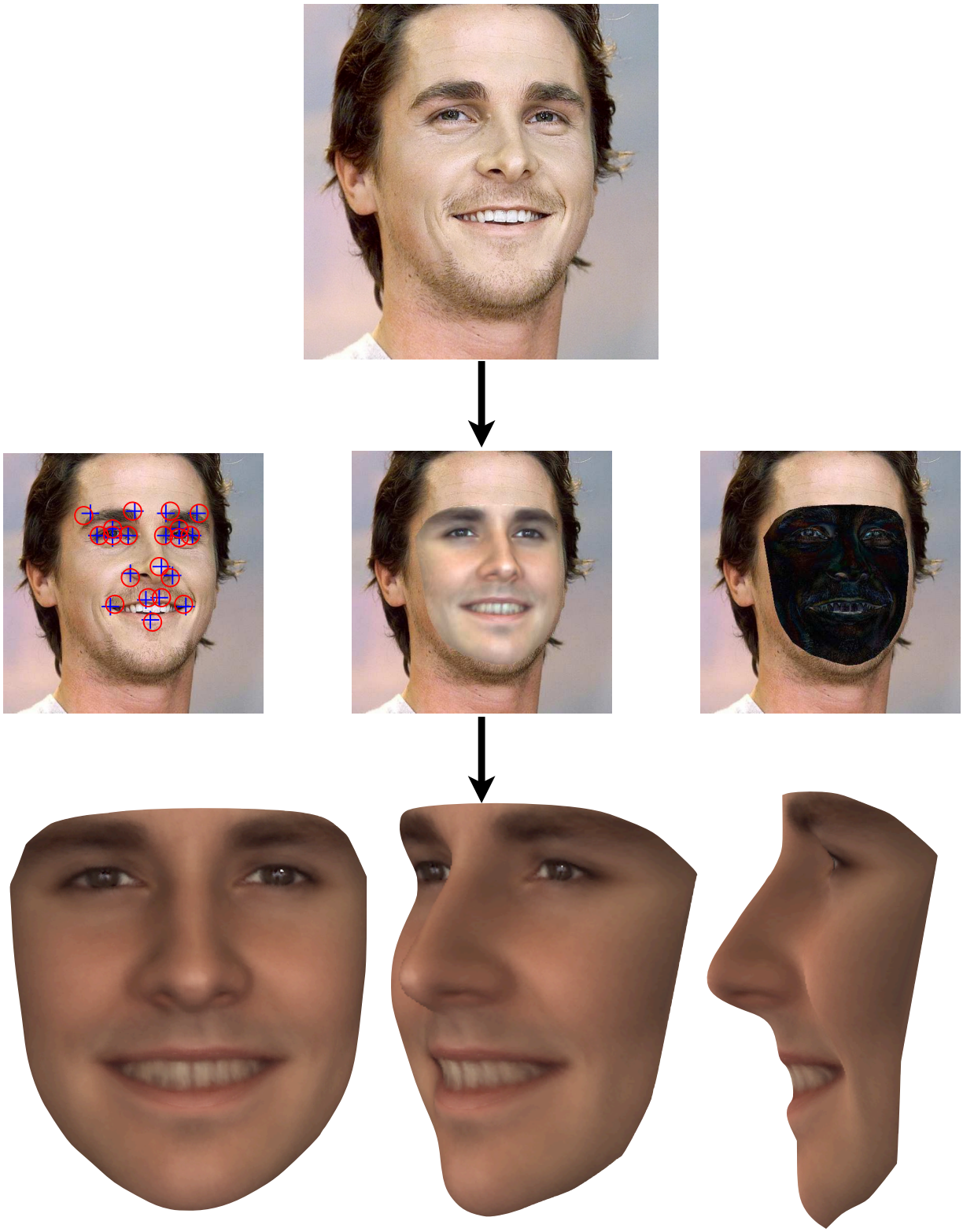

Figure 7.17: As a final test we present another matching with our model, but this time with a subject smiling broadly and at an offset to the camera. Certainly the reproduction suffers as a result, although the accuracy of the nose, eyes, and eyebrows is fairly good. Where the model fails is in capturing the specific nature of the targets smile, the recognisable teeth and the slight opening into the mouth seen below them.

Chapter 8

Conclusion

The work done in this thesis leaves much scope for further study. As stated at the start of chapter 3, the two factors that influence the quality of a 3DMM are the quality of the input data, and how successfully dense correspondence can be established. In regards to the former, the BU-4DFE database employed here leaves much to be desired. Many faces were blurred to some degree, or slightly at an angle to the capture device. This in turn posed problems due to the pseudo-3D nature of the data, both in terms of detail retention, and in landmarking. In short, it would be insightful to use the software tools developed in this thesis on a higher quality, fully 3D dataset, in order to fully analyse its potential.

As for the latter, there are numerous areas for further research. The TPS equation utilized at the core of this pipeline is part of a larger rich family of radial basis functions. Others [3] have used variations on the basic TPS employed here which are based around a physically motivated bending energy that is arguably more applicable to the human face. Fully exploring this space of functions for a superior warp would be an interesting extension.

Another factor that might be limiting the potential of the pipeline proposed is the cylindrical approximation that is made in the warping stage. Others in the field [18][16] have investigated geodesic-preserving surface parametrisations for mesh simplification - potentially the application of similar flattening techniques would allow for fairer warps to be made.

Perhaps the biggest improvement that could be made though, would be to separate teeth and eye reproduction from the rest of the model. The fact that TPS works as an interpolant is because the features of the face can be deformed from one face to the next. However, consider the case of a subject going from a neutral face to smiling broadly, tongue and gums exposed. Although the skin and lips deform in a straightforward fashion, the thin line in the middle of pursed lips does not deform into the complex features of the mouth - these are simply exposed as a consequence

of the deformation of the lips and face. Likewise the iris should never deform in shape, all that is important is how much of the fixed iris is hidden or exposed by the movement of the eyelids. Building a model which accounts for this (perhaps building separate attribute spaces for the inner mouth and eye with their own principal component basis) could potentially vastly improve the performance of the model for dealing with complex emotions.

With all this being said, we have regardless developed a full set of tools for producing 3D Morphable Facial Models from raw facial meshes, and shown that the models it produces have characteristics in line with the Basel Face Model, the standard in the field. Furthermore we have expanded on the state-of-the-art by building a model from faces displaying a range of emotions without having to make special considerations for non-neutral subjects. Interesting evidence was found that a natural separation of emotion and identity might occur when building such 'emotion-native' models, although admittedly further work is required in this particular area before any definite conclusions can be drawn. Finally the mixed-emotion model produced with our pipeline was able to successfully be used in an image matching application, importantly being able to recreate faces displaying emotion with encouraging success. This forms a solid basis for morphable model construction, on which any of the above mentioned avenues for extension could be easily explored.

Appendix A

Bra-ket Notation

Quantum mechanics is formulated exclusively in high-dimensional complex vector spaces. Operations like projecting onto a given basis, changing basis, and calculating inner products are encountered frequently. Dirac's 'bra-ket' notation was developed to make performing these operations simpler. We adopt the use of this notation in chapter 6, with the only stipulation being that while Dirac notation normally implies a complex vector space, our use is confined to real vector spaces only.

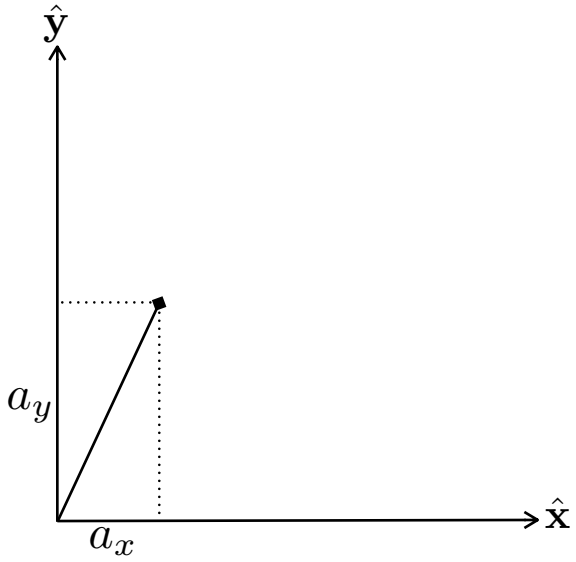

Figure A.1: A simple two dimensional vector.

Figure A.1 shows a vector $\boldsymbol{a}$ represented in a two dimensional real basis (x, y). Commonly, $\boldsymbol{a}$ would be expressed in a column notation as

$$\boldsymbol{a} = \begin{bmatrix} a_x \\ a_y \end{bmatrix} \tag{A.1}$$

where $a_x, a_y \in \mathbb{R}$. Although it goes unsaid, in this representation $\boldsymbol{a}$ is tied to a specific set of basis vectors, namely

$$\hat{\boldsymbol{x}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad \hat{\boldsymbol{y}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{A.2}$$

where more fully we could write

$$\boldsymbol{a} = a_x\hat{\boldsymbol{x}} + a_y\hat{\boldsymbol{y}} \tag{A.3}$$

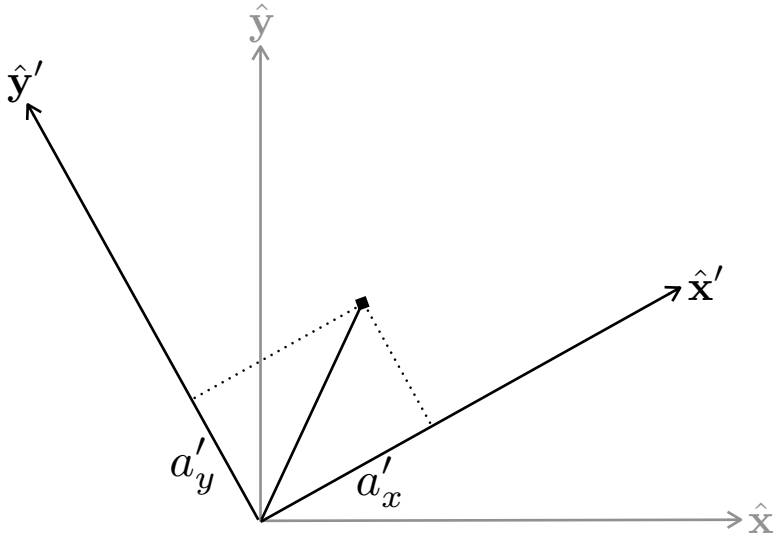

Figure A.2: The same vector object, represented in a new basis.

Figure A.2 shows the same set-up as before, only now a new basis (x', y') is also included. If we wanted to represent $\boldsymbol{a}$ in this new basis we would need to show that the basis has changed by a change of notation

$$\boldsymbol{a}' = \begin{bmatrix} a'_x \\ a'_y \end{bmatrix} \tag{A.4}$$

as the values of the column vector has obviously changed. The relationship between $\boldsymbol{a}$ and $\boldsymbol{a'}$ is given by a transformation matrix which is nothing more than the dot products of the new basis with the old

$$\begin{bmatrix} a'_x \\ a'_y \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}.\boldsymbol{x}' & \boldsymbol{y}.\boldsymbol{x}' \\ \boldsymbol{x}.\boldsymbol{y}' & \boldsymbol{y}.\boldsymbol{y}' \end{bmatrix} \begin{bmatrix} a_x \\ a_y \end{bmatrix} \tag{A.5}$$

where the dot product is defined as $\boldsymbol{x}.\boldsymbol{y} = \boldsymbol{x}^T\boldsymbol{y}$. However, as figure A.2 shows $\boldsymbol{a}$ and $\boldsymbol{a'}$ are truly the same vector object, just described in different ways. The fact that the two representations need to be seperately tracked in our equations is confusing, and doesn't add any more information about

the nature of the vector itself.

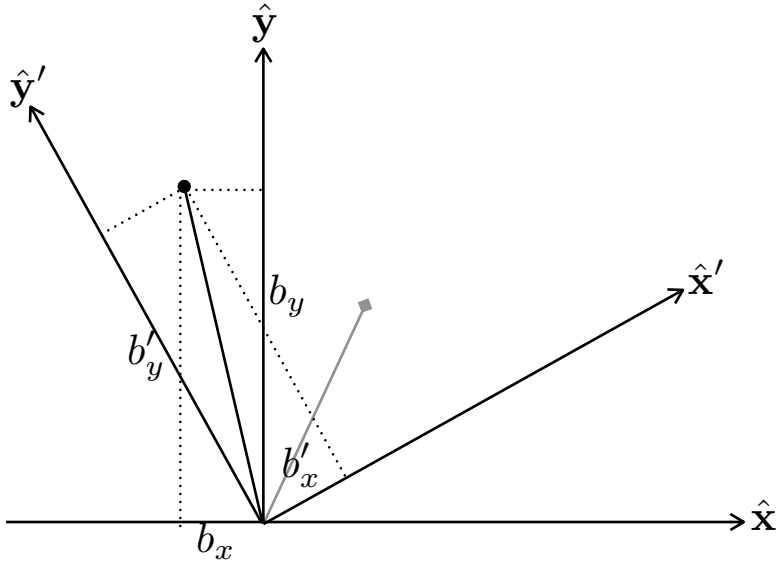

Figure A.3: A new vector represented in both bases.

Figure A.3 adds a second vector, $\boldsymbol{b}$, expressed with respect to the two bases in exactly the same manner as $\boldsymbol{a}$.

Performing operations like the dot product is awkward in this notation as care must be taken to ensure that both vectors are in the same basis, otherwise it is meaningless. This is an artefact of our choice of notation and nothing more, as of course the inner product between the two vectors is invariant to the basis of representation.

Dirac's elegant solution to these notational issues was to abstract a vector away from a particular basis (figure A.4). The 'ket' $|a\rangle$ is the Dirac equivalent to the vector $\boldsymbol{a}$ *and* the vector $\boldsymbol{a'}$; it describes the vector independently of any basis. Kets can be added to other keys just as vectors are added, and this means it's always possible to expand a ket in a basis of other kets

$$|a\rangle = a_x |x\rangle + a_y |y\rangle \tag{A.6}$$

$$|a\rangle = a'_x |x'\rangle + a'_y |y'\rangle \tag{A.7}$$

The other key object in the notation, a 'bra', is conveniently defined as the transpose of the ket

$$\langle a| \equiv |a\rangle^T \tag{A.8}$$

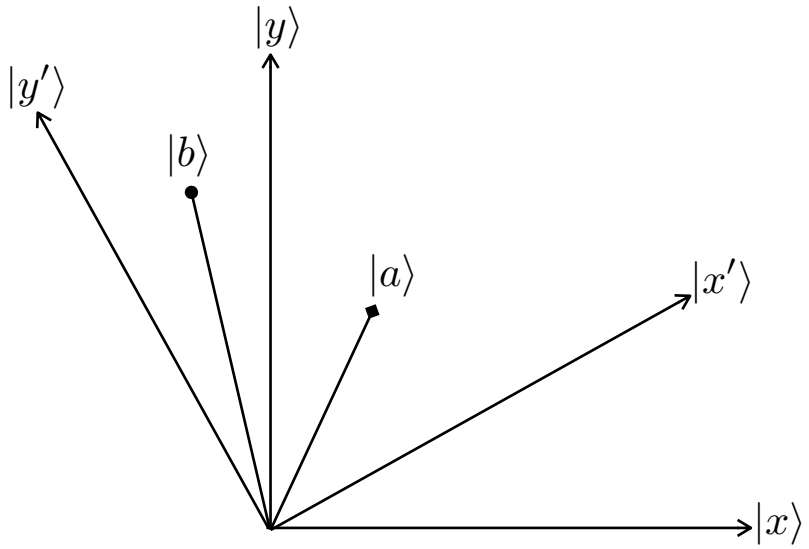

Figure A.4: In bra-ket notation, the focus is on the relationship between the vectors, rather than any specific basis.

which allows for a very natural notation for computing the inner product between two kets as

$$\langle a|b\rangle \tag{A.9}$$

Note that this expression does not explicitly state a basis for either of these kets. It could be the simple case where we choose to expand both in the same basis

$$|a\rangle = a_x |x\rangle + a_y |y\rangle \tag{A.10}$$

$$|b\rangle = b_x |x\rangle + b_y |y\rangle \tag{A.11}$$

and the calculation would be

$$\langle a|b\rangle = (a_x \langle x| + a_y \langle y|)(b_x |x\rangle + b_y |y\rangle) \tag{A.12}$$

$$= a_x b_x \langle x|x\rangle + a_y b_x \langle y|x\rangle + a_x b_y \langle x|y\rangle + a_y b_y \langle y|y\rangle \tag{A.13}$$

$$= a_x b_x + a_y b_y \tag{A.14}$$

as of course

$$\langle x|x\rangle = \langle y|y\rangle = 1 \tag{A.15}$$

$$\langle x|y\rangle = \langle y|x\rangle = 0 \tag{A.16}$$

which is simply a statement that the basis is orthonormal.

However, one could choose to expand $|a\rangle$ in the original basis and $|b\rangle$ in the rotated frame

$$|b\rangle = b'_x |x'\rangle + b'_y |y'\rangle \tag{A.17}$$

and the notation will take care of itself

$$\langle a|b\rangle = a_x b'_x \langle x|x'\rangle + a_y b'_x \langle y|x'\rangle + a_x b'_y \langle x|y'\rangle + a_y b'_y \langle y|y'\rangle \tag{A.18}$$

(c.f with equation A.5 to see that the notation naturally leads to the same thing as changing both vectors into the same basis and then performing the dot product).

For practical calculations, we can always choose one basis and consider the coefficients of the ket in that basis as the entries of a column vector

$$|a\rangle = a_x |x\rangle + a_y |y\rangle \qquad \boldsymbol{a} = \begin{bmatrix} a_x \\ a_y \end{bmatrix} \tag{A.19}$$

and likewise the coefficients of a bra as a row vector.

$$\langle a| = \langle x| a_x + \langle y| a_y \qquad \boldsymbol{a}^T = \begin{bmatrix} a_x & a_y \end{bmatrix} \tag{A.20}$$

and then the notation is nothing more than a different way of writing column and row vectors. However by using bra-ket notation we separate operations on vectors from the complications that arise from representation of the vectors in a particular basis. This is useful in this thesis, where we frequently wish to add face attribute vectors to vectors built from the principal component basis.

Bibliography

[1] Ali Al-Sharadqah and Nikolai Chernov. Error analysis for circle fitting algorithms. *Electron. J. Statist.*, 3:886–911, 2009.

[2] Brian Amberg. *Editing Faces in Videos.* PhD thesis, Uinversity of Basel, 2011.

[3] Ankur Patel and William A. P. Smith. 3D morphable face models revisited. *CVPR*, pages 1327–1334, 2009.

[4] V. Blanz, C. Basso, T. Poggio, and T. Vetter. Reanimating faces in images and video. *Comput. Graph. Forum*, 22(3):641–650, 2003.

[5] Volker Blanz and Thomas Vetter. A Morphable Model for the Synthesis of 3D Faces. In *Computer Graphics Proc.*, SIGGRAPH '99, pages 187–194, 1999.

[6] Fred L Bookstien. Principal Warps: Thin-Plate Splines and the Decomposition of Deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, June 1989.

[7] A.M. Burton and J.R. Vokey. The face-space typicality paradox: Understanding the face-space metaphor. *The Quarterly Journal of Experiemental Psychology*, 51(3):475–483, 1998.

[8] D. Cosker. A FACS valid 3D dynamic action unit database with applications to 3D dynamic morphable facial modeling. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, pages 2296–2303, 2011.

[9] B. Delaunay. Sur la sphere vide. *Otdelenie Matematicheskikh i Estestvennykh Nauk*, 7:793–800, 1934.

[10] P. A. M. Dirac. A new notation for quantum mechanics. In *Mathematical Proceedings of the Cambridge Philosophical Society*, 35, pages 416–418, 1939.

[11] I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis.* John Wiley and Sons, 1998.

[12] IEEE. *A 3D Face Model for Pose and Illumination Invariant Face Recognition*, Genova, Italy, 2009.

[13] Joan Albort Medina. *Study of implementations and extensions of 3D morphable models.* MSc thesis, 2011.

[14] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of Imaging Understanding Workshop*, pages 121–130, 1981.

[15] G. Sandbach, S. Zafeiriou, M. Pantic, and L. Yin. Static and Dynamic 3D Facial Expression Recognition: A Comprehensive Survey. *Image and Vision Computing. (in press)*, June 2012.

[16] A. Sheffer, B. Levy, M. Mogilnitsky, and A. Bolomyakov. ABF++: Fast and robust angle based flattening. *ACM Trans. Graphic.*, 24(2):311–330, 2005.

[17] Lijun Yin, Xiaochen Chen, Yi Sun, Tony Worm, and Michael Reale. A High-Resolution 3D Dynamic Facial Expression Database. In *FGR08*, 2008.

[18] G. Zigelman, R. Kimmel, and N.Kiryati. Texture mapping using surface flattening via multi-dimensional scaling. *IEEE Vis. Comp. Gr.*, 8(2):198–207, 2002.