# 3D Morphable Models: Data Pre-Processing, Statistical Analysis and Fitting

Ankur Patel

Submitted for the degree of Doctor of Philosophy

Department of Computer Science

The University of York

June, 2011

# Abstract

This thesis presents research aimed at using a 3D linear statistical model (known as a 3D morphable model) of an object class (which could be faces, bodies, cars, etc) for robust shape recovery. Our aim is to use this recovered information for the purposes of potentially useful applications like recognition and synthesis. With a 3D morphable model as its central theme, this thesis includes: a framework for the groupwise processing of a set of meshes in dense correspondence; a new method for model construction; a new interpretation of the statistical constraints afforded by the model and addressing of some key limitations associated with using such models in real world applications.

In Chapter 1 we introduce 3D morphable models, touch on the current state-of-the-art and emphasise why these models are an interesting and important research tool in the computer vision and graphics community. We then talk about the limitations of using such models and use these limitations as a motivation for some of the contributions made in this thesis.

Chapter 2 presents an end-to-end system for obtaining a single (possibly symmetric) low resolution mesh topology and texture parameterisation which are optimal with respect to a set of high resolution input meshes in dense correspondence. These methods result in data which can be used to build 3D morphable models (at any resolution).

In Chapter 3 we show how the tools of thin-plate spline warping and Procrustes analysis can be used to construct a morphable model as a shape space. We observe that the distribution of parameter vector lengths follows a chi-square distribution and discuss how the parameters of this distribution can be used as a regularisation constraint on the length of parameter vectors.

In Chapter 4 we take the idea introduced in Chapter 3 further by enforcing a hard constraint which restricts faces to points on a hyperspherical manifold within the parameter space of a

linear statistical model. We introduce tools from differential geometry (log and exponential maps for a hyperspherical manifold) which are necessary for developing our methodology and provide empirical validation to justify our choice of manifold. Finally, we show how to use these tools to perform model fitting, warping and averaging operations on the surface of this manifold.

Chapter 5 presents a method to simplify a 3D morphable model without requiring knowledge of the training meshes used to build the model. This extends the simplification ideas in Chapter 2 into a statistical setting. The proposed method is based on iterative edge collapse and we show that the expected value of the Quadric Error Metric can be computed in closed form for a linear deformable model. The simplified models can used to achieve efficient multiscale fitting and super-resolution.

In Chapter 6 we consider the problem of model dominance and show how shading constraints can be used to refine morphable model shape estimates, offering the possibility of exceeding the maximum possible accuracy of the model. We present an optimisation scheme based on surface normal error as opposed to image error. This ensures the fullest possible use of the information conveyed by the shading in an image. In addition, our framework allows non-model based estimation of per-vertex bump and albedo maps. This means the recovered model is capable of describing shape and reflectance phenomena not present in the training set. We explore the use of the recovered shape and reflectance information for face recognition and synthesis.

Finally, in Chapter 7 we provide concluding remarks and discuss directions for future research.

# Table of Contents

# List of Tables

# List of Figures

# List of Datasets

Following is a list of 3D datasets used in this thesis.

- **[76]**: This dataset (referred as the Basel Face Model) contains a 3D morphable model built using 200 face meshes and they provide an additional 10 unseen (out-of-sample) meshes. All the meshes were obtained using a three view structured light scanner (ABW-3D) and are set into dense correspondence using a modified version of the algorithm of [6]. For this dataset, the training meshes (used for building the model) are not available. Additionally, this dataset is topologically symmetric.

- **[1]**: This dataset contains 100 meshes obtained using a Cyberware 3030PS laser range scanner. The meshes were set into dense correspondence using the modified optical flow algorithm of [14].

- **[104]**: This dataset contains 385 meshes of a single subject acquired using spacetime stereo. Vertex motion is tracked using optical flow and the mouth is unconnected (i.e. it contains a hole).

- **[45]**: This dataset consists of 550 full body meshes of 114 subjects (in 10 poses). The meshes were captured using a Vitronic laser scanner and are set into dense correspondence using an approach similar to [6].

- A dataset containing 150 meshes obtained using a Cyberware 3030PS laser range scanner. The meshes were set into dense correspondence using a thin-plate spline warp (refer Section 3.3.2).

# Acknowledgements

To:

- **My Parents and Brother**: Thank you for all your unstinting support and encouragement throughout the last three years (and prior).
- **Will**: It has been an absolute pleasure and privilege to work with you over the last three years. It is impossible to express my gratitude in a few words. Thank you for everything.
- **Prof. Hancock, Prof. Wilson and Dr. Bors**: Thank you for all the guidance and support whenever I needed it.
- **Colleagues, Friends and Family**: All your contributions (however big or small) are much appreciated and will never be forgotten.
- **EPSRC and The Department of Computer Science**: Thank you for supporting this work.

# Declaration

I declare that all the work in this thesis is solely my own, except where attributed and cited to other authors. Most of the material in this thesis has been previously published (or accepted for publication). Below is a complete list of publications on which the Chapters 2 to 6 are based.

- **Chapter 2**
  - A. Patel and W.A.P. Smith. Automated Construction of Low Resolution, Texture-Mapped,Class-Optimal Meshes. *IEEE Trans. Vis. Comp. Gr.*, 2011 (Accepted).
- **Chapter 3**
  - A. Patel and W.A.P. Smith. 3D Morphable Face Models Revisited. In *Proc. CVPR*, pages 1327-1334, 2009.
- **Chapter 4**
  - A. Patel and W.A.P. Smith. Exploring the Identity Manifold: Constrained Operations in Face Space. In *Proc. ECCV*, pages 112-125, 2010.
- **Chapter 5**
  - A. Patel and W.A.P. Smith. Simplification of 3D Morphable Models. In *Proc. ICCV*, 2011 (Accepted).
- **Chapter 6**
  - A. Patel and W.A.P. Smith. Shape-from-shading driven 3D Morphable Models for Illumination Insensitive Face Recognition. In *Proc. BMVC*, 2009.

# Chapter 1

# Introduction

A set (class) of meshes are in dense correspondence (Figure 1.1), if they share the same connectivity (topology) and corresponding vertices have equivalent meaning in some sense, for example functional, topological, evolutionary etc. Such data can arise in at least two ways (Figure 1.2). The first is when a deforming surface is tracked over time (e.g. [35, 104]), sometimes known as 4D mesh data. In this case, shape variation is the result of surface deformation. The second is when an object class such as human faces [14], body shapes [45] or automobiles [60] is statistically modelled. In this case, shape variation is related to change in identity.

Efficiently and accurately modelling the class of human face shapes has received longstanding attention in computer vision and graphics. In particular, 3D face shape has proved to be a competitive biometric with numerous commercial applications [86]. A recent technology evaluation (FRVT 2006 [80]) showed using 3D face scans results in orders of magnitude improvement in recognition performance over using 2D image data alone. To this end, a Principal Component Analysis (PCA) based linear statistical model of 3D human faces (known as a 3D morphable model [14]) has received considerable research focus in recent years. The model provides a parametric description of object instances and can be fitted to data by using the model generatively in an analysis-by-synthesis framework. The statistical constraint afforded by these models leads to robust performance when fitting to real world, potentially noisy data.

The body of work which uses morphable models offers globally accurate and robust shape recovery from real images. Additionally, given a single image of a person, the deformable 3D model provides automatic estimates of 3D shape and texture/albedo (the intrinsic parameters), which are fully independent of the estimated 3D scene parameters (the extrinsic parameters).

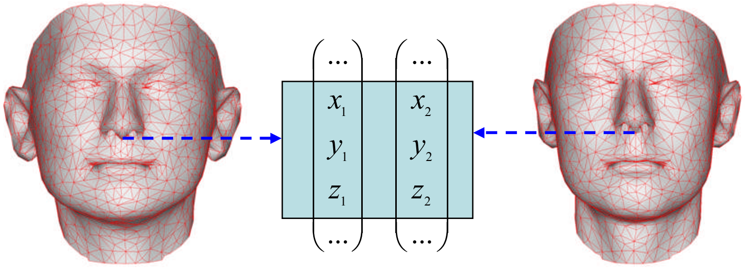

Figure 1.1: A class of 2 meshes in dense correspondence. The meshes share the same topology (red triangular facets) and corresponding vertices (for example the tip of the nose indicated by the blue arrows) have equivalent meaning.

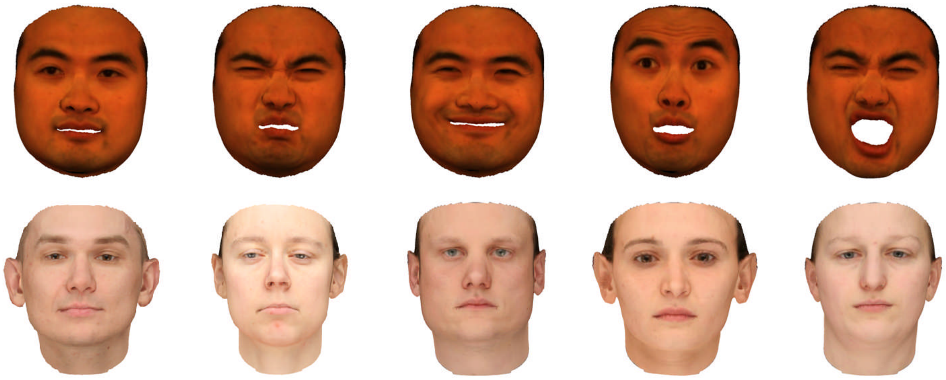

Figure 1.2: Shape variation due to surface deformation (top row). Shape variation due to change in identity (bottom row).

Hence these models provide an excellent means for solving the two main challenges for face recognition i.e. variations in pose and illumination [86]. Besides providing a route to state-of-the-art performance in pose and illumination invariant face recognition [15, 76], these models have many other potentially useful applications such as markerless motion capture [104], image based editing and animation [11] and texture transfer and morphing [4].

However, there are several key challenges which hinder the further development of techniques which use these models. Applications involving face analysis require fitting the model to data (such as images, photometric normals or noisy range scans). The fitting process is computationally expensive and prone to converge on local minima close to the mean. For robust performance, parameter tuning is also required. Applications involving face synthesis require that the model is capable to generating plausible and realistic face instances. Model dominance means synthesised instances lack high frequency detail and hence realism. In summary, we

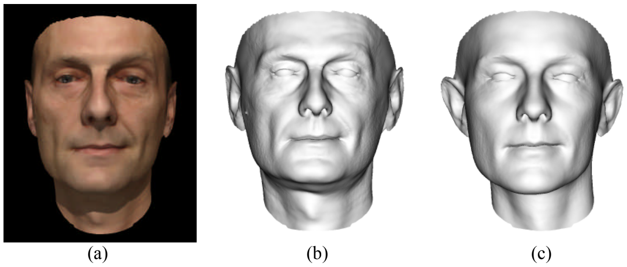

Figure 1.3: (a) Input rendered image, b) Ground truth and c) Shape estimated using a state-of-the-art model fitting algorithm [76]. The algorithm uses an analysis-by-synthesis technique to optimise an objective comprising three error terms based on landmark points, contours and image intensity. In spite of using this state-of-the-art model fitting approach, the estimated shape fails to capture all the atypical features of the face (wrinkles on the cheek). This clearly emphasises model dominance as a serious weakness and shows the need to study additional cues to overcame this drawback.

identify the key problems faced by morphable models as follows:

- **Computational expense**: Fitting the model generally requires solution of a highly nonconvex optimisation problem. In order to avoid local minima, global optimisation strategies such as stochastic search are needed to fit the model to data [15]. Such methods are slow to converge and require many iterations to be confident of obtaining a solution close to the global minimum. In addition, each iteration may require rendering the current estimate of the model and computing the error to the input. Since state of the art 3D morphable models have a very high resolution (typically $> 50000$ vertices [15, 76]), this is an expensive process.

- **Model dominance**: The quality of a reconstructed or synthesised face shape is critically dependent on whether the model was trained using data which contained similar examples. In other words, the ability of the model to generalise to new examples is limited. This is particularly evident in the loss of high frequency (fine scale) surface details which are often the most distinguishing features of a face. Hence, there is a fundamental limit on the accuracy the model can attain. See Figure 1.3 for an illustration. However, even this limit is difficult to reach since iterative methods commencing from a mean initialisation are likely to converge on a minimum close to the mean, further exacerbating the problem of model dominance. Several attempts have been made to overcome this problem. For

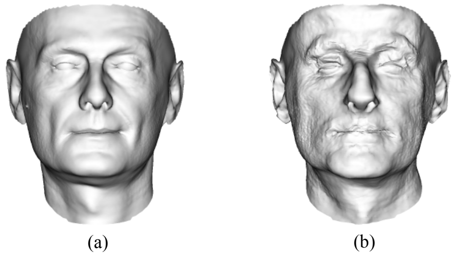

Figure 1.4: (a) Unseen input mesh (ground truth) and (b) Least squares shape estimate. In spite of perfect correspondence between the input data and the model and a linear objective function, we can obtain a grossly over fitted shape estimate. This clearly illustrates the need to study statistical constraints afforded by the model in order to obtain the right trade off between fitting quality and shape plausibility.

example, Romdhani et al. [87] proposed augmenting the objective function with additional terms (such as edges, specular highlights and pixel intensities) which encourage convexity of the search landscape.

- **Parameter tuning**: Obtaining the optimum trade off between fitting quality and shape plausibility is data-dependent and requires manual tuning of a regularisation parameter [12]. If the prior is underweighted, the model will overfit the data and the resulting surface may contain many severe artefacts. If the prior is overweighted, the problem of model dominance is further increased and the solution is forced to lie close to the mean. See Figure 1.4 for an illustration.

## 1.1 Contributions

The central theme to this thesis is a 3D morphable model. The contributions made are aimed at addressing some open problems related to model construction, statistical modelling and fitting. Specifically, this thesis presents: a framework for the groupwise processing of a set of meshes in dense correspondence; an empirically justified reconsideration of the techniques used to construct the model; a study of the statistical constraints afforded by the model; methods which address the afore mentioned limitations associated with using such models in real world applications. Figure 1.5 provides a detailed visual schematic (flow diagram) of the steps involved

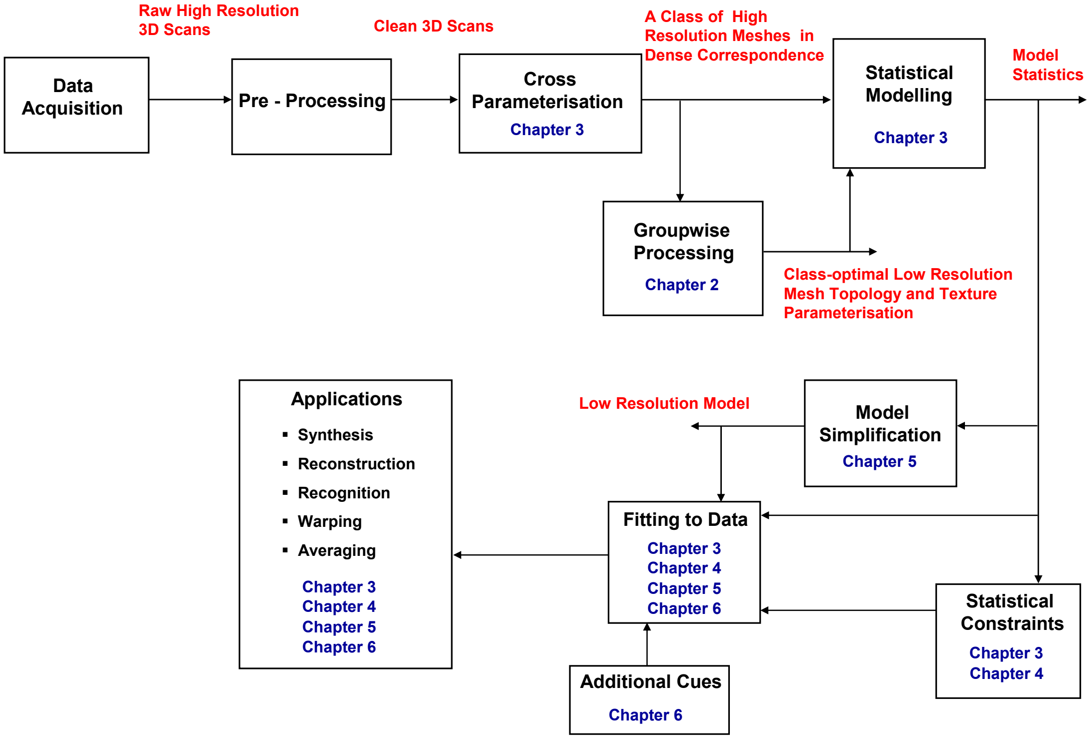

Figure 1.5: Outline of the contributions made in this thesis.

in building and using a morphable model. We highlight where the chapters of this thesis contribute to these steps.

In detail, the contributions made in this thesis are as follows:

- **Processing a Class of Meshes (Chapter 2)**: We present a framework for the groupwise processing of a set of meshes in dense correspondence. We extend a number of mesh processing tools to operate in a groupwise manner. Specifically, we present a geodesic based surface flattening and spectral clustering algorithm which estimates a single class-optimal flattening. We also show how to modify an iterative edge collapse algorithm to perform groupwise simplification whilst retaining the correspondence of the data. Finally, we show how to compute class-optimal texture coordinates for the simplified meshes. We present alternative algorithms for topologically symmetric data which yields a symmetric flattening and low resolution mesh topology. We present flattening, simplification and texture mapping results on three different datasets and our approach results in data that allows the construction of 3D morphable models (at any resolution).

- **3D Morphable Models Revisited (Chapter 3)**: We revisit the process of constructing a high resolution 3D morphable model of face shape variation. We demonstrate how the statistical tools of thin-plate splines and Procrustes analysis can be used to construct a morphable model that is both more efficient and generalises to novel face surfaces more accurately than previous models. We also reformulate the probabilistic prior that the model provides on the distribution of parameter vector lengths. This distribution is determined solely by the number of model dimensions and can be used as a regularisation constraint in fitting the model to data without the need to empirically choose a parameter controlling the trade off between plausibility and quality of fit. As an example application of this improved model, we show how it may be fitted to a sparse set of 2D feature points (approximately 100). This provides a rapid means to estimate high resolution 3D face shape for a face in any pose given only a single face image. We present experimental results using ground truth data and hence provide absolute reconstruction errors.

- **Manifold Based Constraints (Chapter 4)**: We constrain faces to points on a manifold within the parameter space of a linear statistical model. The manifold is the subspace of faces which have maximally likely distinctiveness and different points correspond to unique identities. We develop a novel implementation of log and exponential maps for a hyperspherical manifold. We show how these tools can be used to replace linear operations such as warping and averaging with operations on the surface of this manifold. Finally, we use the manifold to develop a new method for fitting a statistical face shape model to data, which is both robust (avoids overfitting) and overcomes model dominance (is not susceptible to local minima close to the mean face). Our method outperforms generic nonlinear optimisers based on the BFGS Quasi-Newton method and the Levenberg-Marquardt algorithm when fitting using a state-of-the-art morphable model.

- **Model Simplification (Chapter 5)**: We show how to simplify a 3D morphable model.

Our method only requires knowledge of the original highest resolution statistical model and leads to low resolution models in which the model statistics are a subset of the original high resolution model. We employ an iterative edge collapse strategy, where the deleted edge is chosen as a function of the model statistics. We show that the expected value of the Quadric Error Metric can be computed in closed form for a PCA deformable model. Model parameters obtained using the model at any resolution (lower) can be used to reconstruct a high resolution surface, providing a route to super-resolution. We are able to decrease the model resolution and fitting time by factors of approximately 10 and 4 respectively whilst inducing an error which is only slightly larger than the fitting error of the original model.

- **Shape-from-Shading Driven Morphable Models (Chapter 6)**: We show how surface orientation information inferred using shape-from-shading can be used to aid the process of fitting a 3D morphable model to an image of a face. We consider the problem of model dominance and show how shading constraints can be used to refine morphable model shape estimates, offering the possibility of exceeding the maximum possible accuracy of the model. We use this observation to motivate an optimisation scheme based on surface normal error. This ensures the fullest possible use of the information conveyed by the shading in an image. Moreover, our framework allows estimation of per-vertex albedo and bump maps which are not constrained to lie within the span of the model. This means the recovered model is capable of describing shape and reflectance phenomena not present in the training set. We show reconstruction and synthesis results and demonstrate that the shape and albedo estimates can be used for illumination insensitive face recognition using a single gallery image.

## 1.2 Thesis Outline

Each contributing chapter (2 - 6) begins by introducing the problem to be addressed, provides a review of the relevant prior work and states the contributions made. This is followed by a detailed presentation of the theoretical contribution and experimental validation of the proposed methods. Conclusions are provided to end each chapter. All the notations and symbols introduced in each chapter are explained as relevant and are to be used in reference to that particular chapter only. Finally, in Chapter 7 we provide some concluding thoughts and discuss directions for future research.

# Chapter 2

# Processing a Class of Meshes

## 2.1 Introduction

High resolution meshes with per-vertex texture information arise from data acquisition devices such as structured light [76] or laser range scanners [14], or more recently from photometric light stages [64]. If we wish to simplify such a mesh whilst retaining the mapping to the original high resolution texture, a number of existing tools can be used. Typically, the high resolution mesh is flattened using any one of the huge range of surface flattening algorithms [92] to establish a texture parameterisation. The mesh is then simplified using a simplification algorithm such as QSlim [36] and the texture coordinates of the resulting vertices interpolated. Finally, the low resolution mesh may be rendered using the original high resolution texture. However, such tools are not available when the input is a class of meshes which share a common parameterisation, i.e. they are in dense correspondence.

Members of such a class have fixed mesh topology (or connectivity) but varying shape (i.e. vertex positions). The shape variations can be the result of surface deformation (sometimes known as 4D data mesh data) [35, 104] or changes in identity [14, 45, 60].

In this chapter we present an end-to-end system for obtaining a single (possibly symmetric) low resolution mesh topology and texture parameterisation which are optimal with respect to a set of input meshes. We do not consider the problem of obtaining a consistent parameterisation of the set of meshes (known as *cross-parameterisation*), this is already well studied [6, 14, 55, 60, 82]. Our aim here is instead to compute class-optimal flattenings, simplifications and low resolution embeddings which preserve cross-parameterisation over the input set of meshes.

The most obvious application of the methods we present is in compressing 4D mesh data whilst preserving inter-frame correspondence. However, we focus on results for (low resolution) statistical shape/texture modelling. Additionally, our method could form part of a system for automated construction of efficient and robust facial rigs for varying identity or expression.

## 2.2 Related Work

To our knowledge, the problem of groupwise surface flattening has not previously been considered. Surface flattening methods can be categorised with respect to their parameterisation domain and the type of parametric distortion minimised [92]. Early work used fixed boundary parameterisations, e.g. Floater's Mean Value Coordinates [33]. The advantage of such an approach is that it can guarantee bijectivity whilst requiring only the solution of a linear equation. However, it induces significant distortion. Linear methods with a free boundary such as the Least Squares Conformal Map of Levy et al. [61], can reduce this distortion although bijectivity (either local or global) can no longer be guaranteed. Nonlinear methods offer significantly less distortion at increased computational expense.

Zigelman et al. [109] showed how classical MDS could be used to preserve geodesic distances in an efficient manner although the embedding cannot be guaranteed to be bijective. Zhou et al. [108] extended this approach by partitioning the mesh into a number of separate charts, each of which could be flattened bijectively. Sheffer et al. [91] proposed a nonlinear angle based flattening which can be efficiently implemented. Their approach yields an embedding which is locally bijective, meaning that no triangles become inverted.

More recently, Liu et al. [62] proposed a local/global approach in which 3D triangles are mapped to the plane locally and then globally stitched together by solving a sparse linear system. When the triangles are mapped in an *as-rigid-as-possible* manner, the embedding seeks to preserve global shape. Ben-Chen et al. [9] present a conformal flattening method which aims to concentrate all the 3D curvature at a small number of mesh vertices. These singular vertices are chosen automatically and are used to cut the mesh to obtain a disk topology. The method is again linear and yields parameterisations with low quasi-conformal distortion. Instead of computing a strictly conformal flattening, Yang et al. [102] find the best discrete conformal mapping that also minimizes area distortion.

The problem of groupwise mesh simplification has been considered before in the context of modelling motion or deformation of a surface. Mohr and Gleicher [73] were the first to propose a method for simplifying a deforming mesh where the sequence is in dense correspondence. They extended Garland and Heckbert's [36] QSlim algorithm such that edge collapses use an error metric which is summed over all samples in the set. A more sophisticated approach was taken by DeCoro and Rusinkiewicz [27]. They perform simplification of skeletally articulated meshes by incorporating knowledge of potential poses into a probability function, which in turn determines the error quadrics associated with the surface. Using Monte Carlo sampling, they obtain a mesh which approximates the likely poses more accurately than the improbable ones. A similar approach is taken by Landreneau and Schaefer [56], who perform simplification based on a set of user-specified poses that are representative of typical deformations. Their approach also generates resolution-dependent skin weights.

We assume that the input meshes are already in dense correspondence and share the same mesh topology. There are already a range of techniques available for compatible remeshing of sets of meshes. These can be classified into domain specific techniques (which may exploit user prescribed constraints) and those that can be applied to generic meshes. Domain specific techniques include Leotta and Mundy [60], who use a manually constructed coarse template vehicle mesh whose vertices are repositioned to match CAD models of vehicles in the training set. Blanz and Vetter [14] use a variant of optical flow to align surfaces which can be parameterised to a 2D texture space. Their flow objective function includes measures of both texture and shape information such as Gaussian curvature, mean curvature and surface normals. Generic techniques include both Kraevoy and Sheffer [55] and Praun et al. [82]. These methods produce a consistent parameterisation so that the meshes share the same base domain and respect pre-selected feature vertices. Amberg et al. [6] extend the ICP framework to nonrigid registration while retaining the convergence properties of the original algorithm.

## 2.3 Contributions

In Section 2.5 we extend the surface flattening approach of [109], based on multidimensional scaling (MDS), to operate in a groupwise manner. We begin by providing examples to show that two naïve approaches do not yield optimal results. We then provide an algorithm for computing a class-optimal surface flattening, in which MDS is applied to the matrix of average normalised geodesic distances. We show how to efficiently flatten a set of large (in terms of number of vertices) meshes, by performing spectral analysis using a key-points subset. The key-points are obtained using a method for groupwise uniform sampling of a set of meshes. We use the recently-reported method of [10] to extend the embedding to the whole mesh, which is once again computed using class-average geodesic distances. The result is a single flattening which minimises the total distortion induced by flattening all meshes in the class. We consider the case of meshes with symmetric topology and provide a modified version of the algorithm to obtain a class-optimal symmetric flattening. We also extend the spectral clustering algorithm of [108] to operate on a set of meshes. In doing so we show how to flatten a set of meshes by partitioning them into class-optimal charts and also overcome the problem of triangle flips.

In Section 2.6 we present a modification of Garland and Heckbert's [36] iterative edge collapse algorithm which allows groupwise simplification of a set of (possibly symmetric) meshes. In Section 2.7 we show how class-optimal texture coordinates can be computed for the simplified mesh structure.

In Section 2.8 we present results for varying identity on faces from the Basel Face Model (BFM) [76] and for facial motion data using the dataset of [104]. We also present results on a set of body meshes obtained from [45]. Finally, we show how our simplifications can be used to construct low resolution 3D morphable models which capture the same statistics as the high resolution data. A summary of our contributions is shown in Figure 2.1.

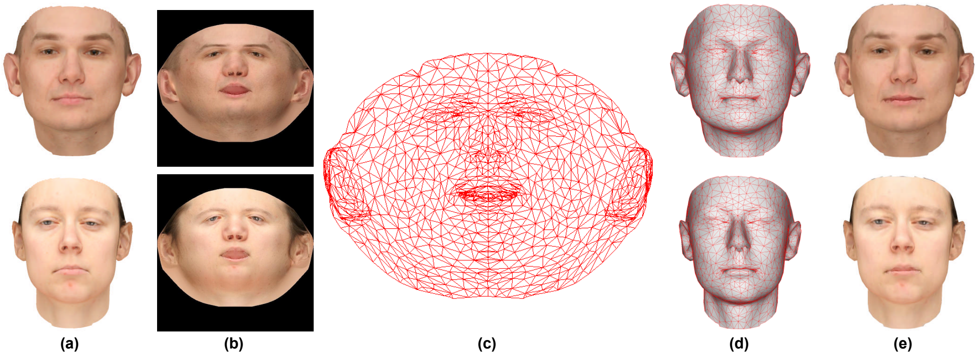

Figure 2.1: From an input set of high resolution meshes (∼ 50k vertices) with per-vertex texture information and in dense correspondence (a), we compute a single class-optimal flattening (b) and simplified (1k vertices) mesh structure (c). Simplified meshes (d) remain in dense correspondence and we compute class-optimal texture coordinates for the low resolution vertices. This allows us to render the simplified meshes with the flattened high resolution textures (e).

## 2.4 Preliminaries

We are interested in processing a set of $F$ meshes which are augmented with per-vertex texture values. We assume that the meshes are already in dense correspondence. That is to say, each mesh has the same number of vertices and that the $i$th vertex in any one mesh corresponds to the $i$th vertex in another. Moreover, each mesh shares the same connectivity.

Formally, we define the $f$th mesh as $\mathcal{M}_f^N = (K^N, \mathbf{S}_f, \mathbf{T}_f)$. Here, $K^N$ is a simplicial complex with $N$ vertices, which is a set whose elements can be vertices $\{i\}$, edges $\{i,j\}$ or triangles $\{i,j,k\}$, with the indices $i,j,k \in [1 \dots N]$. The complex $K^N$ defines the connectivity or topology of a mesh. All meshes in the input set share the same complex. The actual shape of the $f$th mesh is defined by the vector $\mathbf{S}_f \in \mathbb{R}^{3N}$, where the $i$th vertex is given by $\mathbf{v}_i^f = [S_f(3i-2)\ S_f(3i-1)\ S_f(3i)]^T$. Similarly, the per-vertex texture values are given by the vector $\mathbf{T}_f \in \mathbb{R}^{3N}$, where the RGB vector of the texture associated with the $i$th vertex is given by $\mathbf{t}_i^f = [T_f(3i-2)\ T_f(3i-1)\ T_f(3i)]^T$. We centre all meshes, such that $\sum_{i=1}^N \mathbf{v}_i^f = [0\ 0\ 0]^T$.

Although the faces of individual subjects will contain asymmetries [32], facial shapes as a class (and many other biological structures) are symmetric. It has been shown that symmetry is a powerful constraint to exploit in mesh simplification, resulting in more meaningful low resolution meshes [41]. Moreover, it may be more convenient for an artist or animator to rig or design a texture map for a symmetric mesh. For this reason it may be useful to use a symmetric topological structure. We present algorithms for both symmetric and non-symmetric sets of meshes. Where a mesh is symmetric, then for each vertex $\{i\} \in K^N$, there is a corresponding symmetric vertex $\{\text{sym}(i)\} \in K^N$. If $\{i\}$ lies on the line of symmetry, then $\{i\} = \{\text{sym}(i)\}$. Similarly, for each edge $\{i, j\} \in K^N$, there is a corresponding symmetric edge $\{\text{sym}(i), \text{sym}(j)\} \in K^N$ and if both $\{i\}$ and $\{j\}$ lie on the line of symmetry then $\{i, j\} = \{\text{sym}(i), \text{sym}(j)\}$.

## 2.5 Groupwise Surface Flattening

The aim of planar surface flattening is to embed a set of vertices $\{\mathbf{v}_1, \ldots, \mathbf{v}_N\}$ using a mapping $g(i) = \tilde{\mathbf{v}}_i \in \mathbb{R}^2$ such that some measure of distortion is minimised. We focus on perhaps the most natural measure of distortion: the extent to which distances between pairs of points are maintained. For example, by requiring that the geodesic distances on the manifold are preserved, i.e. $d(i,j) \approx \|\tilde{\mathbf{v}}_i - \tilde{\mathbf{v}}_j\|$, where $d(i,j)$ is the geodesic distance between vertices $\mathbf{v}_i$ and $\mathbf{v}_j$. The extent to which an embedding $g$, satisfies this condition is measured by a quantity known as Stress-1 [26]:

$$\sigma_1(g) = \sqrt{\frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left(d(i,j) - \|g(i) - g(j)\|\right)^2}{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \|g(i) - g(j)\|^2}}. \tag{2.1}$$

The advantages of distance based measures are optimality with respect to a global distance distortion metric (e.g. Classical MDS is optimal with respect to strain [26]) and efficiency (fast marching can be used to compute geodesic distances and spectral methods to compute the embedding). We show that MDS based flattening extends naturally to the groupwise case and can be modified in a straightforward manner to handle topologically symmetric data. We use normalised aggregated distances over the dataset and hence the Classical MDS analysis computes a solution whose strain is optimal with respect to average normalised geodesic distance. Note that, depending on the requirements of the flattening, other objectives may be more appropriate. For example, consider a sample of faces in which 10% exhibited unusually long noses.

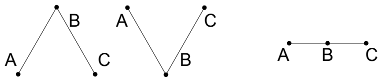

Figure 2.2: Averaging and geodesic distance computation do not commute: $d(A, C) = 2$ for both of the input shapes but $d(A, C) = 1$ for the mean shape.

If these noses were required to be flattened with low strain, geodesic distances over the nose should be weighted higher for the long-nosed samples. However, this would be at the expense of increasing the average case strain.

The drawback of embeddings based on preservation of geodesic distances is that they cannot be guaranteed to be bijective. This leads to the potential problem of triangle flips. The reason for this problem is that for certain surfaces a folded solution may provide better distance preservation. However, Zhou et al. [108] have shown how to overcome this problem by applying spectral clustering based on MDS. We extend their method to the groupwise case.

### 2.5.1 Naïve Approaches

Two naïve strategies for groupwise surface flattening turn out, in practice, to have drawbacks. Both are based on averaging.

The first is to compute the mean of the input set of meshes and then apply a traditional surface flattening algorithm to the resulting mean mesh. The problem with this approach is that the average mesh does not necessarily retain the properties of the input surfaces which the flattening seeks to preserve. For example, consider the two shapes shown on the left of Figure 2.2. $ABC$ forms an equilateral triangle with sides of length 1. Hence, the geodesic distance between A and C for both shapes is 2 and hence the average geodesic distance is 2. On the right, the mean of the two shapes is shown. In this case, the geodesic distance between A and C is 1. Clearly, a flattening which preserves average geodesic distances is not the same as flattening the average shape.

The second is to apply a surface flattening algorithm to each of the input meshes and average the resulting flattened coordinates for each vertex. This also does not yield optimal results with

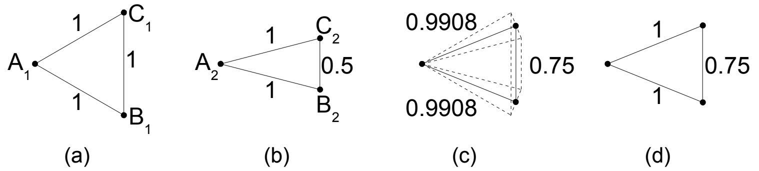

Figure 2.3: Averaging optimal parameterisations does not yield optimal results. (a) and (b) show two triangles flattened with zero distortion. (c) shows the Euclidian average of the two flattenings with $\sigma_1 = 0.1575$. (d) shows the optimal result with $\sigma_1 = 0.1562$.

respect to minimising total distortion. In the example shown in Figure 2.3, corresponding triangles in two meshes are flattened with zero stress-1 to coordinates: $A_1 = (0, 0)$, $B_1 = (\sqrt{0.75}, -0.5)$, $C_1 = (\sqrt{0.75}, 0.5)$ and $A_2 = (0, 0)$, $B_2 = (\sqrt{0.9375}, -0.25)$, $C_2 = (\sqrt{0.9375}, 0.25)$. Hence, the geodesic distance between A and B, and A and C, is 1 in both cases, whilst the distance between B and C is 1 for case 1 and 0.5 for case 2. The optimal solution in this case is an isosceles triangle with two sides of length 1 and one of length 0.75, shown in Figure 2.3(d). This yields a stress-1 value of 0.1562 for both cases. However, taking the Euclidian average of the coordinates of the original flattenings, as shown in Figure 2.3(c), yields a stress-1 value of 0.1575 for both cases. In other words, this approach introduces an unnecessary increase in the distortion of 0.83% for a mesh consisting of only a single triangle. An additional weakness of this approach is that an individual flattening must be computed for every mesh in the input set. This is potentially highly computationally intensive, particularly if there are a large number of input meshes. Nevertheless, in Section 2.8.1 we compare this approach with our groupwise result, using the ABF++ [91] method to compute the individual flattenings.

Next we present a more efficient and flexible approach to obtain a single class-optimal flattening.

### 2.5.2 Multidimensional Scaling

Consider a symmetric distance matrix $D$ in which element $D_{ij}$ is the squared geodesic distance between vertices $\mathbf{v}_i$ and $\mathbf{v}_j$, i.e. $D_{ij} = d^2(i, j)$. Geodesic distances can be computed efficiently on a triangle mesh using the Fast Marching algorithm of Kimmel and Sethian [53]. Classical

MDS converts these distances into equivalent dot products using the double centering formula [26]:

$$\tilde{D}_{ij} = -\frac{1}{2}\left(D_{ij} - \frac{1}{N}W_i - \frac{1}{N}W_j + \frac{1}{N^2}\sum_{k=1}^{N} W_k\right), \tag{2.2}$$

where $W_i$ is the $i$th row sum of $D$:

$$W_i = \sum_{j=1}^{N} D_{ij}. \tag{2.3}$$

The $p$th dimension of the embedding of the vertex $\mathbf{v}_i$ is then given by $e_{ip} = \sqrt{\lambda_p}\psi_{ip}$, where $\psi_{ip}$ is the $i$th element of the $p$th principal eigenvector of $\tilde{D}$ and $\lambda_p$ is the corresponding eigenvalue. Since we are interested in flattening the mesh to a 2D plane, we use the embedding $g(i) = (e_{i1}, e_{i2})$.

Zigelman et al. [109] were the first to propose the use of Classical MDS applied to geodesic distances for surface flattening. However, computing pairwise geodesic distances for every vertex pair and computing the spectral decomposition of the resulting distance matrix is impractical for meshes of a reasonable size. For this reason, they compute geodesic distances for only a small subset of the mesh vertices and apply MDS to this smaller distance matrix. A similar approach is taken by Peyré and Cohen [78] who use Local Linear Embedding on a subset of key-points.

### 2.5.3 MDS of Aggregated Distances

Our method is a groupwise extension of the method of Zigelman et al. [109]. However, we also make two refinements to the algorithm by tackling two questions that were not explored in the original paper. Namely, how to select the key-points subset and how to extend the embedding to the non key-point vertices.

Our set of input meshes gives rise to a set of corresponding distance matrices, differing according to the shape of each mesh. The common variations and individual differences within these distance matrices can be studied using the tools of Replicated MDS [69]. However, we wish to compute a single embedding whose strain is optimal with respect to average normalised geodesic distance. To achieve this, we apply Classical MDS to a matrix of aggregated geodesic distances. We weight the contribution of each mesh according to its scale, where we define the scale of the $f$th mesh as the norm of its shape vector, i.e. $\|\mathbf{S}_f\|$ (recall that input meshes are zero centred). The average normalised geodesic distance between two vertices is given by:

$$\overline{d}(i,j) = \frac{1}{F}\sum_{f=1}^{F}\frac{d_f(i,j)}{\|\mathbf{S}_f\|}, \tag{2.4}$$

where $d_f(i,j)$ is the geodesic distance between the $i$th and $j$th vertices of the $f$th mesh. Our distance matrix is given by $D_{ij} = \overline{d}^2(i,j)$. We apply double centering and compute the spectral decomposition of the resulting matrix. Note that $D$ is only computed for a subset of key-points selected from the vertices of $K^N$. We describe how to choose these key-points and how to extend the embedding to the whole mesh in the following sections.

### 2.5.4 Groupwise Uniform Mesh Sampling

We would like the key-points subset to sample the surfaces in the input set as evenly as possible. Peyré and Cohen [79] recently presented a greedy algorithm for efficient uniform sampling of a manifold. The idea is to choose a set of evenly spaced points on a triangle mesh by iteratively selecting the point furthest from those chosen so far (in terms of geodesic distances). For the groupwise case, we would like to choose a subset of the vertices of $K^N$, such that they are evenly spaced on average over all surfaces in the class.

We initialise by choosing a vertex $\{x_1\} \in K^N$, at random. We compute an average geodesic distance map, $U_1(i) = \overline{d}(x_1, i)$, from $\{x_1\}$ to all vertices $\{i\} \in K^N$. We then iteratively add key-points. Assume the set $S_{M-1} = \{\{x_1\}, \dots, \{x_{M-1}\}\}$ contains the first $M-1$ key-points and that $U_{M-1}$ is the geodesic map to $S_{M-1}$, computed over all meshes in the input class. The new point $\{x_M\}$ is the point furthest from $S_{M-1}$:

$$\{x_M\} = \operatorname*{arg\,max}_{\{i\}\in K^N} U_{M-1}(i). \tag{2.5}$$

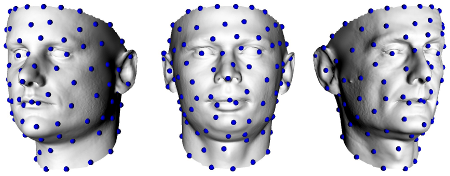

Figure 2.4: Groupwise uniform sampling: the key-points are uniformly distributed on average over all meshes in the input class.

The new geodesic distance map is simply, $U_M = \min(U_{M-1}, U_{x_M})$, where $U_{x_M}$ is the geodesic distance map to $\{x_M\}$. This can be computed efficiently by starting a front from $x_M$ using Fast Marching and restricting the solution to the set $\{i : U_{x_M}(i) \leq U_{M-1}(i)\}$.

An example of this groupwise uniform sampling is shown in Figure 2.4. The sampling is computed for 100 key-points over the 10 meshes of the BFM [76]. The sampling is plotted on three of the meshes.

### 2.5.5 Extending the Embedding

The application of MDS (as above) allows us to compute the embedding $g(x_m) \in \mathbb{R}^2$ for all $M$ key-points $\{x_m\} \in S_M$. To extend the embedding to the whole mesh, we need to interpolate the location $g(i)$ for all remaining $N - M$ vertices $\{i\} \in K^N \setminus S_M$. We do so using the eigenvector learning approach of Bengio et al. [10]. This method uses a kernel derived from the embedding of the key-points to approximate a Nyström integral.

Accordingly, the $p$th dimension of the embedding of the $i$th vertex is given by:

$$g_p(i) = \frac{\sqrt{\lambda_p}}{\lambda_p} \sum_{m=1}^{M} \psi_{mp} \tilde{K}(i, x_m), \tag{2.6}$$

where $\tilde{K}(a, b)$ is a kernel function. This kernel is obtained by using a continuous version of the double-centering formula:

$$\tilde{K}(a,b) = -\frac{1}{2}\left(\vec{d}^2(a,b) - E_x[\vec{d}^2(x,b)] - E_{x'}[\vec{d}^2(a,x')] + E_{x,x'}[\vec{d}^2(x,x')]\right), \tag{2.7}$$

where the expectations are computed over the key-points subset:

$$E_x[\overline{d}^2(x,b)] = \frac{1}{M}\sum_{m=1}^{M} \overline{d}^2(x_m, b), \tag{2.8}$$

$$E_{x'}[\overline{d}^2(a,x')] = \frac{1}{M}\sum_{m=1}^{M} \overline{d}^2(a, x_m), \tag{2.9}$$

$$E_{x,x'}[\overline{d}^2(x,x')] = \frac{1}{M^2}\sum_{m_1=1}^{M}\sum_{m_2=1}^{M} \overline{d}^2(x_{m_1}, x_{m_2}). \tag{2.10}$$

Extending the embedding to all vertices in the mesh requires the computation of $FMN$ geodesic distances, i.e. the geodesic distance from the key-points subset to all other vertices, averaged over all meshes in the class. In practice, this results in Fast Marching being applied $M$ times to each of the $F$ meshes and an overall time complexity of $O(FMN \log N)$ for the geodesic distance calculations.

With the embedding extended to all vertices, the vertex $\mathbf{v}_i^f$ maps to location $(g_1(i), g_2(i))$, for all $f$. In other words, the same vertex maps to the same texture coordinate for all meshes.

### 2.5.6 Symmetric Flattening

Obtaining a symmetric flattening for meshes with symmetric topology is straightforward. We must simply ensure that the distance matrix used in the MDS contains distances which respect symmetry relations. To enforce this, we begin by extending the key-points subset to include the symmetric partner of all key-points, i.e. $S_{2M} = S_M \cup \{\{\text{sym}(x_1)\}, \ldots, \{\text{sym}(x_M)\}\}$. If $S_M$ contains key-points lying on the line of symmetry, then $S_{2M}$ will have corresponding duplicate entries. Hence we ensure that all points in $S_M$ are away from the line of symmetry. We can now set the distance between a pair of key-points as the average of the distance between these points and the distance between the symmetric partners of these points:

$$D_{12} = \left(\frac{1}{2}\left(\overline{d}(x_1, x_2) + \overline{d}(\text{sym}(x_1), \text{sym}(x_2))\right)\right)^2. \tag{2.11}$$

Do the same for all other key-points pairs. Applying Classical MDS to this matrix will yield a symmetric flattening for the key-points. To ensure that the kernel used to extend the embedding to non key-point vertices yields symmetric results, we must also modify Equation 2.7 such that the distances are averaged with symmetric distances where appropriate.

### 2.5.7 Groupwise Charts

When the groupwise flattening obtained for a class of meshes results in large distortion or triangle flips, we can recursively partition the meshes into smaller charts. Each of these charts can be individually flattened using the same method as described above. This recursive partitioning can continue until a satisfactory level of distortion is reached or until no chart contains any triangle flips. This strategy is particularly useful when flattening a set of closed genus-0 meshes which must be cut in order to be topologically equivalent to discs.

In order to partition a set of meshes into class-optimal charts, we extend the single mesh spectral clustering approach of Zhou et al. [108] to operate on a set of meshes. Their algorithm proceeds by choosing vertices (landmark points), which correspond to the min-max values along each of the first $p$ most significant eigenvectors resulting from the MDS spectral decomposition. In our case, this is computed using the average normalised geodesic distances. These vertices are used to cluster the surface into $C$ charts by assigning every triangle to the closest landmark (in terms of the average normalised geodesic distance). Hence we obtain class-optimal landmark points and charts which are shared across the set of meshes. Landmark points which are too close together are discarded and hence $C \leq 2p$. The value of $p$ (hence the number of charts $C$) depends on the shape complexity of meshes [108]. This process can be iterated with each chart being subdivided further based on a spectral analysis of the segmented part. For the set of body meshes [45], we partition the bodies into $C = 30$ charts. For this data we use the $p = 12$ most significant eigenvectors to obtain the initial set of charts. We then split the charts into further sub-charts until the embedding results in no triangle flips. Figure 2.5 shows a body mesh partitioned into 30 charts.

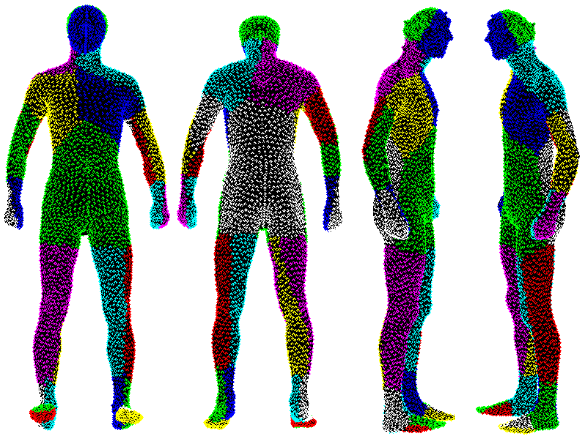

Figure 2.5: A body mesh partitioned into 30 charts. The various colours represent the vertices belonging to the respective charts.

### 2.5.8 Flattened Coordinates and Maps

We can finally obtain texture maps in UV space from the original per-vertex texture values using the computed flattening. We apply a scale $s$ and offset $o$ to the flattened coordinates such that $0 \leq s \cdot g_1(i) + o \leq 1$ and $0 \leq s \cdot g_2(i) + o \leq 1$ for all $\{i\} \in K^N$. We then compute a 2D colour texture map $\bar{\mathbf{T}}_f \in \mathbb{R}^{R \times R \times 3}$ for every mesh in the set. Each $R \times R$ texture map is obtained by regularly sampling the flattened texture function $Tex_f(g_1(i), g_2(i)) = \mathbf{t}_i^f$ using bicubic interpolation.

## 2.6 Groupwise Mesh Simplification

As seen in Section 2.4, each mesh within the input set is characterised by the same complex $K$, having the same number of vertices and the same connectivity, but varying intrinsic shape. Our groupwise mesh simplification algorithm is based on iterative edge contraction. Specifically, we use the Quadric Error Metric [36] to choose class-optimal edge collapses. Our framework follows that of Garland and Heckbert but acts on a set of meshes such that dense correspondence is preserved while the intrinsic shape of each mesh varies.

A pair contraction $(\mathbf{v}_1, \mathbf{v}_2) \rightarrow \bar{\mathbf{v}}$, transforms a pair of vertices $\mathbf{v}_1$ and $\mathbf{v}_2$ to a new position $\bar{\mathbf{v}}$, connects all their incident edges to $\mathbf{v}_1$ and deletes the vertex $\mathbf{v}_2$. Any edges or faces which became degenerate after the contraction are removed.

We extend this concept to a groupwise operation, which contracts a vertex pair over all meshes in a class:

$$(\mathbf{v}_i^f, \mathbf{v}_j^f) \to \bar{\mathbf{v}}_{ij}^f. \tag{2.12}$$

It is important to note that the same pair is contracted for all meshes in the class and hence the same edges and faces are removed. This means that the consistent topology established by the dense correspondence process is preserved through a contraction. Only the new position $\bar{\mathbf{v}}_{ij}^f$ varies, determined by the shape of each individual mesh.

Starting with the original high resolution meshes $\mathcal{M}_f^N = (K^N, \mathbf{S}_f)$, a sequence of pair contractions is applied until the simplification goals are satisfied. Each contraction corresponds to a local incremental modification of the complex $K^N$ and shape vectors $\mathbf{S}_f$. The algorithm generates a sequence of meshes $\mathcal{M}_f^N, \mathcal{M}_f^{N-1}, \mathcal{M}_f^{N-2}, \ldots$ with decreasing resolution. It should be noted that for the case $F = 1$ (a single mesh in a class), our formulation reduces to that proposed in [36]. For the purpose of mesh simplification we do not consider the texture vectors $\mathbf{T}_f$. Instead, in Section 2.7 we show how to compute texture coordinates for the low resolution meshes, to index into the high resolution flattened texture maps.

We consider only edge pairs as valid for contraction, i.e. where $\{i, j\} \in K^N$. We label the valid pairs at initialisation by constructing an edge table from the input triangle list. When we perform a contraction $(\mathbf{v}_i^f, \mathbf{v}_j^f) \to \bar{\mathbf{v}}_{ij}^f$, the complex $K^N$ is modified. Degenerate faces (those that no longer have 3 distinct vertices) and duplicate edges are removed as well as the collapsed edge and the redundant vertex $j$:

$$K^{N-1} = K^N \setminus \left\{\{j\}, \{i,j\}, \{j,\kappa\}, \{i,j,\kappa\} : \{i,j,\kappa\} \in K^N\right\}. \tag{2.13}$$

### 2.6.1 Quadric Error Metric

The quadric $Q$ assigns a value $Q(\mathbf{v})$ to every point in space $\mathbf{v}$ by the second order equation:

$$Q(\mathbf{v}) = \mathbf{v}^T \mathbf{A}\mathbf{v} + 2\mathbf{b}^T \mathbf{v} + c, \tag{2.14}$$

where $\mathbf{A}$ is a 3x3 matrix, $\mathbf{b}$ is a 3-vector and $c$ is a scalar [36].

When considering the contraction of a pair $(\mathbf{v}_i^f, \mathbf{v}_j^f)$, we need to determine the target position $\bar{\mathbf{v}}_{ij}^f$. We select the optimum position $(\bar{\mathbf{v}})$ as the one that minimizes Equation 2.14. Since Equation 2.14 is a quadratic, finding its minimum is a linear problem. Taking partial derivatives we obtain:

$$\nabla Q(\bar{\mathbf{v}}) = 2\mathbf{A}\bar{\mathbf{v}} + 2\mathbf{b}. \tag{2.15}$$

Solving for $\nabla Q(\bar{\mathbf{v}}) = 0$, we find the optimum position to be:

$$\bar{\mathbf{v}} = -\mathbf{A}^{-1}\mathbf{b}. \tag{2.16}$$

If $\mathbf{A}$ is singular, its inverse does not exist and we cannot solve for $\bar{\mathbf{v}}$ using Equation 2.16. In such a situation we select either end point $\mathbf{v}_i$ or $\mathbf{v}_j$ as $\bar{\mathbf{v}}$, depending on which one of these produces the lowest $Q(\bar{\mathbf{v}})$. It should be noted we must compute the target positions $\bar{\mathbf{v}}_{ij}^f$ for each edge $\{i, j\}$ across the whole set of $F$ meshes. The edge quadric used for computing each target placement is obtained by the addition rule [36]. We must also incorporate a boundary penalty into the initial quadrics for vertices lying on a discontinuity edge [36].

With the target position for each pair $(\mathbf{v}_i^f, \mathbf{v}_j^f)$ computed, the last remaining step is the selection of the edge pair for contraction. Following the strategy used for simplifying deforming or articulated meshes [73], we select that pair to be the one which minimises Equation 2.14 over the whole set of meshes. After the removal of $L$ vertices this is given by:

$$\{i^*, j^*\} = \underset{\{i,j\} \in K^{N-L}}{\arg\min} \sum_{f=1}^{F} Q_{i+j}^f(\bar{\mathbf{v}}_{ij}^f). \tag{2.17}$$

After every edge contraction we must update the quadrics and the error heap for the modified entries in the complex $K$ [36]. Our method adds an overhead to the computational complexity of the QSlim method for a single mesh. A single edge contraction requires, $O(F)$ operations to recompute the quadric error in the neighborhood of the contracted edge, plus $O(\log N)$ operations to maintain the heap property. As the number of contractions is bounded by $N$, the iterative contraction phase runs in $O(FN + N\log N)$ time.

### 2.6.2 Symmetric Mesh Simplification

Golovinskiy et al. [41] modified Garland and Heckbert's QSlim algorithm to operate in a symmetric manner. Our groupwise simplification algorithm can be similarly modified to ensure that all simplified meshes preserve the symmetric mesh topology. In order to achieve this, we modify Equation 2.17, to select symmetric edge pairs for contraction as follows:

$$\{i^*, j^*\}, \{\mathrm{sym}(i^*), \mathrm{sym}(j^*)\} = \underset{\{i,j\},\{\mathrm{sym}(i),\mathrm{sym}(j)\}\in K^{N-L}}{\arg\min} \left( \sum_{f=1}^{F} Q^f_{i+j}(\bar{\mathbf{v}}^f_{ij}) + \sum_{f=1}^{F} Q^f_{\mathrm{sym}(i)+\mathrm{sym}(j)}(\bar{\mathbf{v}}^f_{\mathrm{sym}(i)\mathrm{sym}(j)}) \right). \tag{2.18}$$

We contract both the edge $\{i^*, j^*\}$ and its symmetric partner $\{\mathrm{sym}(i^*), \mathrm{sym}(j^*)\}$ as normal. This symmetric version of the algorithm can be implemented efficiently using a symmetric modification of the edge table. Each entry contains symmetric edge pairs and priority is determined based on the sum of their Quadric Error Metric scores. It should be noted that in the symmetric implementation, during each contraction, $L$ (the number of vertices removed) is incremented by 2. However, in the case where $\{i^*, j^*\} = \{\mathrm{sym}(i^*), \mathrm{sym}(j^*)\}$, only one edge is collapsed and hence one vertex is removed during the contraction.

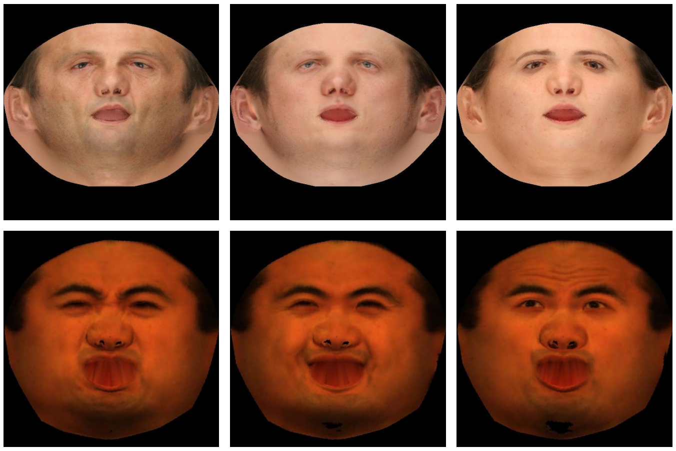

Figure 2.6: Groupwise surface flattening results. The top row shows 3 faces from the BFM and the bottom row shows 3 frames from the Spacetime dataset.

### 2.6.3 Preventing Mesh Inversion

While performing an edge pair contraction, we must ensure that the orientation of the faces is preserved (i.e. none of the faces affected by a contraction fold over). Following [36], we compare the normals of all the affected faces before and after contraction. If any of the normals flip, we disallow that contraction. Unlike [36], we do not impose a penalty for mesh inversion, we simply disallow it. This strategy is important to produce higher quality meshes, especially at lower resolutions.

For the groupwise case, we disallow a contraction if it results in a flipped normal for any one of the $F$ meshes in the input set. Note that because the shape of each mesh varies, flips can occur in some meshes and not others. Disallowing mesh inversions restricts the number of vertices that can be removed and effectively provides a stopping criteria, i.e. the point at which no edge can be contracted without causing a mesh inversion in at least one of the meshes in the set. The level of simplification that can be obtained before this happens is related to the variability and hence the number of meshes in the input set. This is a difference between groupwise and individual mesh simplification and shows that the groupwise case is more heavily constrained.

| **Subject** | **MDS [109]** | | **MDS Groupwise** | | **ABF++ [91]** | | **ABF++ Groupwise** | |
|---|---|---|---|---|---|---|---|---|
| | Stress-1 | Stretch | Stress-1 | Stretch | Stress-1 | Stretch | Stress-1 | Stretch |
| 1 | 0.1217 | 1.942 | 0.1279 | 1.474 | 0.2620 | 1.031 | 0.2525 | 2.177 |
| 2 | 0.1022 | 1.951 | 0.1049 | 1.541 | 0.2251 | 1.019 | 0.2401 | 2.213 |
| 3 | 0.1036 | 1.430 | 0.1099 | 1.517 | 0.2264 | 1.016 | 0.2432 | 2.288 |
| 4 | 0.1146 | 1.634 | 0.1176 | 1.558 | 0.2500 | 1.020 | 0.2437 | 2.219 |
| 5 | 0.1317 | 1.600 | 0.1346 | 1.552 | 0.2735 | 1.036 | 0.2619 | 2.216 |
| 6 | 0.1093 | 3.167 | 0.1209 | 1.562 | 0.2560 | 1.032 | 0.2501 | 2.294 |
| 7 | 0.1217 | 4.732 | 0.1233 | 1.572 | 0.2755 | 1.029 | 0.2506 | 2.271 |
| 8 | 0.1254 | 1.597 | 0.1236 | 1.491 | 0.2710 | 1.009 | 0.2593 | 2.192 |
| 9 | 0.1256 | 1.567 | 0.1333 | 1.493 | 0.2494 | 1.009 | 0.2423 | 2.201 |
| 10 | 0.1080 | 1.661 | 0.1091 | 1.528 | 0.2538 | 1.026 | 0.2512 | 2.262 |
| **Mean** | **0.1164** | **2.128** | **0.1205** | **1.529** | **0.2543** | **1.023** | **0.2495** | **2.233** |

Table 2.1: Surface flattening results for the BFM data.

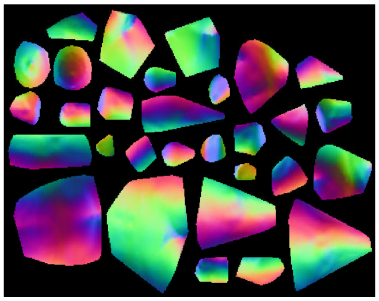

Figure 2.7: The groupwise chart atlas obtained for one of the body meshes.

## 2.7 Low Resolution Texture Mapping

In order to texture map simplified meshes using the original high resolution flattened textures, we must compute the texture coordinates of the simplified vertices. Once again, this must be done in a groupwise manner such that we establish a single mapping from vertex indices at the lower resolution to the texture space. Following [22], we do this by associating each low resolution vertex to the closest vertex in the original high resolution mesh, looking up the texture coordinates for this high resolution vertex and then averaging over all meshes in the class. This step runs in $O(FN(N-L))$ time.

Consider a vertex $\hat{\mathbf{v}}_i^f$ in a simplified mesh which has had $L$ vertices removed and where $\{i\} \in K^{N-L}$. We define $h(\hat{\mathbf{v}}_i^f)$ as the function which returns the index of the vertex $\mathbf{v}_j^f$ which

| **Subject** | Single Mesh [108] | Groupwise |
|---|---|---|
| 1 | 0.0313 | 0.0410 |
| 2 | 0.0326 | 0.0403 |
| 3 | 0.0335 | 0.0484 |
| 4 | 0.0336 | 0.0428 |
| 5 | 0.0346 | 0.0509 |
| 6 | 0.0339 | 0.0530 |
| 7 | 0.0311 | 0.0443 |
| 8 | 0.0330 | 0.0481 |
| 9 | 0.0314 | 0.0438 |
| 10 | 0.0320 | 0.0391 |
| **Mean** | **0.0327** | **0.0452** |

Table 2.2: Surface flattening results for the body meshes. We report stress-1 values averaged over the 30 charts in each case.

is closest to $\hat{\mathbf{v}}_i^f$ and where $\{j\} \in K^N$:

$$h(\hat{\mathbf{v}}_i^f) = \underset{\{j\} \in K^N}{\arg\min} \|\hat{\mathbf{v}}_i^f - \mathbf{v}_j^f\|. \tag{2.19}$$

Recall that $g$ is the embedding of the vertices at the highest resolution $N$. The embedding of a vertex from a simplicial complex with a reduced number of vertices is given by:

$$\hat{g}(i) = \frac{1}{F} \sum_{f=1}^{F} g\left(h(\hat{\mathbf{v}}_i^f)\right). \tag{2.20}$$

When the meshes have been flattened into charts, we assign each triangle in the low resolution mesh to one chart (at the highest resolution). The optimal chart is chosen as the one which is closest to the triangle in terms of Euclidian distance (on average over all the meshes in the set). Low resolution texture coordinates are computed by using Equations 2.19 and 2.20, with the constraint that the solution is restricted to the entities associated with the corresponding charts.

## 2.8 Experimental Results

We now present experimental results of applying our groupwise surface flattening and simplification to sets of meshes and using the flattened textures to texture map the simplified meshes.

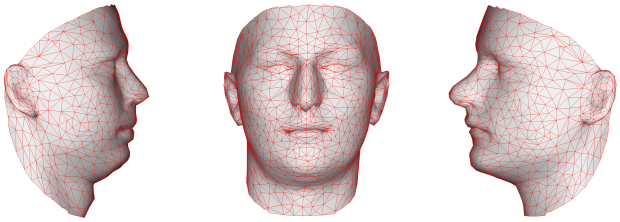

Figure 2.8: Simplification results for 3 of the faces from the BFM (at 1000 vertices).

We experiment with three datasets. The first consists of 10 faces from the BFM [76] which were obtained using a three view structured light scanner and are set into dense correspondence using a modified version of the algorithm of Amberg et al. [6]. These meshes are topologically symmetric. The second is the Spacetime faces [104] motion sequence which comprises 385 meshes of a single subject acquired using spacetime stereo. Vertex motion is tracked using optical flow and the mouth is unconnected (i.e. it contains a hole). For the Spacetime dataset, we use a subset of 96 meshes (every 4th frame) for our experiments. Finally, the third dataset consists of 550 full body meshes of 114 subjects [45]. The meshes are captured using a Vitronic laser scanner and are also set into dense correspondence using an approach similar to [6]. We use a subset of 10 bodies (different subjects in the same pose, i.e. pose 0) for our experiments. The two face datasets contain meshes which are topologically equivalent to a disc, whereas the body dataset contains genus-0 closed surfaces.

### 2.8.1 Surface Flattening

In Figure 2.6 we show examples of applying groupwise surface flattening to the two face datasets. We show three samples from each dataset, flattened using a single flattening calculated over the whole set. In both cases, the flattening was computed using 200 key-points which were obtained using groupwise uniform sampling. For the BFM data, 200 additional symmetric key-points were also used. The BFM flattening is perfectly symmetric whilst the Spacetime faces data is close to symmetric since the averaging of geodesic distances over varying expressions averages out asymmetries. Note that the boundary vertices are free to assume any position. The approximately elliptical and circular boundaries are simply the optimal MDS solution for

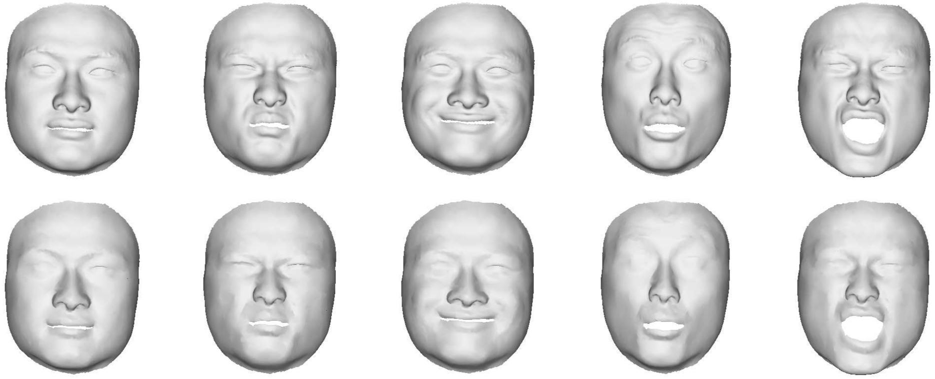

Figure 2.9: Simplification results for 5 frames from the Spacetime dataset. The original sequence (23728 vertices) is shown in the top row and the bottom row shows the sequence simplified to 2000 vertices.

| **Subject** | **Single Mesh [36]** | | **Groupwise** | |
|---|---|---|---|---|
| | Mean | RMS | Mean | RMS |
| 1 | 0.0923 | 0.130 | 0.106 | 0.148 |
| 2 | 0.106 | 0.146 | 0.121 | 0.169 |
| 3 | 0.129 | 0.176 | 0.148 | 0.204 |
| 4 | 0.11 | 0.155 | 0.126 | 0.176 |
| 5 | 0.104 | 0.144 | 0.118 | 0.165 |
| 6 | 0.119 | 0.165 | 0.132 | 0.182 |
| 7 | 0.103 | 0.141 | 0.115 | 0.158 |
| 8 | 0.111 | 0.154 | 0.132 | 0.183 |
| 9 | 0.093 | 0.131 | 0.109 | 0.151 |
| 10 | 0.092 | 0.126 | 0.104 | 0.145 |
| **Mean** | **0.106** | **0.147** | **0.121** | **0.168** |

Table 2.3: Surface simplification results for the BFM data.

the respective datasets.

Quantitative results for the BFM data are shown in Table 2.1. We use two metrics to assess the quality of the flattening: stress-1 (as described in Equation 2.1) and mean $L_2$ stretch [89]. The stress-1 metric evaluates global distortion of geodesic distances. Since it requires computing pairwise distances between all points, we compute this only over the 400 key-points. The stretch metric evaluates local distortion by measuring how a unit circle would be non-uniformly scaled at each triangle. Stretch values are computed using the tool Graphite [43].

We provide results for 4 approaches. The first set of results are obtained by applying the MDS based flattening method of Zigelman et al. [109] to each mesh individually. Note that we select key-points and extend the embedding to non key-points vertices using the techniques

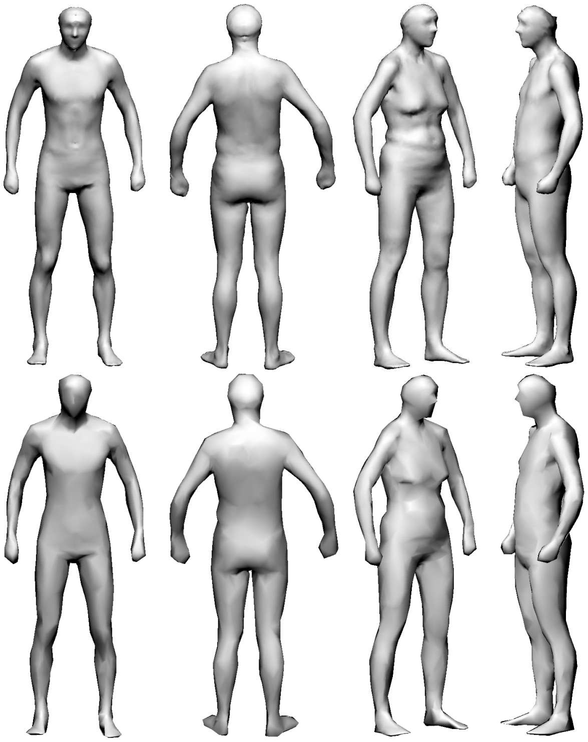

Figure 2.10: Simplification results for the body meshes. The top row shows 4 bodies at the highest resolution (6449 vertices) and the bottom row shows the simplified meshes (1000 vertices).

described in Section 2.5. This method provides optimal results for each mesh with respect to strain and is likely to be close-to-optimal for the related measure of stress-1. The second set of results are for our groupwise extension. Here, a single flattening is computed from all of the input meshes. We then measure the distortion induced by applying this flattening to each mesh. The third set of results are obtained by applying the Fast and Robust Angle Based Flattening (ABF++) method of Sheffer et al. [91] individually to each mesh. This approach yields very low stretch values and is included to provide a baseline comparison to the MDS based results. Since this is the first work to consider groupwise flattening, we have no results against which to directly compare. Instead, we use the ABF++ results to produce a naïve groupwise flattening (as described in Section 2.5.1). We compute the average embedding produced by ABF++ and measure the distortion induced by applying this flattening to each of the input meshes. Note that we also computed individual and naïve groupwise flattenings using the Hierarchical Least

| **Subject** | **Single Mesh [36]** | | **Groupwise** | |
|---|---|---|---|---|
| | Mean | RMS | Mean | RMS |
| 1 | 0.7450 | 0.9620 | 1.1435 | 1.5650 |
| 2 | 0.7585 | 0.9700 | 1.1750 | 1.6090 |
| 3 | 0.7650 | 0.9935 | 1.1725 | 1.5985 |
| 4 | 0.7720 | 0.9955 | 1.1800 | 1.6155 |
| 5 | 0.7660 | 0.9890 | 1.2135 | 1.6515 |
| 6 | 0.7310 | 0.9460 | 1.1575 | 1.5900 |
| 7 | 0.7695 | 1.0025 | 1.1595 | 1.5895 |
| 8 | 0.8255 | 1.0620 | 1.2710 | 1.7380 |
| 9 | 0.7530 | 0.9660 | 1.1140 | 1.5300 |
| 10 | 0.7605 | 0.9810 | 1.1405 | 1.5530 |
| **Mean** | **0.7646** | **0.9868** | **1.1727** | **1.6040** |

Table 2.4: Surface simplification results for the body meshes.

Squares Conformal Mapping method of Ray and Lévy [83] and the Mean Value Coordinate method of Floater [33]. These methods performed worse than ABF++ so we do not include them here.

The best individual flattenings are given by MDS (with respect to stress-1) and ABF++ (with respect to stretch). Our groupwise modification to MDS induces only a modest mean increase in stress (3.5%) and a reduction in stretch (compared to the MDS based individual flattenings). On the other hand, the naïve groupwise approach given by averaging the ABF++ embeddings results in a large increase in stretch (by a factor of two) whilst the stress-1 remains approximately equal to the ABF++ individual flattenings. Moreover, the tabulated results demonstrate that our groupwise modification to MDS yields a single flattening which is quantitatively superior to the naïve approach whilst not being significantly worse than individual flattenings.

In Figure 2.7 we show an example of applying groupwise spectral clustering and flattening to the body data. The meshes are partitioned into 30 charts and then flattened to obtain the chart atlas. We show the flattened bump map which is computed from the corresponding vertex normals. The body data has a relatively small number of vertices (compared to the face datasets). Hence the spectral analysis was performed using all the vertices (i.e. we do not subselect key-points) . Table 2.2 provides the stress-1 values obtained for the 10 body

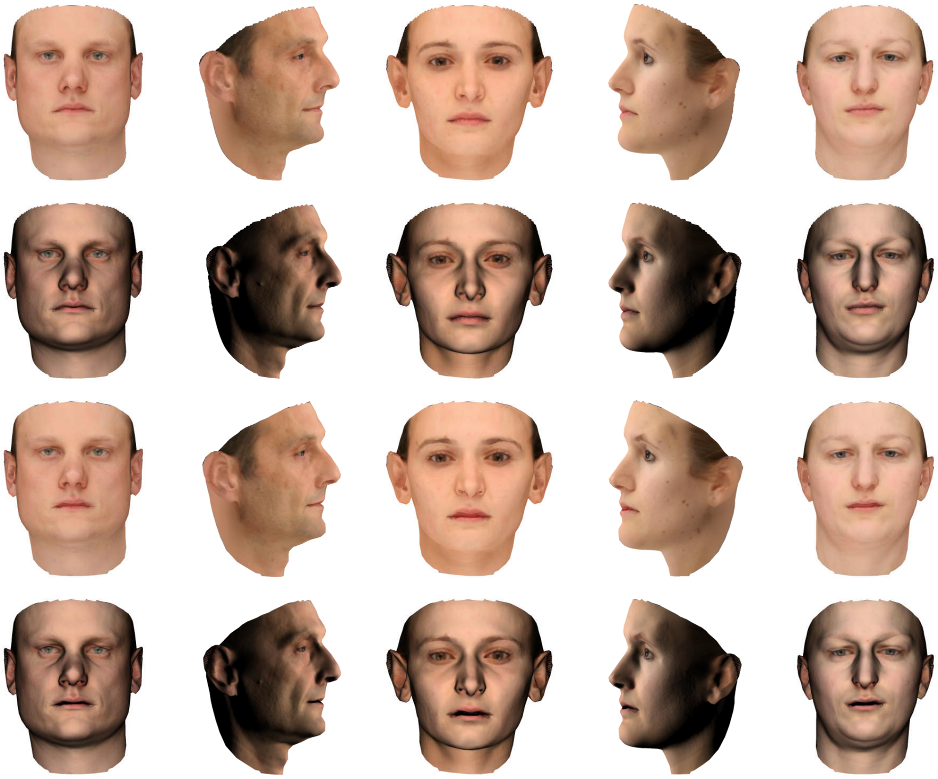

Figure 2.11: Highest resolution (53490 vertices) meshes from the BFM rendered with texture maps (first row) and texture plus bump maps (second row). Simplified meshes (1000 vertices) rendered with texture maps (third row) and texture plus bump maps (fourth row).

meshes used in our experiment. We also compare our groupwise chart atlas, with the chart atlas obtained for each individual body mesh (i.e. the method of [108]). We use the same chart partitioning to compare the stress-1 values resulting from the two methods.

### 2.8.2 Surface Simplification

In Figure 2.8 we show simplification results on the BFM data. The original data is topologically symmetric and this is preserved in the simplification process. In Figure 2.9 we show simplification results for the Spacetime faces data. For 5 frames containing different expressions we show the original (23728 vertices) and simplified (2000 vertices) sequences. Finally, Figure 2.10 shows the simplification results for the body meshes. Table 2.3 shows the mean and RMS distances between the highest resolution meshes (53490 vertices) and the simplified (1000 vertices) meshes

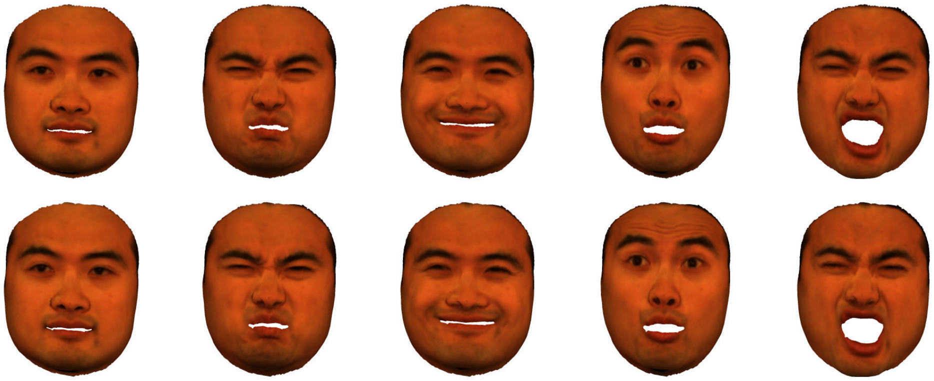

Figure 2.12: Highest resolution (23728 vertices) meshes from the Spacetime dataset rendered with texture maps (first row) and simplified meshes (2000 vertices) rendered with texture maps (second row).

for the BFM data. These distances are computed using the tool Metro [23]. We report the average of the forward and backward distances [23]. As expected the groupwise modification of the QSlim algorithm induces a modest increase in the mean and RMS distances. Similar results for the body data is shown in Table 2.4. In this case the highest resolution is 6449 vertices and the simplified resolution is 1000 vertices. All the reported distances are in mm.

### 2.8.3 Texture Mapping

In Figure 2.11 we show results of mapping the highest resolution flattened textures/normals onto the highest resolution (53490 vertices) and simplified meshes (1000 vertices). The first two rows show the highest resolution BFM data mapped with texture values (top row) and texture plus vertex normal values (second row). The bottom two rows show similar results for the low resolution BFM data. Note that the perceptual difference at the two resolutions is negligible. Similar results are shown for the Spacetime faces data in Figure 2.12. In Figure 2.13 we show three low resolution BFM faces texture mapped with a checkerboard pattern. This highlights both the smooth nature of the surface flattening, with little distortion to the pattern and the stability of the texture mapping between subjects. Finally, in Figure 2.14 we show bump mapping results for the body meshes. Each body is bump mapped using the respective chart atlas.

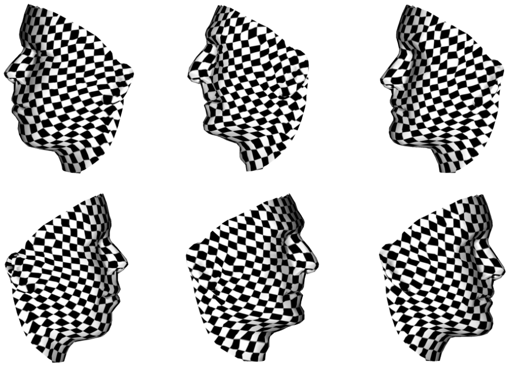

Figure 2.13: Three BFM faces (rows) at a resolution of 1000 vertices texture mapped with a checkerboard pattern.

### 2.8.4 Application: Statistical Modelling

Finally, we demonstrate using the output of our approach for the purposes of training a (low resolution) 3D morphable model. We will take a detailed look at the process of building a morphable model in Chapetr 3. For now in order to demonstrate the application of the proposed techniques to statistical modelling we will refer to the pioneering morphable model of Blanz and Vetter [14]. In [14], Principal Components Analysis (PCA) was applied to the high resolution shape and texture vectors, $\mathbf{S}_f$ and $\mathbf{T}_f$. This yields a statistical model comprising an average shape/texture vector and the principal modes of shape/texture variation. The first mode of variation of such a model can be seen in the top row of Figure 2.15. On the left we show texture mapped surfaces demonstrating texture and shape variation jointly, whilst on the right we show shape variation only.

We contrast this with applying PCA to the simplified shape vectors, $\hat{\mathbf{S}}_f \in \mathbb{R}^{3(N-L)}$, and the flattened high resolution texture maps, $\bar{\mathbf{T}}_f \in \mathbb{R}^{R\times R\times 3}$. Once again, this yields an average shape vector and principal modes of shape variation and an average texture map and principal modes of texture variation. Instead of learning per-vertex texture variation, we learn variation in the high resolution texture maps which can be mapped to the low resolution meshes using the embedding $\hat{g}$. The first mode of variation of this model is shown in the second row of Figure 2.15. Modes of variation captured by the two models are very similar and the texture mapped

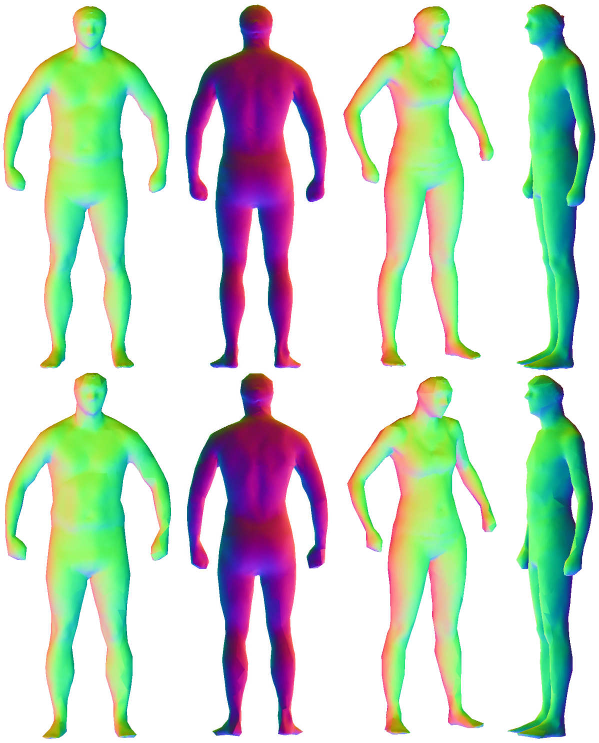

Figure 2.14: Highest resolution (6449 vertices) bodies rendered with bump maps (first row) and simplified bodies (1000 vertices) rendered with bump maps (second row).

versions are almost indistinguishable. This means that a morphable model with over 50 times less vertices can be used to synthesise face appearance of a similar quality. For applications such as fitting a morphable model to images, the lower resolution model will vastly reduce computational complexity.

## 2.9 Conclusions

We have shown how a set of meshes (possibly symmetric) which are in dense correspondence can be flattened and simplified in a groupwise manner which preserves correspondence. Groupwise flattening is performed by applying Classical MDS to a aggregated distance matrix. Where appropriate, groupwise spectral analysis can be used to partition the meshes into class-optimal charts. Groupwise simplification is performed by modifying an iterative edge collapse algorithm to make class-optimal edge collapses at each iteration. The techniques have application in motion compression (we are able to reduce the number of vertices in the Spacetime data by a

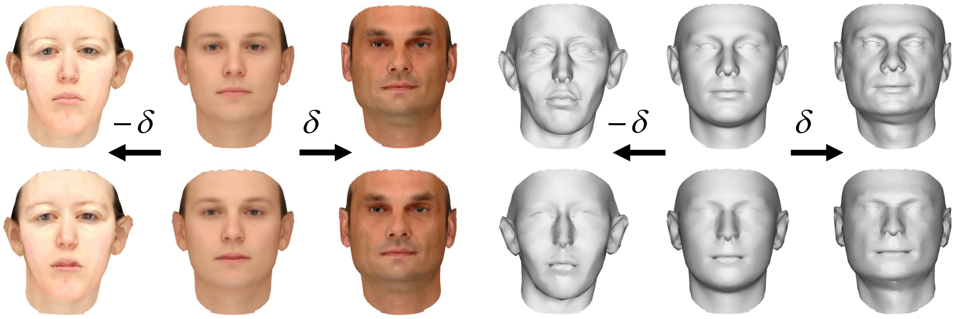

Figure 2.15: Shows the deviations (around the mean) in the most significant parametric mode for the texture and shape statistical models. The top row corresponds to models built at the highest resolution (53490 vertices) and the bottom row corresponds to models built at resolution of 1000 vertices. $\delta$ is the standard deviation corresponding to the most significant parametric mode.

factor of 10 with no appreciable difference in rendered results), in training a (low resolution) 3D morphable model and in the automated construction of facial rigs.

# Chapter 3

# 3D Morphable Models Revisited

## 3.1 Introduction and Related Work

Linear statistical models have been used to model variation in 2D [25] and 3D [54] shape, appearance [24] and texture [15]. These models are generative in nature, in the sense that instances similar to those used to train the model can be computed from a low dimensional parameter vector. Faces have proven a particularly suitable class to model using such approaches. The most common method for learning such models from data, Principal Components Analysis (PCA), is based on the assumption that faces form a Gaussian cloud in a high dimensional space. The principal axes of this cloud are estimated from a training sample, allowing any face to be approximated in terms of a small number of parameters.

In this chapter we revisit the process of constructing a 3D morphable model from training data. We reference back to the pioneering work of Blanz and Vetter [14, 15], provide alternative means of establishing dense correspondences between a set of 3D meshes and subsequently using those meshes to build a 3D morphable model. We also reformulate the probabilistic prior that the model provides on the distribution of parameter vector lengths and provide an illustrative example (by fitting the model to sparse data) of how this prior can be used to control the trade-off between the fitting quality and shape plausibility.

## 3.2 Contributions

First, we provide a new framework for constructing a 3D morphable model from a training set of facial meshes. By making use of techniques from the statistical shape analysis literature, we show how to construct a morphable model whose captured variance is of greater utility, in the sense that the generalisation error (i.e. average error when representing out of sample surfaces) is lower for both a fixed number of model dimensions retained or a fixed percentage of total variance captured. Second, we show that the distribution of parameter vector lengths follows a chi-square distribution and discuss how the parameters of this distribution can be used as a regularisation constraint on the length of parameter vectors. Finally, we use our improved model and statistical prior in the setting of fitting a dense 3D morphable model to sparse 2D feature points. We verify empirically that our analytical prediction of the parameter vector length constraint coincides with the optimum operating point of our algorithm.

## 3.3 Morphable model construction

The process of constructing a morphable model is divided into three stages: 1. establishing a dense correspondence, 2. shape alignment and 3. statistical modelling. In each case, we outline previous methods before describing our approach. We begin by describing how our approach allows us to construct a morphable model as a shape space.

### 3.3.1 Morphable Models as Shape Spaces

The 3D morphable face model of Blanz and Vetter [15] captures the class-specific properties of faces by finding a low dimensional parameterisation of 3D face shape and texture. The model is learnt from a sample of high resolution 3D face scans. A common interpretation of the meaning of the *shape* of an object is Kendall's [29] notion that shape is the geometrical information that remains after the effects of location, scale and rotation have been removed. There is a comprehensive toolbox of techniques available for the statistical analysis of shape using data which is provided in terms of coordinates of named point locations or landmarks. Namely, these are Kendall's shape space and the application of linear multivariate statistics in the tangent space at the Procrustes average [29]. This group of techniques has become to be known as geometric morphometrics. A *landmark* is a hypothesis of equivalence under a particular measure of similarity, e.g. anatomy, topology or function. In effect, the implied meaning of a landmark point is, in some sense, the same across the whole population.

However, the statistical analysis of continuous curves or surfaces (which contain only relatively sparse salient points) is not so well developed. For example, only a relatively small proportion of the face surface contains salient points which may be identified with good repeatability across all faces. The remainder of the face comprises large areas of smoothly shaded, textureless surface. It is therefore not obvious how a landmark based statistical approach can be applied to model the variations in the face surface.

The morphable model of Blanz and Vetter [15] described above is based on transforming a set of face surfaces into a vector space such that any convex combination of members of the training set results in a viable new face. However, their model is not a shape space. They only coarsely remove the effects of rotation, translation and scale before the dense correspondence between samples is known. In other words, they ultimately treat every vertex in the model as a landmark but do not remove pose effects with respect to these landmarks.

We propose an alternative approach for constructing a 3D morphable model using Kendall's notion of shape space. Our work closely follows the semilandmark approach of Bookstein [18]. The key idea is to compute correspondences for deficient regions (i.e. those lacking landmark points) using the part of the data that is not deficient. This is done in a principled manner by minimising a physically motivated bending energy of the data about its average. We use Procrustes analysis to obtain pose free shape vectors. This combination of techniques allows us to construct a dense 3D morphable model as a shape space.

### 3.3.2 Finding Dense Correspondences

Blanz and Vetter's [15] approach is to effectively treat every vertex of a face mesh as a distinct landmark point. This is possible because a modified optical flow algorithm is used to find dense correspondences between all samples in the training set. These correspondences are based on matching regions with similar colour and topography to a reference face and subsequently resampling every face in a consistent manner.

The advantage of their approach is that a model may be constructed automatically with little manual intervention. However, the similarity measure used to find corresponding points between faces relies on an *ad hoc* formulation of local surface features, such as 3D position, texture, local curvature and the surface normal. It is unclear which features should be chosen and how their relative importance should be weighted. Moreover, the utility of different features will vary spatially and between samples. For example, when registering a sample with a beard to one without, texture is an unreliable feature to use. The second problem is that large areas of the face contain no salient structures, either in the texture or shape domain. For example, the forehead and cheeks. In these regions the calculated flow field is noisy and unreliable. Blanz and Vetter [15] overcome this problem by smoothing and interpolating the flow fields. Finally, the choice of reference face will affect the quality of detected correspondences and ultimately the final model.

At the expense of introducing some manual intervention, we suggest an alternative approach which offers potentially more stable performance. Because our method does not require the selection of a reference face, only one possible model can be constructed from a given set of training data.

We commence with a set of face surfaces obtained by a Cyberware 3030PS laser range scanner. These surfaces are parameterised in cylindrical coordinates. This provides a convenient representation of the facial manifold in 2 dimensions, $(u, v)$. A set of sparse 2D landmark points are manually identified on the parameterisation of each face surface. The landmark points are chosen such that they can be reliably located on all training samples. With these sparse, but reliable, correspondences in hand, the mean coordinates of each landmark point are found. The $x$, $y$ and $z$ coordinates of each vertex can be expressed as a function in $(u, v)$ space, e.g. $x(u, v)$. Similarly for each colour channel in the texture map. We warp the landmark points of each sample to the mean landmarks. We interpolate this warp using a physically motivated bending energy, through the application of a thin-plate spline warp [17]. Finally, we resample the vertex

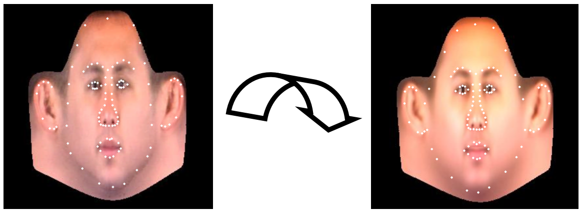

Figure 3.1: Shows the correspondence of scans based on the principle of thin-plate splines. Using two sets of 2D points (white dots), a novel scan (left) is warped to the mean scan (right) using the thin-plate spline function $-U(r) = -r^2 \log r^2$.

coordinate functions in a consistent manner across all faces. The result is that a point $(u, v)$ corresponds to the same point on each face in the training set, i.e. we have established a dense correspondence. This process is demonstrated in Figure 3.1.

In drawing a comparison between our approach and Blanz and Vetter's [15] optical flow algorithm: both methods begin with a sparse set of correspondences (ours manually landmarked, theirs the set of automatically detected correspondences that are considered reliable) and both interpolate the remainder. Construction of the morphable model is done offline prior to its use in an application such as face shape recovery. Hence the manual processing required by our methods is an acceptable burden if it results in more accurate correspondences.

### 3.3.3 Shape Alignment

In Blanz and Vetter's [15] morphable model, shape alignment is treated as a preprocessing step. The raw face meshes are marked with a small number of feature points and a 3D-3D transform is used to align each face to a reference face. In other words, when this alignment takes place, the dense correspondence between faces is unknown and the scale, translation and rotation necessary to register each face to the reference is only a very coarse approximation. Further, as with computing the dense correspondences, the choice of reference face will affect the final model. We propose instead to use Procrustes analysis as a rigorous means to remove pose effects without having to choose a reference face (the reference face is instead the Procrustes mean which is iteratively updated).

With our sample of faces in dense correspondence (forming a vector space) we can proceed with shape alignment using the standard tools of statistical shape analysis. The idea here is to remove any effects of scale, rotation and translation to obtain a pure shape model that captures only variation in identity. The $i$th face is represented by the shape vector $\mathbf{x}_i = (x_1, y_1, z_1, \ldots, x_p, y_p, z_p)^T \in \mathbb{R}^{3p}$, that contains the $x$,$y$,$z$ coordinates of its $p$ vertices.

Our aim is to transform the shape vectors into a shape space. We do this by aligning the shape vectors to a common coordinate frame using generalised Procrustes analysis. This is an iterative procedure which alternates between aligning all samples to the current estimate of the mean shape and then re-estimating the mean from the aligned vectors. These two steps are iterated until convergence.

Our mean shape estimate (for $m$ face scans) is the Procrustes mean:

$$\mathbf{x}_0 = \frac{1}{m} \sum_{i=1}^{m} \mathbf{x}_i. \tag{3.1}$$

In order to maintain a constant scale for the model, we fix the length of the mean shape at each iteration:

$$\bar{\mathbf{x}} = \frac{\mathbf{x}_0}{\|\mathbf{x}_0\|}. \tag{3.2}$$

All samples are aligned to the current estimate of the mean shape using a 3D similarity transform, $T_r(\mathbf{x}_i, \gamma_i) = (x'_1, y'_1, z'_1, \ldots, x'_p, y'_p, z'_p)^T$, where:

$$\begin{pmatrix} x'_j \\ y'_j \\ z'_j \end{pmatrix} = s\mathbf{R} \begin{pmatrix} x_j \\ y_j \\ z_j \end{pmatrix} + \mathbf{t}. \tag{3.3}$$

Here, $j \in [1 \ldots p]$, $\mathbf{R} \in \mathrm{SO}(3)$ is a rotation matrix, $s \in \mathbb{R}$ is a scaling and $\mathbf{t} \in \mathbb{R}^3$ is a translation. The optimal pose parameters $\gamma_i = (\mathbf{R}, s, \mathbf{t})$ which map a sample onto the mean shape vector are found by using Horn's method [48] to solve:

$$\gamma_i = \arg \min_{\gamma} \|T_r(\mathbf{x}_i, \gamma) - \bar{\mathbf{x}}\|^2. \tag{3.4}$$

A summary of the steps involved in generalised Procrustes analysis is as follows:

1. Find the Euclidian mean of the face shape vectors (Equation 3.1).
2. Rescale the mean shape vector to unit length (Equation 3.2).
3. Find the optimal pose parameters, $\gamma_i$, to align each shape vector, $\mathbf{x}_i$, to the mean.
4. Set $\mathbf{x}_i = T_r(\mathbf{x}_i, \gamma_i)$.
5. If change in estimate of mean indicates convergence, stop. Otherwise iterate to step 1.

This process converges very rapidly (typically within 3 iterations).

Since scale has been removed from the shape vectors, they all lie on the surface of a curved manifold in shape space (since we set the model scale to 1, the shape vectors will all lie on a unit hypersphere). This invalidates the application of linear statistical analysis using tools such as PCA. A standard technique to overcome this problem is to apply a stereographic projection to the shape vectors in order to transform them to points on the tangent space to the Procrustes average. This is simply a case of rescaling the aligned shape vectors as follows:

$$\mathbf{x}''_i = \frac{1}{\bar{\mathbf{x}} \cdot \mathbf{x}_i} \mathbf{x}_i. \tag{3.5}$$

It is to these rescaled vectors that we apply further analysis. In practice, this rescaling slightly improves the efficiency of the model (typically reducing the dimensions required to capture 95% variance by one).

In our experimental results, we demonstrate that our shape alignment procedure results in a superior model to that of Blanz and Vetter [15].

### 3.3.4 Statistical Modelling

We apply PCA to the set of pose free shape vectors $\mathbf{x}''_i$. This performs a basis transformation to an orthonormal coordinate system spanned by the $m$ eigenvectors $\mathbf{P}_i$. Any face surface $\mathbf{x}$, may

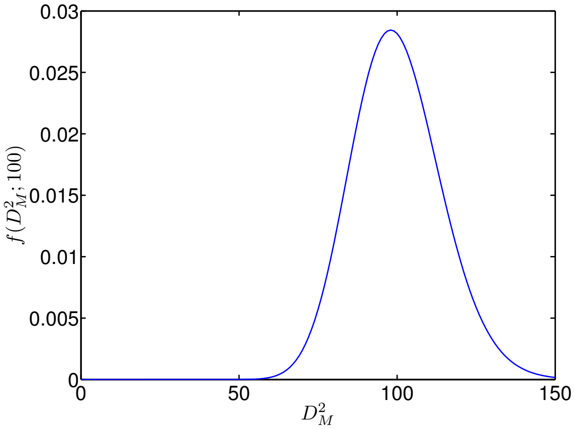

Figure 3.2: Probability density function for parameter vector lengths (measured in terms of the squared Mahalanobis distance) for a 100 parameter model.

now be represented as a linear combination of the average surface and the model eigenvectors:

$$\mathbf{x} = \bar{\mathbf{x}} + \sum_{i=1}^{m} b_i \mathbf{P}_i, \tag{3.6}$$

where $\mathbf{b} = (b_1, \ldots, b_m)^T$ is a vector of parameters. We stack the eigenvectors to form a matrix $\mathbf{P}$, such that we may write: $\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b}$. The PCA eigenvalues $\lambda_i$ provide a measure of how much of the variance of the training data is captured by each eigenvector. We may choose to retain $n < m$ model dimensions, such that a certain percentage of the cumulative variance is captured. We discuss the effect of the number of model dimensions in our experimental results.

Our statistical model also provides an estimate of the probability distribution of the shape vectors. We begin by defining the distance of a sample from the mean in terms of the square of the Mahalanobis distance:

$$D_M^2(\mathbf{b}) = \sum_{i=1}^{n} \left( \frac{b_i}{\sqrt{\lambda_i}} \right)^2. \tag{3.7}$$

Since we assume each parameter follows a Gaussian distribution, the parenthesised terms are independent, normally distributed random variables with zero mean and unit variance. This is exactly the definition of the chi-square distribution.

In other words, the lengths of the parameter vectors (as measured by the square of the

Mahalanobis distance from the mean) follow a chi-square distribution with $n$ degrees of freedom, i.e. $D_M^2 \sim \chi_n^2$. Such a distribution has a mean value of $n$ and variance $2n$. The probability density function over the parameter vector length $g$ for an $n$ parameter model is:

$$f(g;n) = \frac{1}{2^{n/2}\Gamma(n/2)} g^{(n/2)-1} e^{-g/2}. \tag{3.8}$$

The interesting observation here is that the expected length of the parameter vector of a $n$-dimensional model is $n$. The likelihood of a sample having a length close to zero (i.e. approximately the mean sample) is extremely small. For example, a model with 100 dimensions would have a mean vector length of 100 and over 99% of parameter vectors would have lengths greater than 70. The probability of a vector length less than 50 is negligibly small. In Figure 3.2 we show the probability density function for parameter vector lengths for a model with 100 parameters. Note that as $n$ increases, the shape of the chi-square distribution tends to a Gaussian.

This prior on the parameter vector lengths is starkly different to Blanz and Vetter's [15] assumption that the parameters are normally distributed with zero mean. According to their assumption the most probable parameter vector is that with length zero and that the probability decreases as the parameter vector length increases. Infact this contradiction is known as *The Face-Space Typicality Paradox* [21].

We use this prior distribution on the parameter vector lengths to motivate imposing a hard constraint on the length during the fitting process. In effect, we force all samples to lie on the shell of a hyperellipsoid in parameter space (i.e. we fix the vector length to its expected value). We provide empirical validation justifying our choice of manifold in Sections 3.5.2, 4.4.4 and 4.4.5.

## 3.4 3D Face shape from sparse feature points

In this section we lay the framework for a nonlinear, iterative fitting algorithm to estimate a high resolution 3D face surface given the positions of $k$ 2D annotations on the subjects face ($k << p$). In contrast to the analysis-by-synthesis approach of Blanz and Vetter [15], we do not use face appearance or a model of texture variation to reconstruct the 3D surface of a face. This makes the shape recovery process approximately two orders of magnitude faster.

### 3.4.1 The need for regularisation

If the effects of pose are discounted (i.e. if it is assumed that the rotation required to align the model with a set of feature points in the image plane is known), then estimating the morphable model shape parameters that give rise to a particular configuration of landmark points can be solved using linear least squares. Such an approach is impractical as it leads to gross overfitting of the data. Clearly, there is a trade off between the quality of fit to the observed data and prior probability as measured by the model. Using our model of prior probability described in Section 3.3.4 as a regularisation constraint results in a nonlinear optimisation problem.

Blanz et al. [12] proposed a linear, single step solution to this problem based on their assumption of a Gaussian distribution over the parameter vectors. However, the 3D rotations on the position of 2D feature points in the image plane also introduce nonlinearities. To sidestep this problem, Blanz et al. [12] use small angle approximations which are only valid for very small changes in pose. Due to this approximation, the estimated pose may be inaccurate. To overcome this problem, they repeat the process using the result of the first pass as an initialisation, in effect turning their one shot method into an iterative one.

We choose instead to separate the influence of pose and shape parameters on the optimisation and solve the problem using nonlinear, iterative optimisation. As stated above, the chi-squared distribution of parameter vector lengths implies that the parameter vectors lie on the surface of a hyperellipsoid in parameter space. We use this observation to motivate imposing a hard constraint on the length of the estimated parameter vectors. If the number of parameters in the model is $n$, we enforce the constraint $D_M^2 < n$ (recall that the mean length of the parameter vector of a $n$- dimensional model is $n$). In practice, because of the tendency to overfit, the result is that $D_M^2 = n$. To impose this constraint, at each iteration of the minimization we scale the estimated parameter vector such that its length in terms of squared Mahalanobis distance from the mean is $n$:

$$\mathbf{b} = \frac{\sqrt{n}}{D_M(\mathbf{b})}\mathbf{b}. \tag{3.9}$$

### 3.4.2 Fitting to sparse data

Given a set of $k$ annotations marked on the input face ($L_{2d} \in \mathbb{R}^{2k}$), we can determine the vertices corresponding to those salient points on the mean face as shown in Figure 3.6. Once we have the $k$ indexed vertices we can extract their corresponding 2D projections using:

$$\hat{L_{2d}} = P_k T_r^{-1} \left( \bar{\mathbf{x}} + \mathbf{Pb}, \gamma \right). \tag{3.10}$$

where, the term $\bar{\mathbf{x}} + \mathbf{Pb}$ provides the estimated shape, $\gamma$ provides the 3D pose with respect to the mean shape and $P_k$ is the projection of the $k$ indexed vertices under an orthographic projection.

Using Equation 3.10 our aim is to minimize the error between $L_{2d}$ and $\hat{L_{2d}}$ subject to the constraint on the parameter vector length. The quality of fit to the data is measured by:

$$E\left(\mathbf{b}, \gamma\right) = \left\| L_{2d} - \hat{L_{2d}} \right\|^2. \tag{3.11}$$

The optimal parameters are therefore given by:

$$\left(\mathbf{b}^*, \gamma^*\right) = \underset{D_M^2(\mathbf{b}) \leq D_{\max}^2, \gamma}{\arg\min} \; E\left(\mathbf{b}, \gamma\right), \tag{3.12}$$

where $D_{\max}^2$ is the maximum allowable parameter vector length. We examine the effect of

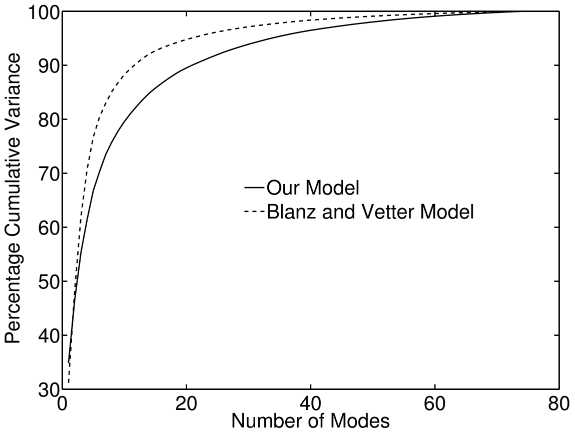

Figure 3.3: Cumulative variance curve.

varying this value in our experimental results. We solve this minimization using Levenberg-Marquardt optimization [63].

## 3.5 Experimental Results

In this section we present the results of our experimental evaluation. We begin by evaluating our approach to constructing a morphable model as a shape space and compare it to that of Blanz and Vetter [15]. We then demonstrate an example application of our model by using it to reconstruct high resolution 3D face surfaces from sparse 2D landmarks.

### 3.5.1 Morphable Model Construction

In this section we compare our strategy for morphable model construction described in Section 3.3 with the state of the art. In order to ensure a fair comparison, we use data provided by [1]. This comprises 100 3D face scans which have been set into correspondence using a modified optical flow algorithm [15]. Blanz and Vetter's [15] model is obtained by applying PCA directly to these shape vectors. Our model uses generalised Procrustes analysis to obtain a stable estimate of the mean face and shape vectors that are free of scale, translations and rotations. We rescale each shape vector according to the tangent space projection given in Equation 3.5.

We divide the data into a training set of 75 scans and a test set of 25 scans. In Figure

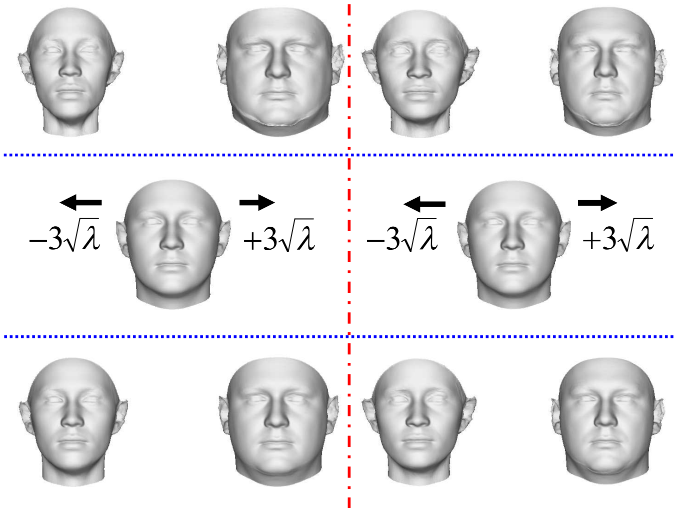

Figure 3.4: Shows the deviations in the two most significant parametric modes. The deviations are shown for our model (left hand side) and the Blanz and Vetter Model (right hand side). The deviations from the mean head (middle row) of the most significant (top row) and second most significant (bottom row) parameters shows the subtle differences in the morphing ability of the two models.

3.3 we plot the percentage cumulative variance captured as a function of the number of model dimensions. It is clear that the model of Blanz and Vetter captures a larger proportion of cumulative variance for a given number of model dimensions. This is often seen as evidence that a model is more efficient and hence superior. Our results show that in fact, the variance captured is spurious and is related to variations in pose rather than identity. Although our model apparently captures less cumulative variance, the variance it does capture is of more use for representing out of sample data. The modes of variation of the two models are visually distinguishable. In Figure 3.4 we show the effect of adding and subtracting the first two modes of variation to the mean face for both models. As can be seen, the characteristics captured by the modes are subtly different.

We compare generalisation ability by measuring the accuracy with which the two models can reconstruct out of sample face meshes. For a novel shape vector, $\mathbf{s}$, we find the optimal parameter vector (in a least squares sense), i.e. $\mathbf{b}^* = \mathbf{P}^T(\mathbf{s} - \bar{\mathbf{x}})$. The reconstructed surface is

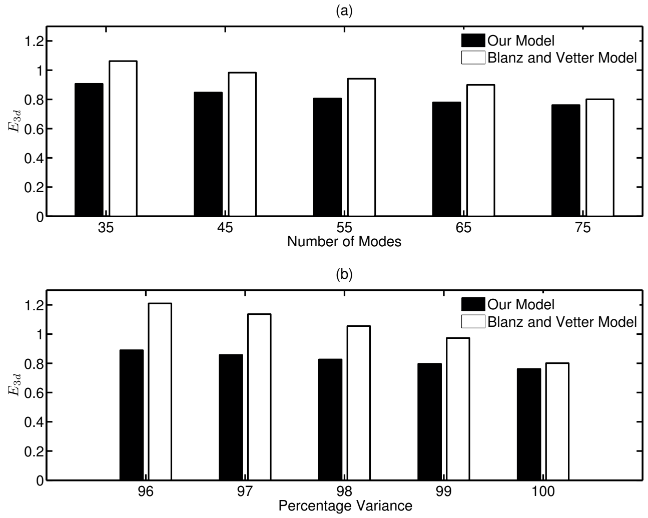

Figure 3.5: (a) $E_{3d}$ (in mm) vs Number of modes, (b) $E_{3d}$ (in mm) vs Percentage variance.

given by $\mathbf{Pb}^* + \bar{\mathbf{x}}$. The average reconstruction error over the whole test set is given by:

$$E_{3d} = \frac{1}{tp} \sum_{i=1}^{t} \sum_{j=1}^{p} \left\| \begin{pmatrix} x_{i,j} \\ y_{i,j} \\ z_{i,j} \end{pmatrix} - \begin{pmatrix} x^r_{i,j} \\ y^r_{i,j} \\ z^r_{i,j} \end{pmatrix} \right\|, \tag{3.13}$$

where $t$ is the number of samples in the test set, $x_{i,j}$ is the $x$ component of the $j$th vertex in the $i$th test sample, with $x^r_{i,j}$ being the corresponding reconstructed value after projection onto the model. Note in Equation 3.13, $\| \cdot \|$ denotes average Manhattan distance.

Figure 3.5 shows the absolute reconstruction errors (in mm) for both models. We vary the number of model dimensions retained and observe its effect on the generalisation error. In Figure 3.5a, we use the same number of dimensions for both models. In Figure 3.5b, we retain as many dimensions as are required to capture a fixed proportion of the cumulative variance. By either measure, our model generalises to unseen data more accurately, even when less cumulative variance has been retained. This makes our model both more efficient (fewer model dimensions required to obtain a given generalisation error) and more accurate (even with all dimensions retained, our model provides higher accuracy).

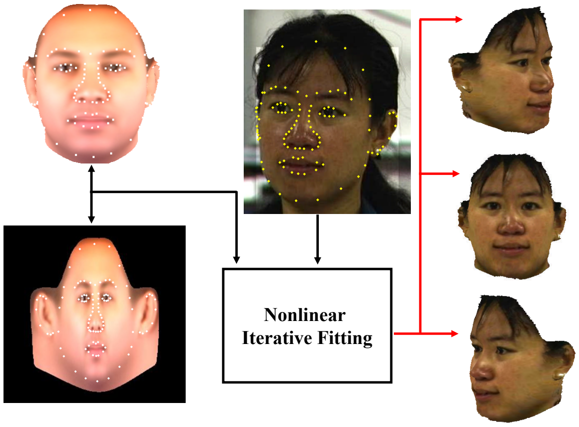

Figure 3.6: Framework for the proposed system. The mean mesh (top left) and 2D input image with $k$ indexed annotations (center) are the inputs to the system. The mean texture map (bottom left) is used to determine the vertices (white dots) corresponding to the $k$ indexed points (yellow dots) in the input image. The system outputs the estimated shape and pose (right column). A simple blending function is used to estimate the occluded texture. The 2D input image is taken from [93].

### 3.5.2 Estimation of 3D faces from Sparse 2D Features

In this section we show the result of using the technique described in Section 3.4 to estimate high resolution 3D face surfaces from sparse 2D feature points. For these experiments we use a morphable model constructed from 100 face scans using the techniques described in Section 3.3. The scans are preprocessed to remove the hair and neck regions and are set into correspondence using a thin-plate spines based warping (Figure 3.1). Each face is represented by $p = 50468$ vertices. To test the effects of the resolution of the model, we also obtained a lower resolution model composed of face meshes containing $p = 3147$ vertices. In both cases, we retain the 99 most significant modes.

Our reconstruction algorithm requires the user to annotate the positions of the landmarks on the input image (Figure 3.6). Note that in practice this could be done using a 2D feature detector. We use $k = 104$ annotated points. The fitting time (on a 1.78 GHz AMD Athlon processor) is approximately 550ms and 10000ms for the $p = 3147$ vertices and $p = 50468$ vertices face models respectively.

As discussed in Section 3.4, we impose a hard constraint, $D^2_{\text{max}}$, on the length of the estimated parameter vectors during the fitting process. Without this constraint, the tendency of

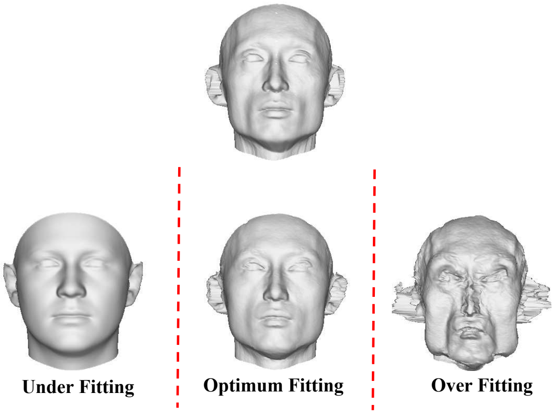

Figure 3.7: Shows the tradeoff between the fitting quality and shape plausibility. Given an unseen face (top row), the bottom row shows the three cases of fitting: (a) Under Fitting (b) Optimum Fitting and (c) Over Fitting.

the algorithm is to overfit the sparse data resulting in a very poor global shape estimate. We examine the effect of varying the value of this constraint on the 3D error of the reconstructed surface. In effect, this parameter controls the trade off between fitting quality and shape plausibility. Our statistical prior predicts that the average length of the parameter vectors for an $n$ parameter model is $n$. For values of $D^2_{\max}$ significantly greater than $n$, the system clearly overfits and the faces are heavily distorted. For values of $D^2_{\max}$ close to zero, the shape estimate is always similar to the average face and the fitting quality is low. We show that the optimum operating point of our algorithm coincides with the prediction of our statistical prior on real data, i.e. optimal performance occurs when $D^2_{\max} \approx n$. An example of underfitting, overfitting and optimal performance is shown in Figure 3.7.

To provide quantitative confirmation of this assertion, we applied our shape estimation algorithm to 50 ground truth samples. These were disjoint from the samples used to construct the morphable model. For each scan we obtained the $k$ indexed vertices corresponding to the salient annotations. These were projected to 2D. The faces were in approximately frontal pose (variations of up to 12° from frontal occurred in practice). For different values of $D^2_{\max}$, we fitted our morphable model to this sparse data and computed the per vertex average 3D reconstruction error over all 50 samples. This experiment was carried out for both the high and

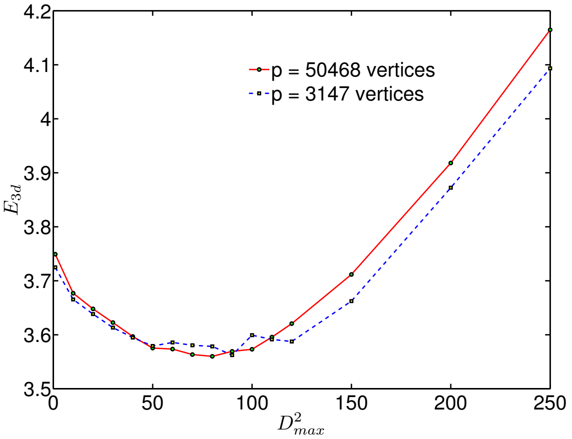

Figure 3.8: $E_{3d}$ (im mm) vs $D^2_{\text{max}}$.

low resolution models.

Figure 3.8 shows the result of this experiment. For both models, the minimum reconstruction error occurs approximately when $D^2_{\text{max}} = n$. The error of the reconstructed faces is approximately $3.6mm$. Although this is only a slight quantitative improvement over using the average face, the perceptual improvement is much greater, as evidenced by Figure 3.7.

Our findings show that our analytical prediction of the average parameter vector lengths coincide with the optimum operating point of our reconstruction algorithm and hence provide a non-heuristic constraint for optimal fitting.

## 3.6 Conclusions

In this chapter we revisited the process of constructing a morphable model from training data. We showed how the tools of thin-plate spline warping and Procrustes analysis can be used to construct a morphable model as a shape space. We also formulated the probabilistic prior over the distribution of parameter vector lengths. In Chapter 3, we further investigate the statistical prior and show how this information can be used to develop a framework that constrains faces to points on a manifold within the parameter space of a linear statistical model.

# Chapter 4

# Manifold Based Constraints

## 4.1 Introduction

Principal Components Analysis (PCA) based linear statistical models have proven to be a very efficient means of modelling human faces [24, 15]. Applying these models to face analysis tasks requires a means to fit the model to observed data. Often this fitting process is underconstrained, prone to converge on local minima and computationally expensive. For these reasons, there is strong motivation for developing additional constraints to reduce the search space of the fitting process.

Psychological results [75, 96] have shown that the model parameter space has an interesting perceptually-motivated interpretation: *identity* relates to direction in parameter space while *distinctiveness* is related to vector length (or equivalently distance from the mean). The reason for this is that increasing the length of a parameter vector simply exaggerates its differences from the average linearly, in other words its *features*, whereas rotating a parameter vector changes the *mix* of features present in the face. This is the justification for using angular difference in face space as a measure of dissimilarity for face recognition [15].

This decomposition also allows a useful probabilistic interpretation. Under the Gaussian assumption, each model parameter is independent and distributed according to a Gaussian distribution. This means that all faces lie on or near the surface of a hyperellipsoid in parameter space, with the probability density over the parameter vector lengths following a chi-square distribution. In other words, distinctiveness is subject to a statistical prior with the distinctiveness of most samples clustered around the expected length.

In this chapter, we use these observations to motivate a representation for faces which decomposes face appearance into identity and distinctiveness subspaces. We focus on statistical models of 3D face shape, though all of our results are equally applicable for any parametric data representation. We use ideas from differential geometry to develop tools which operate in the identity subspace, i.e. which retain constant distinctiveness. We provide empirical justification for constraining samples to have fixed distinctiveness, determined by the expected vector length.

We propose a new algorithm for fitting a statistical face model to data. Many such methods have been proposed previously, the details being dependent on the precise nature of the model and data. This inevitably involves a nonlinear optimisation over the model parameters. Our approach is more general and can be applied to any objective function. It operates via gradient descent on the manifold of equal distinctiveness. In other words, we solve for identity and assume distinctiveness takes its expected value. We show how the method naturally lends itself to a coarse-to-fine optimisation strategy and how the result avoids local minima or overfitting without having to select a regularisation weight parameter. We show that this offers improved performance over two generic nonlinear optimisation algorithms.

## 4.2 Related Work

Perhaps the best known statistical face model is the Active Appearance Model (AAM) [24] which combines a linear model of 2D shape and 2D appearance. Rather than model appearance, the 3D Morphable Model of Blanz and Vetter [15] models the shape and texture which give rise to appearance via a model of image formation. Xiao et al. [101] have used a 3D model in conjunction with a 2D appearance model to enforce geometric constraints on the 2D shape generated.

Construction or training of a statistical face model involves a number of steps: 1. data collection, 2. cross-parameterisation and 3. statistical analysis. When represented in a vector space, face-like samples can be synthesised by taking convex combinations of training faces. However, it is the statistical analysis which allows us to study how the face samples distribute themselves in high dimensional space and which regions of this space correspond to plausible faces, i.e. face space.

Although statistical face models have useful applications when used in a purely generative manner (e.g. for the synthesis of faces), the most compelling applications necessitate face analysis through fitting the model to observed data. This data may take many forms, such as the appearance of a face in one [15, 24, 101] or more [5, 49] images, a noisy and incomplete 3D scan [13] or the location of a sparse set of feature points in an image [12, 54].

When the objective function is underconstrained or ill-posed, the classical approach is to use Tikhonov regularisation (for a linear objective) [95] or more generally to augment the objective function with a regularising term using a Lagrange multiplier [30]. Typically, the regularisation term encourages smaller norms or equivalently, solutions closer to the mean face. With a suitable choice of the regularisation weight, this prevents overfitting and ensures that the resulting face is plausible. However, the optimal choice of regularisation weight may be different for different data samples. By choosing a conservative value, fitting results are likely to be too close to the mean face to capture features of the input face.

Nonlinear objective functions are usually solved using local optimisation methods such as Levenberg-Marquardt or Newton's method. Examples include Cootes's [24] original algorithm for fitting AAMs to images which assumes that the relationship between error and optimal additive parameter updates is constant. Matthews and Baker's [67] inverse compositional algorithm avoided this assumption allowing faster and more robust convergence.

A challenging problem occurs if the objective function is nonconvex and contains local minima far from the global optimum. In this case, global optimisation methods may provide improved performance, for example Blanz and Vetter [15] use stochastic Newton optimisation in an analysis-by-synthesis framework. Careful initialisation and regularisation is required to obtain stable performance. Another alternative is to augment the objective function with additional terms which encourage convexity of the search landscape. This was the strategy adopted by Romdhani et al. [87] who fitted a 3D morphable model using a hybrid objective function composed of image features (such as edges, specular highlights and pixel intensities) in conjunction with model priors. It was found that this hybrid objective was smoother and solutions closer to the global optimum could be found using local optimisation. Another approach is to design an objective function which can be expressed as a linear system of equations or other convex optimisation problem. In this case the global optimum can be found efficiently. For example, Romdhani et al. [85] showed how to use linear equations to recover the shape and texture parameters irrespective of pose and lighting conditions of the face image. All these approaches trade off satisfaction of a model based prior against quality of fit. To ensure robust performance, these approaches must favour the prior, resulting in model dominance.

In this chapter, we propose instead to solve the model fitting problem within the subspace of maximally likely faces. This requires the solution of an optimisation problem on the surface of a manifold. This problem has been considered previously in the medical imaging [2], signal processing [66], computer vision [65], robotics [46] and projective geometry [59] communities. Generic methods for optimisation on arbitrary manifolds have also been proposed [50]. We focus on the case of a hyperspherical manifold and derive efficient implementations of the log and exponential maps. We use these to develop a hyphershperical gradient descent algorithm which exploits the closed nature of the manifold to reduce line searches to interval searches.

## 4.3 Contributions

In Section 4.4 we begin by describing our statistical model and manifold. We first introduce tools from differential geometry which are necessary for developing our methodology and then provide empirical validation to justify our choice of manifold. In Section 4.5 we describe how warps and averages between two or more faces can be constrained to the manifold and compare the result with linear methods. In Section 4.6 we present our principle contribution: a method for fitting the model to data within the subspace defined by the manifold. In Section 4.7 we provide results for two contrasting objective functions (one overconstrained, the other underconstrained) and compare with generic nonlinear optimisers using a regularised objective.

## 4.4 Statistical Modelling

Consider a sample of 3-dimensional face meshes which are in dense correspondence (i.e. the same point on every face has the same vertex index). The $i$th shape is represented by a vector of $p$ vertices $\mathbf{s}_i = (x_1, y_1, z_1, \dots, x_p, y_p, z_p) \in \mathbb{R}^{3p}$. Given $m$ such shape vectors, we use principal components analysis to obtain an orthogonal coordinate system spanned by the $m$ eigenvectors, where $\mathbf{p}_i$ is the $i$th eigenvector. Any shape vector $\mathbf{s}$ may now be represented as a linear combination of the average shape and the model eigenvectors:

$$\mathbf{s} = \bar{\mathbf{s}} + \sum_{i=1}^{m} c_i \mathbf{p}_i, \tag{4.1}$$

where $\mathbf{c} = [c_1 \ \dots \ c_m]^T$ is a vector of parameters. We stack the eigenvectors to form a matrix $\mathbf{P}$, such that we may write: $\mathbf{s} = \bar{\mathbf{s}} + \mathbf{Pc}$. The PCA eigenvalues, denoted $\lambda_i$ for the $i$th eigenvalue, provide a measure of how much of the variance in the training data is captured by each eigenvector. We may choose to retain $n < m$ model dimensions, such that a certain percentage of the cumulative variance is captured. Psychological results show us that the dimensionality of face space is relatively small (Meytlis and Sirovich [70] suggest 100 dimensions is sufficient, even using a crude eigenface model). We discuss the effect of the number of model dimensions and empirically evaluate their stability in Section 4.4.4.

Our interest in this paper is to explore how shape samples drawn from a population distribute themselves in parameter space and how we can use this knowledge to constrain operations. We define the vector $\hat{\mathbf{c}} = [c_1/\sqrt{\lambda_1} \ \dots \ c_n/\sqrt{\lambda_n}]^T$ as the variance-normalised parameter vector. This vector is distributed according to a multivariate Gaussian with zero mean and unit variance, i.e. $\hat{\mathbf{c}} \sim \mathcal{N}(0, \mathbf{I}_n)$. This is the prior constraint typically used in the model fitting process to ensure that solutions remain plausible. It is maximised by a zero vector, which corresponds to the mean sample.

However, another interpretation based on the parameter vector length is possible. The squared norm of $\hat{\mathbf{c}}$ corresponds to the square of the Mahalanobis distance of $\mathbf{c}$ from the mean:

$$\|\hat{\mathbf{c}}\|^2 = D_M^2(\mathbf{c}) = \sum_{i=1}^{n} \left( \frac{c_i}{\sqrt{\lambda_i}} \right)^2 . \tag{4.2}$$

Since we assume each parameter follows a Gaussian distribution, the parenthesised terms are independent, normally distributed random variables with zero mean and unit variance. The sum of the square of such variables follows a chi-square distribution with $n$ degrees of freedom, i.e. $\|\hat{\mathbf{c}}\|^2 \sim \chi_n^2$. This distribution has expected value $n$ and variance $2n$. Hence, the standard deviation grows as the square root of the mean and the vector lengths become relatively more tightly concentrated about the mean length as the number of dimensions grows.

These two apparently contradictory distributions suggest that the mean face is the most probable sample but has a highly improbable vector length (this has been reported in the psychology literature as *The Face-Space Typicality Paradox* [21]). For example, a model with 100 dimensions would have an expected vector length of 100 and over 99% of parameter vectors would have lengths between 70 and 130. The probability of a vector length less than 50 is negligibly small.

A note of caution is required to accompany this analysis. Under the assumption that each parameter vector follows a Gaussian distribution, the chi-square analysis holds. However, since PCA eigenvectors capture the maximum possible variance, the eigenvalues decay rapidly with increasing dimension number. The effect of this is to amplify less significant parameters, i.e. the denominator in Equation 4.2 becomes small. The reason that this causes a problem is that eigenvectors associated with smaller eigenvalues are more susceptible to the influence of noise, particularly when the training set size is small. The empirical analysis in Section 4.4.4 confirms the above mentioned.

### 4.4.1 Identity as Direction

Our argument is that valid members of the class will occupy a subspace of parameter space. These points will lie close to the surface of a hyperellipsoid, the diameters of which are determined by the eigenvalues of the data.

To negate the need for regularisation, we choose to force all samples to lie on the surface of the hyperellipse, i.e. we fix distinctiveness (vector length) to its expected value as a hard constraint. With distinctiveness fixed, points on the manifold correspond to unique identities. Using this representation, face processing and analysis are transformed to operations on the surface of a manifold.

The analysis of data on a hyperellipsoidal manifold is extremely complex. Therefore, without loss of generality, we transform the manifold to a hypersphere by scaling each dimension by its corresponding standard deviation. For the remainder of this paper, we therefore represent parameter vectors with squared Mahalanobis length $n$ as unit vectors in $\mathbb{R}^n$: $\mathbf{x} = \frac{1}{\sqrt{n}} \left[ \frac{c_1}{\sqrt{\lambda_1}} \ \ldots \ \frac{c_n}{\sqrt{\lambda_n}} \right]^T$, where $\|\mathbf{x}\| = 1$.

### 4.4.2 Log and Exponential Maps

Linear operations in Euclidian space such as averaging, warping and computing partial derivatives must be reformulated for data which lies on a curved manifold. This is conveniently done in tangent space, where geodesic curves through the point of tangency correspond to straight lines. Transforming points from the surface of a manifold to tangent space and back is done using operations from differential geometry, namely the log and exponential map.

A unit vector in $n$-dimensional space $\mathbf{x} \in \mathbb{R}^n$, may be considered as a point lying on the hyperspherical manifold $x \in S^{n-1}$. The two are related by $\mathbf{x} = \Phi(x)$ where $\Phi : S^{n-1} \mapsto \mathbb{R}^n$ is an embedding. If $v \in T_b S^{n-1}$ is a vector in the tangent space to $S^{n-1}$ at a base point $b \in S^{n-1}$, the exponential map, denoted $\text{Exp}_b$ of $v$ is the point on $S^{n-1}$ along the geodesic in the direction of $v$ at distance $\|v\|$ from $b$. Figure 4.1 provides a visual illustration of the operation for the $S^2$ manifold. The inverse of the exponential map is the log map, denoted $\text{Log}_b$.

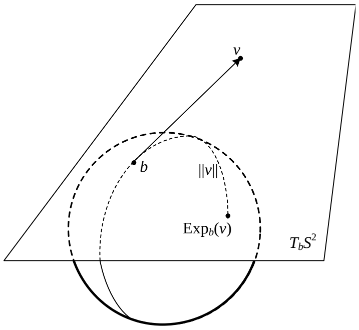

Figure 4.1: Exponential map for the $S^2$ manifold.

The geodesic distance (i.e. angular difference) between two points $x_1, x_2 \in S^{n-1}$ on the unit hypersphere can be expressed in terms of the log map, i.e. $d(x_1, x_2) = \|\mathrm{Log}_{x_1}(x_2)\| = \arccos(\Phi(x_1) \cdot \Phi(x_2))$. In Section 4.4.3, we derive a simple and efficient means to compute the log and exponential maps for the unit hypersphere. In the remaining sections, we use the log and exponential maps to perform useful operations on the manifold.

### 4.4.3 Log and Exponential Maps for the Hypersphere

We propose a novel implementation of the exponential and log maps for a unit hypersphere which is both efficient and conceptually simple. We do so using a stereographic projection. The log map of a point $x$at basepoint $b$ is calculated as follows. We define the tangent vector $v' \in T_bS^{n-1}$ as the stereographic projection of $x$ from $-b$ to the tangent space to $S^{n-1}$ at $b$. This tangent vector has the correct direction but incorrect magnitude. To obtain the log map of $x$, we rescale $v'$ giving $v$, such that $\|v\| = d(b, x)$. The exponential map is computed by reversing this process, i.e. by applying an inverse stereographic projection to the rescaled tangent vector. Figure 4.2 clarifies the geometry involved for the $S^1$ case.

In practice, we represent points on both the hyperspherical manifold and the tangent space as vectors embedded in $\mathbb{R}^n$. Our proposed implementation of the log map of $x$ at base point $b$ is therefore computed with respect to unit vectors in $\mathbb{R}^n$: $\mathbf{b} = \Phi(b)$ and $\mathbf{x} = \Phi(x)$.

We begin by computing the angle $\alpha$ from the known edge lengths of the triangle formed by $\mathbf{b}$, $\mathbf{x}$ and $-\mathbf{b}$: $d(-\mathbf{b}, \mathbf{x}) = \|\mathbf{b} + \mathbf{x}\|$, $d(\mathbf{b}, \mathbf{x}) = \|\mathbf{x} - \mathbf{b}\|$ and since $\mathbf{b}$ is a unit vector: $d(-\mathbf{b}, \mathbf{b}) = 2$.

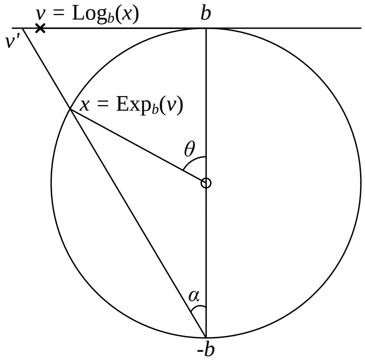

Figure 4.2: Computing log and exponential maps using a stereographic projection for the $S^1$ manifold.

From the law of cosines, we can therefore write:

$$\alpha = \arccos\left(\frac{4 + \|\mathbf{b}+\mathbf{x}\|^2 - \|\mathbf{x}-\mathbf{b}\|^2}{4\|\mathbf{b}+\mathbf{x}\|}\right). \tag{4.3}$$

Considering now the right angled triangle formed by $\mathbf{v}'$, $\mathbf{b}$ and $-\mathbf{b}$, we can compute $d(-\mathbf{b}, \mathbf{v}') = 2\cos^{-1}\alpha$. The stereographic projection of $\mathbf{x}$ to the tangent plane at $\mathbf{b}$ is therefore given by:

$$\mathbf{v}' = \frac{2(\mathbf{b}+\mathbf{x})}{\|\mathbf{b}+\mathbf{x}\|\cos\alpha} - \mathbf{b}. \tag{4.4}$$

The log map is given by rescaling a vector from $\mathbf{b}$ to $\mathbf{v}'$ such that its length is equal to the angular distance between $\mathbf{b}$ and $\mathbf{x}$, i.e. $\theta = \arccos(\mathbf{b}\cdot\mathbf{x})$:

$$\Phi_T\left(\mathrm{Log}_b(x)\right) = \mathbf{b} + \frac{\theta(\mathbf{v}'-\mathbf{b})}{\|\mathbf{v}'-\mathbf{b}\|}. \tag{4.5}$$

The result is a vector in the tangent space $T_bS^{n-1}$ embedded in $\mathbb{R}^n$ according to an arbitrary embedding $\Phi_T : T_bS^{n-1} \mapsto \mathbb{R}^n$.

A similar expression can be derived for the exponential map. Given a tangent vector embedded in $\mathbb{R}^n$, $\mathbf{v} = \Phi_T(v)$, our aim is to find the point which corresponds to a stereographic projection of $x = \mathrm{Exp}_b(v)$ to $T_bS^{n-1}$. This can then be rescaled to the unit sphere. With reference to the same geometry as above, the angular distance is given by:

$$\theta = \|v\| = \|\mathbf{b} - \mathbf{v}\|. \tag{4.6}$$

With this angle to hand, the following distances can be computed:

$$d(-\mathbf{b}, \mathbf{x}) = \sqrt{2(1 + \cos\theta)}, \tag{4.7}$$

$$d(\mathbf{b}, \mathbf{x}) = 2\sin(\theta/2), \tag{4.8}$$

which can be used to find the stereographic projection angle:

$$\alpha = \arccos\left(\frac{4 + d(-\mathbf{b}, \mathbf{x})^2 - d(\mathbf{b}, \mathbf{x})^2}{4d(-\mathbf{b}, \mathbf{x})}\right). \tag{4.9}$$

It is then straightforward to find the stereographically projected point:

$$\mathbf{v}' = \mathbf{b} + \frac{(\mathbf{v} - \mathbf{b})2\tan\alpha}{\|\mathbf{v} - \mathbf{b}\|}, \tag{4.10}$$

and finally to invert the stereographic projection to give the exponential map:

$$\Phi(\mathrm{Exp}_b(v)) = d(-\mathbf{b}, \mathbf{x})\frac{\mathbf{v}' + \mathbf{b}}{\|\mathbf{v}' + \mathbf{b}\|} - \mathbf{b}. \tag{4.11}$$

These expressions hold for unit vectors in any number of dimensions.

### 4.4.4 Empirical Evaluation: $\chi^2$ Prediction

Before we consider applications of processing data on the manifold described above, we provide some empirical assessment of how well real world data adheres to the theoretical prediction made in Section 4.4.1. In order for all plausible data samples to lie on or near the predicted manifold, the assumption of parameter vector lengths following the chi-squared distribution must hold. In turn, the distribution of faces along each eigenvector must follow a Gaussian distribution. In

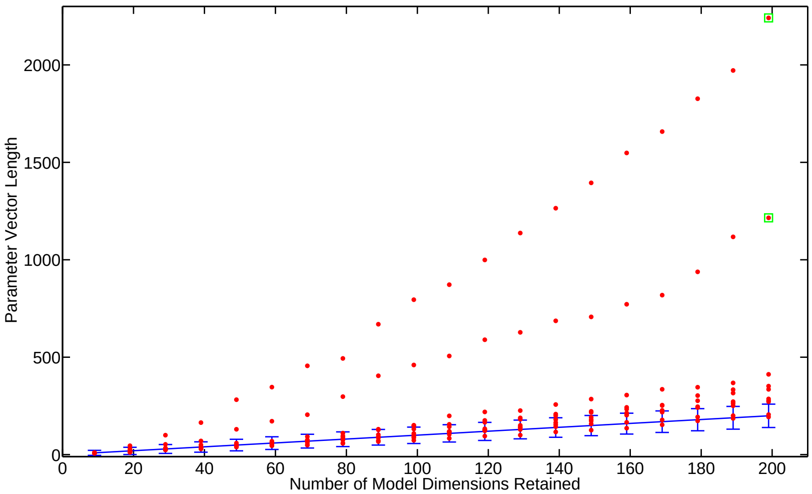

Figure 4.3: Parameter vector length vs Number of model dimensions retained. The red points represent the vector length for each out-of-sample face. The error bars (blue) represent three standard deviation variation around the mean of the chi-square distribution. The green squares denote the two faces which grossly overfit when projected onto the model (see column (b) in Figure 4.6).

practice, these eigenvectors are estimated from a sparse sample of a high dimensional space. In the case of a dense 3D face shape model, observations typically consist of tens of thousands of vertices while the training set typically comprises only hundreds of samples.

Clearly, the validity of the estimated manifold depends on the quality of the estimated eigenvectors and therefore the size and diversity of the training set. Within-sample data (i.e. that used to train the model) adheres almost exactly to the manifold assumption. Hence, we empirically evaluate whether out-of-sample data follows the theoretical prediction. For our empirical test we use the Basel Face Model (BFM) [76]. The BFM is a 3D morphable model constructed from 200 faces. An additional 10 unseen (out-of-sample) faces are provided which are in correspondence with the model.

Given an out-of-sample face, $\mathbf{s}$, the optimal parameter vector (in a least squares sense) is given by simply projecting the face onto the model, i.e. $\mathbf{c}^* = \mathbf{P}^T(\mathbf{s} - \bar{\mathbf{s}})$. Substituting $\mathbf{c}^*$ back into Equation 4.1, we obtain $\mathbf{s}^{\text{mod}}$, the shape which minimises $\|\mathbf{s}^{\text{mod}} - \mathbf{s}\|^2$. We do this for each face and measure the distance of the resulting point in parameter space from the mean (in terms of squared Mahalanobis distance). We vary the number of model dimensions and show the results in Figure 4.3. The blue line shows the expected vector length which grows linearly with the number of dimensions. We indicate the expected spread of vector lengths by using error bars to show three standard deviations either side of the mean. Red points represent the vector length for a sample projected onto the model.

There are a number of interesting observations to make on this plot. For a small number of model dimensions, the samples adhere to the statistical prediction. However, as the number of dimensions increases, certain samples deviate rapidly from the prediction. These are faces which are dissimilar to the those in the training set and whose shape is poorly approximated by the model eigenvectors. The two extreme cases indicated by the green squares are gross overfits, visualisations of which can be seen in Figure 4.6. The explanation for this is that with sparse training data, the less significant eigenvectors are unstable and cannot be reliably estimated. A much larger training set may mitigate this problem and lead to a model for which out-of-sample faces adhere more closely to the statistical prediction. Nevertheless, to retain the expressiveness of the model the dimensions with smaller eigenvalues are important and cannot be discarded. What is required are constraints which prevent overfitting and ensure that model instances remain plausible. Our proposal to do this by enforcing a hard constraint on the parameter vector lengths is evaluated in the next section.

### 4.4.5 Empirical Evaluation: Manifold Approximation

Irrespective of how well out-of-sample data adheres to the manifold assumption, from a practical perspective the more important question is whether forcing samples to lie on the manifold provides a useful constraint. We attempt to answer this by measuring the effect of enforcing the manifold constraint on the "plausibility" of a face. For a face to be plausible is must appear face-like but to be a plausible representation of a specific face it must also have a low perceptual error between the original face and its model representation. There are many proposed measures for computing the perceptual error between a mesh and its reconstruction. Most are based on

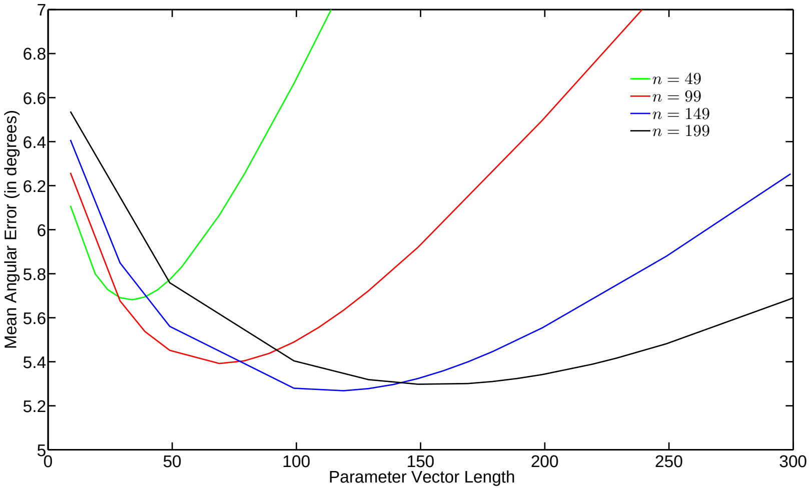

Figure 4.4: Mean angular error vs Parameter vector length (for four different values of model dimensions retained). All reported error measures are averaged over the 10 unseen faces in the BFM.

the surface derivatives since it is surface orientation which determines appearance. Hence, we measure perceptual error in terms of the angular difference between surface normals.

We compare the optimal model-based reconstruction described above, $\mathbf{s}^{\text{mod}}$, to that obtained by projecting $\mathbf{c}^*$ to the closest point on the hyperspherical manifold:

$$\hat{\mathbf{c}}^{\text{man}} = \frac{\sqrt{n}}{\|\hat{\mathbf{c}}^*\|}\hat{\mathbf{c}}^*. \tag{4.12}$$

We refer to the resulting shape as $\mathbf{s}^{\text{man}}$. It should be noted we use the variance-normalised parameter vectors in Equation 4.12. We begin by establishing whether the expected vector length predicted by the chi-square distribution is a good choice with which to define the manifold. To do so, in Figure 4.4 for the 10 out-of-sample faces in the BFM, we show the effect when the vector given by projection onto the model is rescaled to various lengths. The $x$-axis shows the enforced vector length, the $y$-axis shows the mean angular error in the surface normals (i.e. perceptual error). We perform this test for $n = 49$, 99, 149 and 199 parameter model.

The plot for each model shows a similar trend, with small and large vector lengths having a higher error (underfitting and overfitting respectively) and a minimum occurring close to the

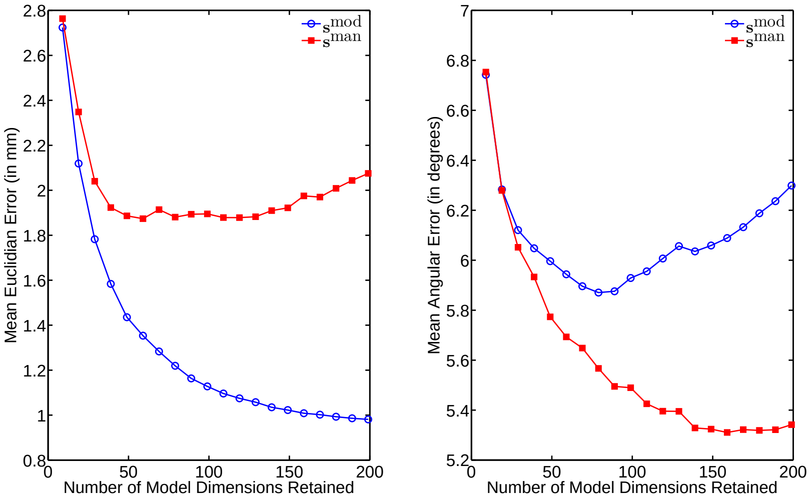

Figure 4.5: Left plot (a): Mean Euclidian error vs Number of model dimensions retained; Right plot (b): Mean angular error vs Number of model dimensions retained. All reported error measures are averaged over the 10 unseen faces in the BFM.

chi-square prediction (i.e. when the parameter vector lengths are forced to $n$). This suggests our statistically motivated choice of hard constraint is reasonable.

Finally, we wish to show that forcing samples to lie on the manifold reduces perceptual error. In Figure 4.5 (a) we plot the mean Euclidian error for $\mathbf{s}^{\text{mod}}$ and $\mathbf{s}^{\text{man}}$. Since it is optimal, $\mathbf{s}^{\text{mod}}$ achieves a lower Euclidian error than $\mathbf{s}^{\text{man}}$ for all $n$ and this error decreases monotonically as the number of dimensions increases. However, the purpose of our choice of manifold is to enforce plausibility. If we repeat the same experiment but instead plot angular (perceptual) error, shown in Figure 4.5 (b), we see that $\mathbf{s}^{\text{man}}$ achieves a lower angular error than $\mathbf{s}^{\text{mod}}$ for all $n$. Increasing the number of model dimensions yields an almost monotonic reduction in perceptual error for $\mathbf{s}^{\text{man}}$, whilst the perceptual error of the optimal least-squares surfaces ($\mathbf{s}^{\text{mod}}$) begins to increase beyond about 80 dimensions. Two visual examples are shown in Figure 4.6. The two out-of-sample faces in column (a) are grossly overfitted when allowed to minimise least squares error (column (b)). When rescaled to the manifold (column (c)) the perceptual error reduces and the faces are visually plausible.

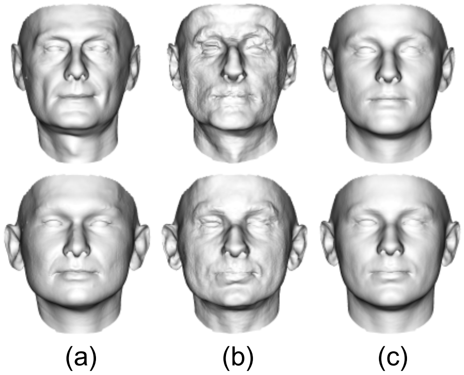

Figure 4.6: Manifold approximation: (a) input unseen face; (b) least squares fit to vertices; (c) parameter vector of (b) rescaled to manifold. All the results are for a $n = 199$ parameter model.

## 4.5 Plausibility-preserving warps and averages

### 4.5.1 Warps

Warping between faces or, more generally, computing weighted combinations of two or more faces has applications in animation and in the production of stimuli for psychological experiments [75]. The most obvious way to warp between two shapes that are in dense correspondence is to linearly warp each vertex from its position in one shape to its position in the other. Equivalently, this can be approximated by linearly warping between the two vectors of PCA parameters. However, in either case the intermediate faces will not correspond to plausible faces. Since the manifold of maximally probable distinctiveness is curved, any linear warp will include faces that do not lie on the manifold, with the least plausible face occurring halfway along the warp.

Face-antiface warps provide a particularly interesting special case. An antiface is the antipodal point of a source face on the manifold. Perceptually, antifaces appear *opposite* in some sense to the original face. The vector connecting a face to its antiface in parameter space passes through the mean. A linear warp between a face and antiface is therefore well-defined but will include implausible faces for the duration of the warp. There is a further problem with such linear warps. Psychological studies have shown that there is a perceptual discontinuity as the face trajectory crosses the mean [75]. In other words, as identity flips from face to antiface, the

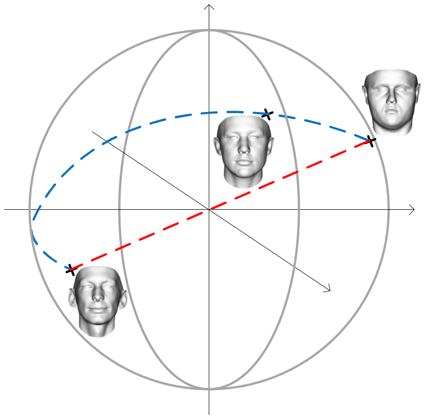

Figure 4.7: Warping between face and antiface on the $S^2$ manifold. A linear warp is shown in red and one of the possible plausibility-preserving warps is shown in blue.

perceptual effect of a small movement through face space is exaggerated.

Instead, we propose warps which take place across the surface of the manifold, following the geodesic curve between the two source faces. Another way to view these warps is as a rotation of a unit vector in $\mathbb{R}^n$. All intermediate faces in this case have equal distinctiveness and are equally plausible. In the case of antifaces, there is no single geodesic warp connecting face to antiface. In fact, there are an infinite number of valid warps, all of length $\pi$. Any such warp will smoothly vary identity from the source face to its antiface, via a series of faces with uniform distinctiveness. One way to conceptualise this is that we can set off from a point on the hyperspherical manifold in any direction and reach the antiface after travelling a distance $\pi$.

An interesting result of this observation is that we can choose any intermediate face as a target which will be visited on the warp from face to antiface. This gives us a way to specify one of the infinite face-antiface warps and may also have interesting applications in generating stimuli for psychological studies. This idea is demonstrated in Figure 4.7 for the $S^2$ manifold, which shows the difference between a plausibility-preserving and linear warp.

For a source face $x_{src}$ and intermediate target face $x_{tar}$, we can define a unit vector in the tangent space, $v \in T_{x_{src}}S^{n-1}$, from $x_{src}$ in the direction of $x_{tar}$: $v = \frac{\mathrm{Log}_{x_{src}}(x_{tar})}{d(x_{src},x_{tar})}$. A geodesic warp from $x_{src}$ to $x_{tar}$ is therefore given by following this vector by a distance specified by the

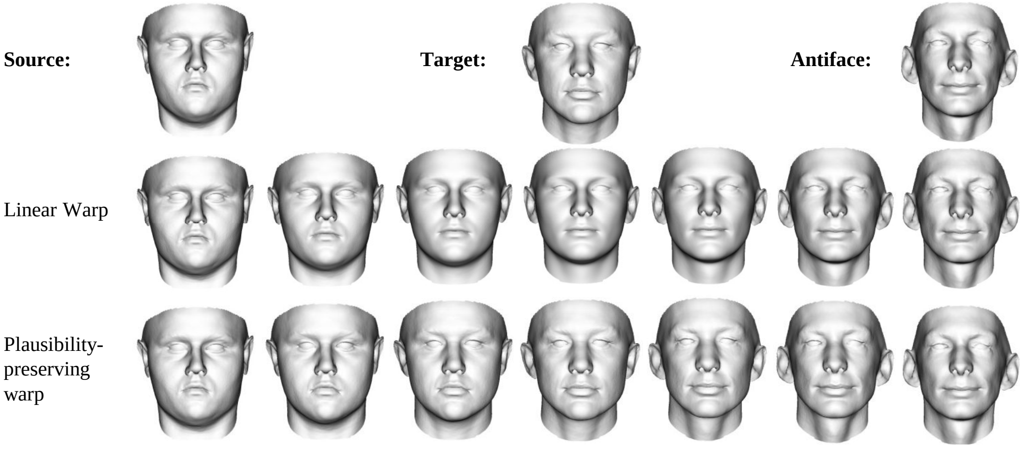

Figure 4.8: Linear versus plausibility-preserving warp from face to antiface.

warping parameter $w$:

$$x_{war} = \text{Exp}_{x_{src}} \left( w \frac{\text{Log}_{x_{src}}(x_{tar})}{d(x_{src}, x_{tar})} \right). \tag{4.13}$$

When $w = 0$ we obtain the source face, i.e. $x_{war} = x_{src}$, and when $w = d(x_{src}, x_{tar})$ we obtain the target face, i.e. $x_{war} = x_{tar}$. If we set $w = \pi$ we obtain the antiface to $x_{src}$. Intermediate faces are obtained when $w \in (0, \pi)$.

We show an example warp from face to antiface via an intermediate target face in Figure 4.8 using the 199 parameter BFM [76]. Note that the effect is of smooth variation of identity, with each of the intermediate faces containing significant detail. We contrast this with a linear warp through the mean face which results in implausibly smooth intermediate faces and no transition through intermediate identities. In Figure 4.9 we plot the parameter vector lengths for the linear and plausibility-preserving warps.

### 4.5.2 Averages

Given $u > 2$ source faces, $x_1, \dots, x_u \in S^{n-1}$, we wish to compute a plausible average face which captures characteristics of each of the source faces. The linear or Euclidian mean of the parameter vectors minimises the sum of square error in $\mathbb{R}^n$ from the average to each of the source faces. This is the *extrinsic mean* and will not lie on the manifold. The result is that the face is implausibly smooth and lacking in features. We propose the use of the *intrinsic* or

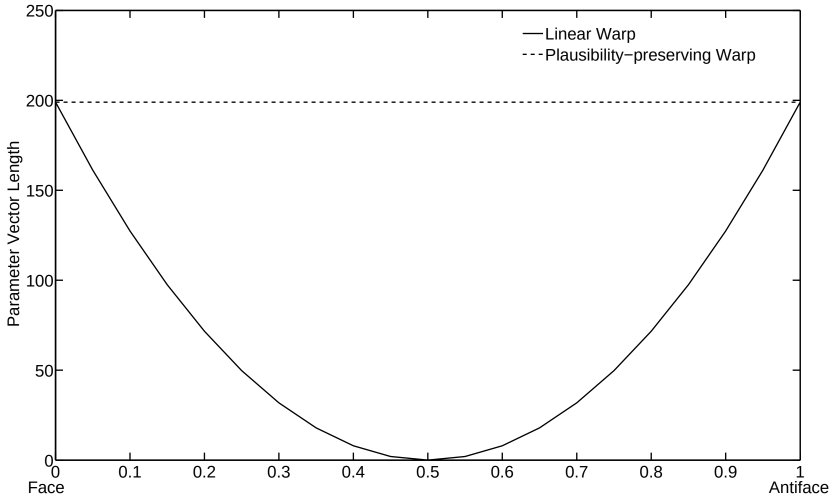

Figure 4.9: Vector length or ‘plausibility’ is plotted throughout a warp between a face and antiface (see Figure 4.8).

Karcher mean [77]. For $u = 2$, this can be found using the warping equation given above with $w = 0.5$. For $u > 2$, this is the point $x_\mu \in S^{n-1}$ which minimises the total squared geodesic distance to each of the source faces:

$$x_\mu = \underset{x \in S^{n-1}}{\arg\min} \sum_{i=1}^{u} d(x, x_i)^2. \tag{4.14}$$

This point cannot be found analytically, so we solve it as an iterative optimisation using the gradient descent method of Pennec [77]. We initialise our estimate as one of the source data points, i.e. $x_\mu^{(0)} = x_1$. The estimated intrinsic mean is then iteratively updated as follows:

$$x_\mu^{(j+1)} = \mathrm{Exp}_{x_\mu^{(j)}} \left( \frac{1}{u} \sum_{i=1}^{u} \mathrm{Log}_{x_\mu^{(j)}}(x_i) \right). \tag{4.15}$$

This process converges rapidly, typically within 5 iterations. In Figure 4.10 we compare our plausibility-preserving averages with linear averaging of the 74 dimensional parameter vectors obtained using the data provided by [1]. Notice that each of the Euclidian averages appears unrealistically smooth, whereas the averages computed on the manifold clearly show the presence of distinct features present in the source faces (for example, the broader nostrils of face 1 are

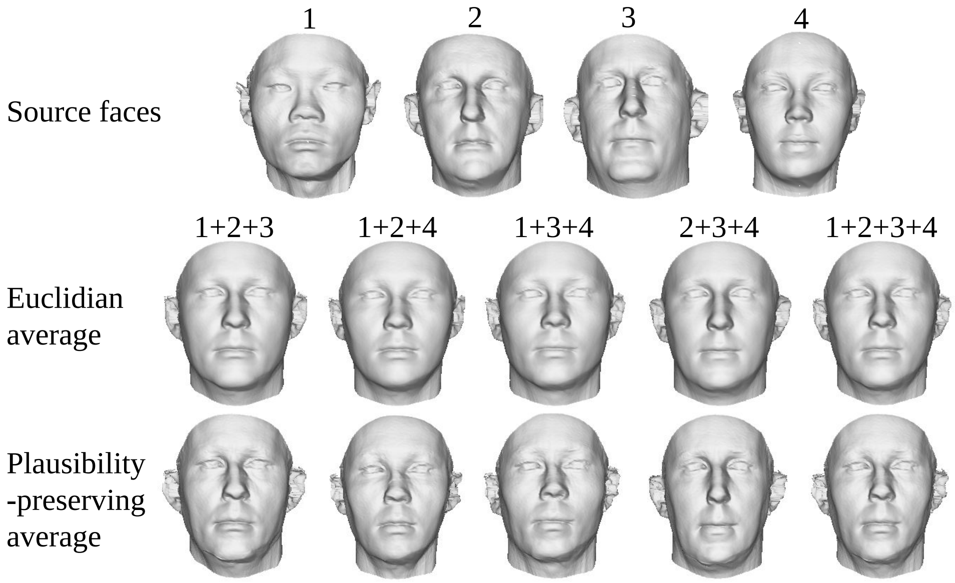

Figure 4.10: Linear versus plausibility-preserving averages.

visible in the first three averages but not the fourth).

## 4.6 Model fitting on the manifold of plausible faces

The most powerful application of the identity manifold is to use it for the purpose of constraining the process of fitting a model to data. Suppose the function $\varepsilon : S^{n-1} \mapsto \mathbb{R}$ is an objective function which evaluates the quality of fit of a face represented by a point on the plausibility manifold to some observed data. This function could take any form, for example the difference between predicted and observed appearance in an analysis-by-synthesis framework or the error between a sparse set of feature points. We pose model fitting as finding the point on the manifold which minimises this error, i.e.:

$$x^* = \arg \min_{x \in S^{n-1}} \varepsilon(x). \tag{4.16}$$

In doing so, we ensure that plausibility is enforced as a hard constraint. Note also that the optimisation is more heavily constrained since the dimensionality of the hypersphere is 1 less than the parameter space.

### 4.6.1 Local Optimisation

We can perform gradient descent on the surface of the manifold to find a local minimum in the error function. The fact that our manifold is hyperspherical has some interesting implications for such an approach. We must first compute the gradient of the objective function in terms of a vector on the tangent plane: $\nabla\varepsilon(x) \in T_x S^{n-1}$. To do so, we compute the gradient in terms of a vector in $\mathbb{R}^n$ and project the result to the tangent plane as follows:

$$\nabla\varepsilon(x) = \mathrm{Log}_x \left( \Phi^{-1} \left( \frac{\mathbf{x} - \mathbf{g}}{\|\mathbf{x} - \mathbf{g}\|} \right) \right) \tag{4.17}$$

where $\mathbf{x} = [x_1 \ \ldots \ x_n]^T = \Phi(x)$. The gradient in $\mathbb{R}^n$:

$$\mathbf{g} = [\partial_{x_1}\varepsilon(x) \ \ldots \ \partial_{x_n}\varepsilon(x)]^T , \tag{4.18}$$

is approximated by using finite differences to calculate the partial derivatives:

$$\partial_{x_i}\varepsilon(x) \approx \frac{\varepsilon(x'_i) - \varepsilon(x)}{\epsilon}, \tag{4.19}$$

where $x'_i = \Phi^{-1}([x_1 \ \ldots \ x_i + \epsilon \ \ldots \ x_n])$.

With a means to compute the gradient, we can iteratively minimise the objective function by adapting the gradient descent algorithm to operate on the surface of a manifold:

$$x^{(t+1)} = \mathrm{Exp}_{x^{(t)}} \left( -\gamma \nabla\varepsilon(x^{(t)}) \right) , \tag{4.20}$$

where $\gamma$ is the step size. Note that as $\gamma$ varies, the point $\mathrm{Exp}_x \left( -\gamma\nabla\varepsilon(x) \right) \in S^{n-1}$ traces out a great circle about the hypersphere. This is the search space for the one-dimensional line search at each iteration of gradient descent.

### 4.6.2 Coarse-to-fine Model Fitting

The difficulty with our approach is choosing an unbiased initialisation. Existing methods for fitting statistical models to data typically commence from an initialisation of the mean (i.e. zero parameter vector), e.g. [15, 24]. However, this point lies far from the plausibility manifold and is therefore unsuitable in our case.

We tackle this problem and also reduce susceptibility to becoming trapped in local minima by proposing a coarse-to-fine algorithm which iteratively increases the number of model dimensions considered in the optimisation.

Consider in the simplest case a two dimensional model (in the one dimensional case the manifold collapses to a pair of points and is the boundary of a line segment). In two dimensions the manifold is $S^1$ (i.e. a unit circle) and requires the optimisation of a single angular parameter, $\theta$. The result in two dimensions, $\mathbf{x}^{(2)} = [\cos\theta^* \;\sin\theta^*]^T$, is given by solving the following interval search problem:

$$\theta^* = \arg\min_{\theta} \varepsilon(\Phi^{-1}([\cos\theta \;\sin\theta]^T)),\; 0 \le \theta < 2\pi, \tag{4.21}$$

which we solve using golden section search [52]. We use this result to initialise the solution in three dimensions, initially setting the third parameter to zero: $\mathbf{x}^{(n)}_{\text{init}} = \left[\mathbf{x}^{(n-1)} \mid 0\right]$. We then perform gradient descent. We continue this process, incrementally adding dimensions to the optimisation, each time setting the new parameter to zero and then performing gradient descent on the new manifold using this as an initialisation. Hence, the result of a local optimisation in $n$ dimensions is used as the initialisation for optimisation in $n+1$ dimensions ensuring that the solution is already constrained to the right region of the manifold.

The nature of the hyperspherical manifold can be used to inform the stepsize used in the gradient descent optimisation. We assume that the result in $n$ dimensions has restricted the solution to the correct hemisphere of the hypersphere. Travelling in the direction of the negative gradient reduces the error. To travel in this direction whilst remaining in the same hemisphere means the maximum arc distance that can be moved is $\frac{\pi}{2}$. Hence, the result in $n$ dimensions is given by $\mathbf{x}^{(n)} = h(\gamma^*)$, where

$$h(\gamma) = \mathrm{Exp}_{\Phi^{-1}(\mathbf{x}_{\text{init}}^{(n)})}\left(-\gamma\frac{\nabla\varepsilon\left(\Phi^{-1}(\mathbf{x}_{\text{init}}^{(n)})\right)}{\left\|\nabla\varepsilon\left(\Phi^{-1}(\mathbf{x}_{\text{init}}^{(n)})\right)\right\|}\right). \tag{4.22}$$

The arc distance $\gamma$ determines how far we travel along the great circle implied by the gradient of the objective function. Since we wish to constrain our solution to the same hemisphere, $\gamma$ must lie in the interval $\left[0, \frac{\pi}{2}\right]$ and we hence find $\gamma^*$ using golden section search [52] to solve: $\gamma^* = \arg\min_{\gamma} h(\gamma),\ 0 \leq \gamma \leq \frac{\pi}{2}$. Multiple iterations of gradient descent can be used each time a dimension is added to the optimisation. In our results we use four iterations per dimension.

## 4.7 Model Fitting Examples

For our experimental evaluation, we use the algorithm described above to fit our 3D morphable shape model to unseen data. We show results for two different objective functions and compare our results with those obtained using two different generic optimisers.

### 4.7.1 Overconstrained Optimisation

We choose as an objective function the angular error between surface normals at each vertex of the model. This is an interesting choice of objective function for two reasons. First, the search landscape of the objective function is littered with local minima. Second, the fitted result is likely to have lower perceptual error than a least squares fit directly to the vertices. Whilst such a least squares fit gives minimal geometric error, the result is often a gross over-fit which does not resemble the input face. Minimising the surface normal error is a non-linear problem which is related to minimising appearance error, as undertaken by analysis-by-synthesis of image data [15].

From an input face shape, represented by $p$ vertices, we compute surface normals at each vertex [68]. If $\mathbf{N}^i$ is the surface normal at vertex $i$, our objective function is the sum of squared angular errors between input and model surface normals:

$$\varepsilon(x) = \sum_{i=1}^{p} \left(\arccos(\mathbf{n}^i(\Phi(x)) \cdot \mathbf{N}^i)\right)^2, \tag{4.23}$$

where $\mathbf{n}^i([x_1 \ \dots \ x_n])$ is the surface normal of the $i$th vertex of the shape given by: $\bar{\mathbf{s}}+\mathbf{Pc}$, where the parameter vector is computed by transforming the unit vector back to the hyperellipse:

$$\mathbf{c} = \sqrt{n}\left[x_1\sqrt{\lambda_1} \ \dots \ x_n\sqrt{\lambda_n}\right]^T. \tag{4.24}$$

We compare our manifold optimisation with direct optimisation of the objective function using a generic optimiser based on the BFGS Quasi-Newton method with a cubic line search [20]:

$$\mathbf{c}^* = \arg\min_{\mathbf{c}} \sum_{i=1}^{p} \left(\arccos(\mathbf{n}^i(\mathbf{c}) \cdot \mathbf{N}^i)\right)^2. \tag{4.25}$$

Note that the generic optimiser converges close to the mean if all parameters are optimised simultaneously. We therefore take the same coarse-to-fine approach as for the manifold fitting, whereby we iteratively increase the number of dimensions considered in the optimisation.

We provide results on the BFM [76] data. The scans were obtained obtained using the structured light scanning system provided by ABW-3D [76] and are set into correspondence using a modified version of the Optimal Step Nonrigid ICP Algorithm [6]. In Figure 4.11, column (a) shows input bump maps for three unseen subjects. Column (b) shows the ground truth shape estimates. Column (c) shows the result of using the BFGS non-linear optimiser to solve Equation 4.25. Because of local minima close to the mean, these faces are implausibly smooth. Finally, our manifold fitting result is shown in column (d). Note that this result represents a trade off between over and underfitting. Table 4.1 tabulates the angular error of the surface normals averaged over all the out-of-sample faces in the BFM.

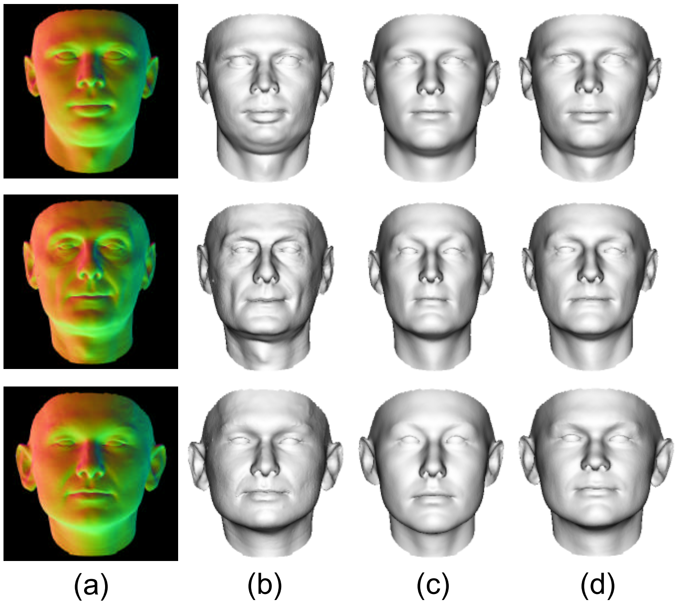

Figure 4.11: Overconstrained model fitting example: (a) input bump maps for 3 unseen subjects; (b) ground truth; (c) BFGS optimisation; (d) manifold optimisation. All the results are for a $n = 99$ parameter model.

| Method | Angular Error (in °) |
|---|---|
| BFGS method | 7.23 |
| Our method | 5.33 |

Table 4.1: Model fitting results (overconstrained objective).

### 4.7.2 Underconstrained Optimisation

We now consider an objective function which is highly underconstrained. In other words, solutions which minimise the objective function lead to highly implausible faces. The problem we consider is estimation of a high resolution 3D face surface given the positions of $k = 70$ 2D annotations ($k << p$). A linear version of this problem has been considered previously [12], where it was observed that the problem leads to a trade off between the quality of fit to the observed data and prior probability as measured by the model. The parameter to control this trade off can be determined heuristically [12], although no single value will give optimal performance for all faces. In contrast, our proposed approach requires no such regularisation constraint and ensures that the fitted results have high quality shape estimates which are plausible.

Our aim is to recover face shape parameter estimates from a set of $k$ 2D annotations. We represent the $i$th observed feature point by $\mathbf{L_i} \in \mathbb{R}^{\mathbf{2}}$. We write $\mathbf{r}^i(\Phi(x)) \in \mathbb{R}^4$ for the 3D position of the vertex corresponding of the $i$th feature point represented in homogeneous coordinates. This is extracted from the model shape vector given by: $\bar{\mathbf{s}} + \mathbf{Pc}$, where the parameter is vector $\mathbf{c}$ is computed by transforming the unit vector $\Phi(x)$ back to the hyperellipse using Equation 4.24. The corresponding projected 2D position is given by:

$$\hat{\mathbf{L_i}} = [e_i/g_i \ f_i/g_i]^T , \tag{4.26}$$

where $[e_i \ f_i \ g_i]^T = \mathbf{C}\mathbf{r}^i(\Phi(x))$ and $\mathbf{C} \in \mathbb{R}^{3\times4}$ is a camera matrix [44] which performs a perspective projection. Our objective function is taken by measuring the sum squared Euclidian distances between the observed 2D feature point positions and the projected model estimates:

$$\varepsilon(x) = \sum_{i=1}^{k} \|\mathbf{L_i} - \hat{\mathbf{L_i}}\|^{\mathbf{2}}. \tag{4.27}$$

The conversion from homogeneous to 2D Euclidian coordinates means that the error is a nonlinear function of the shape parameters. We assume that the camera matrix is known, since our aim here is to evaluate a simple underconstrained, nonlinear objective function. However, for a real world implementation this can be estimated using the Gold Standard algorithm [44] and the two steps of pose and shape estimation iterated to convergence.

We compare our manifold optimisation with direct optimisation of the objective function using a generic optimiser based on the Levenberg-Marquardt algorithm (LMA) [63]. Note that since the problem is underconstrained, direct optimisation of the objective function using LMA leads to gross overfitting. We therefore also provide results for the regularised version:

$$\mathbf{c}^* = \arg\min_{\mathbf{c}} \sum_{i=1}^{k} \|\mathbf{L}_i - \hat{\mathbf{L}_i}\|^2 + \eta D_M^2(\mathbf{c}), \tag{4.28}$$

where $\eta$ is a Lagrange multiplier which controls the influence of the regularisation term. It

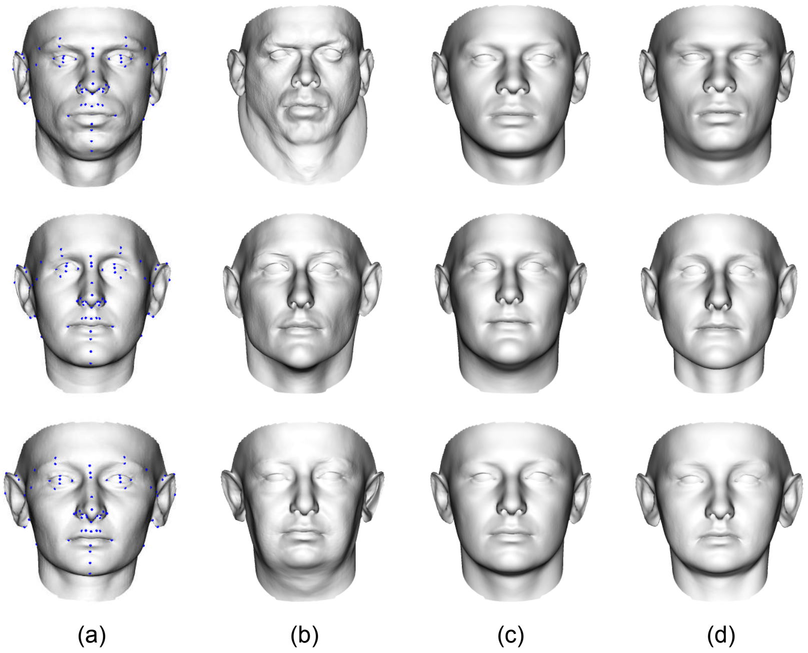

Figure 4.12: Underconstrained model fitting example: (a) ground truth faces (unseen) with the input feature points (blue circles); (b) LMA with no regularisation constraint; (c) LMA with regularisation constraint;(d) manifold optimisation. All the results are for a $n = 50$ parameter model.

should be noted in this case $\hat{\mathbf{L}_i} = \mathbf{C}\mathbf{r}^i(\mathbf{c})$.

In Figure 4.12 we show results on the BFM [76] data. Column (a) shows the ground truth faces (unseen) with the input feature points (blue circles). Column (b) shows the result of using LMA to solve Equation 4.28, with $\eta = 0$. In this case there is no regularisation constraint applied and hence we obtain grossly overfitted shape estimates. Column (c) shows the result of solving Equation 4.28 using LMA, with $\eta$ chosen experimentally to provide optimal average performance. To provide stable performance over all faces, the regularisation weight must be set conservatively which means some solutions are restricted to lie too close to the mean. The resulting faces therefore lack salient detail. Figure 4.13 demonstrates the effect of varying the regularisation weight. Finally, our manifold fitting result is shown in column (d). This result represents a trade off between over and underfitting via the hard manifold constraint. Our method does not require tuning of a parameter and provides stable performance in all cases. Table 4.2 tabulates the mean Euclidian error over all vertices in the mesh averaged over all the out-of-sample faces in the BFM.

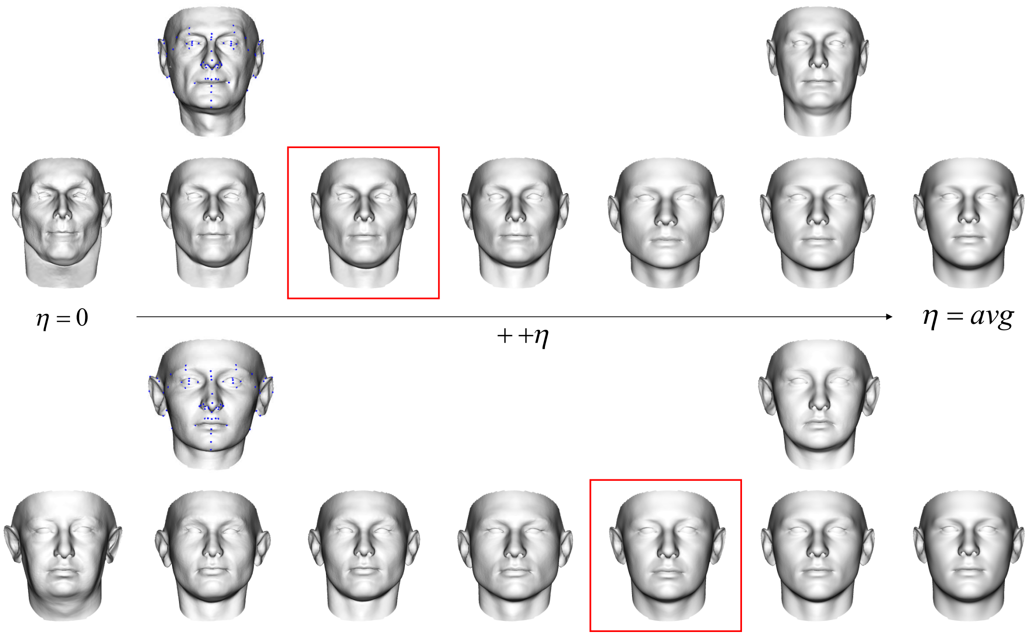

Figure 4.13: Demonstration of parameter selection for LMA optimisation of regularised objective. For two different subjects (shown on the left in the first and third rows), the optimal regularisation weight (corresponding to the shape estimate with the bounded box) is different. The manifold solution (shown on the right in the first and third rows) does not require parameter tuning and provides improved results.

| Method | Euclidian Error (in mm) |
|---|---|
| LMA without regularization | 5.68 |
| LMA with regularization | 4.81 |
| Our method | 4.21 |

Table 4.2: Model fitting results (underconstrained objective).

## 4.8 Conclusions

We have shown how a number of useful operations can be performed on the manifold of equally distinctive faces. This provides a new way to constrain operations involving the parameters of a statistical model. In particular, we have shown how to constrain the process of fitting a model to data which is robust but does not require the selection of a regularisation weight parameter. We avoid using a biased initialisation and improve efficiency by using a coarse-to-fine strategy. This approach outperforms the use of two generic nonlinear optimisation algorithms on two different objective functions.

# Chapter 5

# Model Simplification

## 5.1 Introduction

State of the art 3D morphable models have a very high resolution, typically $> 50000$ vertices [15, 76]. Fitting these high resolution models to unseen data, amounts to optimisation of complex nonlinear objectives. This iterative process is computationally expensive and requires significant amount of time to converge [15]. Depending on the resolution of the input data, or whether a coarse-to-fine strategy is employed, the use of such a high resolution model may be unnecessary and wasteful of computational resources. This provides motivation for the presented work.

In this chapter we show how to simplify a 3D morphable model without requiring knowledge of the shapes used to train the model (thus extending the simplification ideas in Chapter 2 into a statistical setting). Similar in spirit to the progressive mesh of Hoppe [47], we compute a multiresolution mesh structure which is optimal at each resolution with respect to the statistics of the original high resolution model. This allows us to define a continuous sequence of simpler versions of the morphable model, all of which are a subset of the original high resolution model. Our approach requires the storage of only a single mean mesh and modes of variation. The proposed method is based on iterative edge collapse and we show that the expected value of the Quadric Error Metric [36] can be computed in closed form for a PCA deformable model. We also present a variant of the approach which can applied to models with topologically symmetric mesh structure (though the shapes themselves need not be symmetric).

We show how the morphable models generated at different resolutions can be used to achieve super-resolution. By fitting a low resolution morphable model to data, we implicitly make a statistical estimate of the corresponding high resolution surface. We show that fitting a simplified 3D morphable model to data only slightly increases the fitting error (as compared to fitting the original model to the same data), while offering considerable improvement in the efficiency of model fitting.

## 5.2 Related Work

Simplification of individual triangular meshes is a well studied problem. Simplification algorithms can be classified according to how they modify a mesh and the mesh property they seek to preserve after modification. The former includes vertex decimation [90], vertex clustering [88] and edge contraction [36]. The latter includes geometric error, volume and appearance. However, the property which has proven both efficient to compute and to lead to simplified meshes with low geometric error is the Quadric Error Metric [36].

Simplifying sets of meshes which are in dense correspondence has only recently begun to receive attention amongst the graphics community [27, 56, 73]. Standard mesh operations such as simplification, remeshing, surface flattening or smoothing must be adapted to operate on a set of meshes whilst preserving dense correspondence. In [73] the authors extend the QSlim algorithm [36], such that the contraction edge is selected by summing over all the samples in the set. DeCoro and Rusinkiewicz [27] perform simplification of skeletally articulated meshes by incorporating knowledge of potential poses into a probability function which in turn determines the error quadrics associated with the surface. The idea is to obtain a mesh which approximates the likely poses more accurately than the improbable ones. However, they were unable to obtain a closed form solution for the error quadrics and resorted to Monte Carlo sampling. A similar approach is taken by Landreneau and Schaefer [56] who perform simplification based on a set of user-specified poses that are representative of typical deformations.

The solution strategy in [73] could be used to construct a multiresolution morphable model by applying it to the training set of meshes. A statistical model could be constructed at each resolution using the appropriately simplified meshes. However, this would require the storage of PCA statistics at every resolution. Moreover, the modes of variation computed at each resolution may be unrelated. The methods in [27] and [56] are not suitable for this application, since each mesh in the training set would have to be regarded as equally probable. In contrast (to [73]), the method we propose requires only knowledge of the 3D morphable model at the highest resolution. The mean shape and modes of variation remain the same and lower levels of detail are simply mesh structures which use a subset of the original model vertices. Edges for collapse are chosen according to the statistics of the model.

## 5.3 Contributions

First, we show how the Quadric Error Metric [36] can be formulated (in closed form) as a function of the model statistics. Second, we show the simplified models can be used very efficiently to achieve multiscale fitting and super-resolution. Finally, we provide evaluation for the model simplification and model fitting experiments.

## 5.4 Preliminaries

A 3D morphable model describes the variability of a set of meshes in dense correspondence. The model is obtained by applying PCA to the set of shape vectors $(x_1, y_1, z_1, \ldots, x_p, y_p, z_p)$, where $p$ is the number of vertices in the mesh. With a sufficiently large training set, an unseen shape can be accurately approximated as a linear combination of the average surface and the model eigenvectors:

$$\mathbf{s}(\mathbf{b}) = \bar{\mathbf{s}} + \sum_{l=1}^{n} b_l \mathbf{P}_l = \bar{\mathbf{s}} + \mathbf{P}\mathbf{b}, \tag{5.1}$$

where $\mathbf{b} = (b_1, \ldots, b_n)^T$ is a vector of shape parameters and $\bar{\mathbf{s}}$ the mean shape. We retain the $n$ most significant eigenvectors which are stacked to form the $3p \times n$ matrix $\mathbf{P}$. Additionally, each shape parameter $b_l$ is distributed according to a Gaussian with zero mean and variance given by the $l$th eigenvalue $\lambda_l$:

$$P(b_l) \sim e^{-\frac{1}{2}\left(\frac{b_l}{\sqrt{\lambda_l}}\right)^2}. \tag{5.2}$$

Similarly any unseen texture vector $\mathbf{t}$ can be represented as a linear combination of the average texture and the model eigenvectors:

$$\mathbf{t}(\mathbf{c}) = \bar{\mathbf{t}} + \sum_{l=1}^{n} c_l \mathbf{R}_l = \bar{\mathbf{t}} + \mathbf{R}\mathbf{c}, \tag{5.3}$$

where the vector $\mathbf{t} \in \mathbb{R}^{3p}$ contains the texture value at each of the $p$ vertices for each of the 3 colour channels. $\mathbf{R}$, $\bar{\mathbf{t}}$ and $\mathbf{c}$ are the texture eigenvectors, mean texture and texture parameter vector respectively. The texture parameters are normally distributed, similar to the shape parameters.

This statistical analysis is used to define a deformable mesh $\mathcal{M}(\mathbf{b}, \mathbf{c})^p = (K^p, \mathbf{s}(\mathbf{b}), \mathbf{t}(\mathbf{c}))$, which is a function of the shape and texture parameter vectors. The connectivity or topology of the deformable mesh is fixed and is given by the simplicial complex $K^p$ with $p$ vertices, which is a set whose elements can be vertices $\{i\}$, edges $\{i, j\}$ or triangles $\{i, j, k\}$, with the indices $i, j, k \in [1 \dots p]$. The actual shape of the mesh, given by the vector $\mathbf{s}(\mathbf{b}) \in \mathbb{R}^{3p}$, varies with the parameter $\mathbf{b}$, as in Equation 5.1. We denote the $i$th vertex as $\mathbf{v}^i \in \mathbb{R}^3$, which is composed of elements $3i - 2$, $3i - 1$ and $3i$ of the vector $\mathbf{s}(\mathbf{b})$. Similarly, the RGB vector $\mathbf{u}^i$ of the texture at the $i$th vertex is given by elements $3i - 2$, $3i - 1$ and $3i$ of the vector $\mathbf{t}(\mathbf{c})$.

## 5.5 Simplification of Morphable Models

Our starting point is a 3D morphable model built at a resolution $p$ as described by Equations 5.1 and 5.3. Our algorithm for simplification of a morphable model is based on iterative edge contraction. The key idea behind our method is to select an edge for contraction whose removal induces the minimum error over all possible shapes generated by the morphable model, weighted by their probability.

Simplified models comprise a subset of the vertices (and hence statistics) of the original model and a corresponding mesh topology. This ensures that the model statistics need only be computed once and that the mean face (shape and texture) and model eigenvectors at any resolution can be obtained by simply indexing the appropriate rows corresponding to the retained vertices. Consequently, a given shape and texture parameter vector can be used to generate a shape and texture at any of the model resolutions.

For this reason, edges must be contracted to one of the vertices at either end of the edge. This is not the optimum vertex placement policy if simplifying a mesh [36]. However, since we do not have knowledge of the individual meshes for simplification (we have only the model statistics), we are restricted to picking a subset of the original model and hence contracting an edge to either of its end vertices. In addition, if the initial morphable model is symmetric (i.e. every vertex has a corresponding symmetric partner), we would like to maintain this symmetry. In this case, we must adapt the cost function to select the pair of symmetric edges which together induce the minimum error.

Starting with the original high resolution morphable model, a sequence of edge contractions is applied until the simplification goals are satisfied. The algorithm generates a progressive sequence of model statistics which in turn can be used to generate meshes with decreasing resolution $(\mathcal{M}(\mathbf{b},\mathbf{c})^p, \mathcal{M}(\mathbf{b},\mathbf{c})^{p-1}, \mathcal{M}(\mathbf{b},\mathbf{c})^{p-2}, \ldots)$.

We consider only edge pairs as valid for contraction, i.e. where $\{i,j\} \in K^p$. We label the valid pairs at initialisation by constructing an edge table from the initial mesh structure. When we perform a contraction, the complex $K^p$ is modified. Degenerate faces (those that no longer have 3 distinct vertices) and duplicate edges are removed as well as the collapsed edge and redundant vertex $j$ [36]:

$$K^{p-1} = K^p \setminus \{\{j\}, \{i,j\}, \{j,\kappa\}, \{i,j,\kappa\} : \{i,j,\kappa\} \in K^p\}\,. \tag{5.4}$$

Finally, we need to keep track of the indices of the vertices in the original model which are retained after each edge contraction. This is a straightforward extension of the modification to simplicial complex $K$. Let $I^p = [1 \dots p]$ be a list of vertex indices at the highest resolution $p$. The updated index list at a lower resolution (after $\gamma$ edge contractions) would have a length $p - \gamma$ with its contents being the indices of the retained vertices upto that instance i.e. $I^{p-\gamma} = [g_1 \dots g_{p-\gamma}]$, with $g_1 \dots g_{p-\gamma} \in I^p$.

### 5.5.1 Quadric Error Metric as a Function of the Morphable Model Statistics

In order to select a pair for contraction at every iteration, we use the Quadric Error Metric proposed in [36]. We formulate the error quadrics as a function of the shape parameters ($\mathbf{b}$), the shape eigenvectors ($\mathbf{P}$) and the mean shape ($\bar{\mathbf{s}}$). We then integrate over the shape parameters, weighting by probability, to obtain the expected error quadric for each edge pair. It should be noted that the texture model plays no role in the computation of the Quadric Error Metric.

In order to compute the error quadrics, we make use of the assumption that each shape parameter ($b_l$) is an independent random variable that is normally distributed (Equation 5.2). Using this assumption we determine a fundamental quadric for each vertex based on each parameter ($b_l$). Hence each vertex will be associated with $n$ fundamental quadrics. A fundamental quadric for the $i$th vertex and $l$th parameter is defined as:

$$Q_l^i = (\mathbf{D}_l^i, \mathbf{e}_l^i, f_l^i) = \sum_{m=1}^{\text{planes}(i)} (\mathbf{D}_l^m, \mathbf{e}_l^m, f_l^m), \tag{5.5}$$

where $\mathbf{D}_l^m = \mathbf{n}_l^m \mathbf{n}_l^{mT}$ is a 3x3 matrix, $\mathbf{e}_l^m = d_l^m \mathbf{n}_l^m$ is a 3-vector and $f_l^m = d_l^{m2}$ is a scalar. $Q_l^i$ represents the set of triangular planes (defined by $[\mathbf{n}_l^{mT} d_l^m]^T$) that meet at the $i$th vertex. The summation is done using the component-wise additive rule [36].

Let $\mathbf{s}_l(b_l) = \mathbf{P}_l b_l + \bar{\mathbf{s}}$ be the shape obtained by deforming the mean shape according to the $l$th eigenvector by an amount given by $b_l$. Here, $\mathbf{P}_l$ is the $l$th column of the shape eigenvector matrix. The plane equations needed to compute the initial quadrics $Q_l^i$ can be obtained from $\mathbf{s}_l(b_l)$. We incorporate the boundary penalty [36] into the initial quadrics for vertices lying on a discontinuity edge. This results in $\mathbf{D}_l^i$, $\mathbf{e}_l^i$ and $f_l^i$, as polynomials in $b_l$ with degrees 6, 7 and 8 respectively. We make the assumption that the triangular polygons of our deformable mesh have approximately uniform area prior to simplification. This allows us to avoid normalising the plane normals to unit length and hence leads to quadrics which are polynomials in $b_l$. It also enables efficient implementation of the component-wise addition of the quadrics. Our assumption is reasonable if the model training data is uniformly remeshed as part of the process of establishing dense correspondence. This holds for the morphable model we use in our experiments [76].

Once the initial $n$ quadrics for each of the $p$ vertices are computed, we can derive the quadric for each edge pair $\{i,j\}$ using the additive rule i.e. $Q_l^{i+j} = Q_l^i + Q_l^j$. The quadric $Q_l^{i+j}$ assigns a value $Q_l^{i+j}(\mathbf{v}_l^w)$ to the $w$th vertex by the second order equation:

$$Q_l^{i+j}(\mathbf{v}_l^w) = \mathbf{v}_l^{wT}\mathbf{D}_l^{i+j}\mathbf{v}_l^w + 2\mathbf{e}_l^{i+j^T}\mathbf{v}_l^w + f_l^{i+j}, \tag{5.6}$$

where $w$ takes the values $i$ and $j$, the two end points of the edge pair $\{i,j\}$. Substituting and rearranging results in a polynomial in $b_l$ with degree 8:

$$Q_l^{i+j}(\mathbf{v}_l^w) = \delta_{8l}{b_l}^8 + \delta_{7l}{b_l}^7 + ... + \delta_{0l}, \tag{5.7}$$

where $\delta_{8l}, \delta_{7l}, ..., \delta_{0l}$ are constants. These constants are determined by the statistical shape model (i.e. the actual values of the eigenvectors and mean shape). The formulas for computing these constants are obtained using the MATLAB Symbolic Math Toolbox.

Since $Q_l$ is a continuous function in $b_l$, we determine the quadric error for the edge pair $\{i,j\}$ as:

$$E_l^{\{i,j\}}(\mathbf{v}_l^w) = \int_\alpha^\beta P(b_l)Q_l^{i+j}(\mathbf{v}_l^w)\,db_l. \tag{5.8}$$

The total contraction error for the edge pair $\{i,j\}$ can be obtained by summing over the error resulting from the $n$ independent shape parameters:

$$E^{\{i,j\}}(\mathbf{v}^w) = \sum_{l=1}^{n} E_l^{\{i,j\}}(\mathbf{v}_l^w). \tag{5.9}$$

Using Equations 5.2 and 5.7 we have:

$$E^{\{i,j\}}(\mathbf{v}^w) = \sum_{l=1}^{n} \int_{\alpha}^{\beta} e^{-\frac{1}{2}\left(\frac{b_l}{\sqrt{\lambda_l}}\right)^2} \left(\delta_{8l} {b_l}^8 + ... + \delta_{0l}\right)\, db_l. \tag{5.10}$$

This integral has a closed form solution in terms of error functions. We evaluate the definite integral over the interval $[\alpha, \beta] = [-3\sqrt{\lambda_l}, 3\sqrt{\lambda_l}]$. This range captures $> 99\%$ of the variance. Increasing the interval beyond this point has a negligible effect on the error values. Keeping in mind our need to retain either ends of the contracted edge, we store the error associated with an edge pair $\{i, j\}$ as:

$$\Gamma^{\{i,j\}} = \min\left(E^{\{i,j\}}(\mathbf{v}^i), E^{\{i,j\}}(\mathbf{v}^j)\right). \tag{5.11}$$

We select the contraction edge pair to be the one that minimises Equation 5.11:

$$\{i^*, j^*\} = \underset{\{i,j\} \in K^{p-\gamma}}{\arg\min}\ \Gamma^{\{i,j\}}. \tag{5.12}$$

After every edge contraction we must update the quadrics and the error heap for the modified entries in the complex $K$ [36].

### 5.5.2 Symmetric Model

If the original model has symmetric topology, then we would like to ensure that the symmetry is preserved during simplification. We would like to select a symmetric pair of edges for contraction at each iteration. Assuming the highest resolution model is symmetric, then for each edge $\{i, j\}$, there is a corresponding symmetric edge $\{\breve{i}, \breve{j}\}$. In the case where $\{i, j\}$ lies on the line

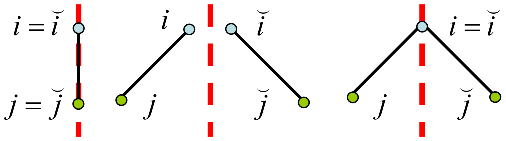

Figure 5.1: Case 1 (left): The edge lies on the line of symmetry. In this case only one vertex is removed during the symmetric edge pair contraction. Case 2 (Center): The symmetric edge pair lies away from the line of symmetry. This is the most common case encountered. Case 3 (Right): One vertex of the symmetric edge pair lies on the line of symmetry. In this case we always retain the vertex pair lying on the line of symmetry.

of symmetry, then $\{\breve{i}, \breve{j}\} = \{i, j\}$. We select a symmetric edge pair for contraction as follows:

$$\{i^*, j^*\}, \{\breve{i}^*, \breve{j}^*\} = \underset{\{i,j\},\{\breve{i},\breve{j}\} \in K^{p-\gamma}}{\arg\min} \Gamma^{\{i,j\},\{\breve{i},\breve{j}\}}, \tag{5.13}$$

where:

$$\Gamma^{\{i,j\},\{\breve{i},\breve{j}\}} = \min\left(E^{\{i,j\}}(\mathbf{v}^i) + E^{\{\breve{i},\breve{j}\}}(\mathbf{v}^{\breve{i}}), E^{\{i,j\}}(\mathbf{v}^j) + E^{\{\breve{i},\breve{j}\}}(\mathbf{v}^{\breve{j}})\right). \tag{5.14}$$

This symmetric version of the algorithm can be implemented efficiently using a symmetric modification of the edge list structure. It should be noted that in the symmetric implementation, during each iteration, $\gamma$ (the number of vertices removed) is incremented by 2. However, in the case where $\{\breve{i}, \breve{j}\} = \{i, j\}$, only one vertex is removed during the contraction.

We need to make a special consideration if the symmetric edge pairs have only one vertex lying on the line of symmetry (either $i, \breve{i}$ or $j, \breve{j}$). In order to preserve the line of symmetry, we need to enforce that the retained vertices in this case are the ones lying on the line of symmetry (i.e. set Equation 5.14 to the error value associated with the symmetric vertex pair lying on the line of symmetry). Figure 5.1, provides a visual illustration of the 3 types of symmetric edge pairs that are encountered. Figure 5.2, shows an example of the symmetry preserving implementation of the algorithm.

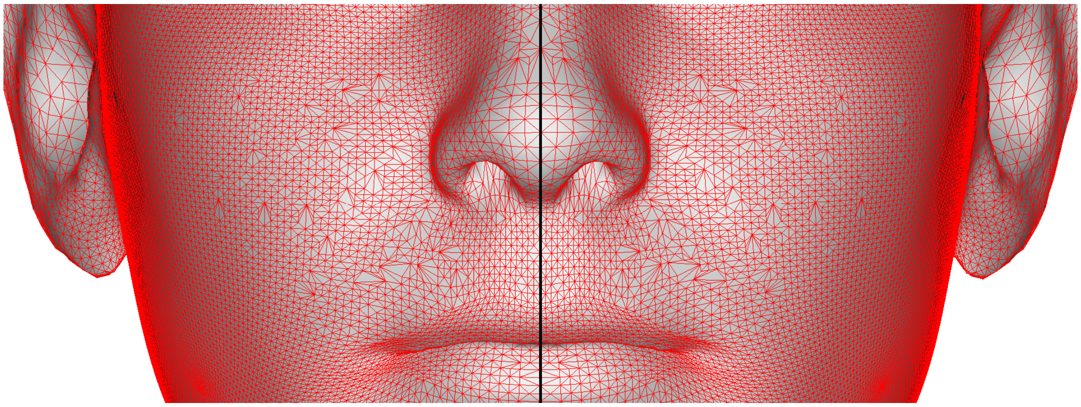

Figure 5.2: Cross-section of a face showing the symmetry preserving implementation of the algorithm.

### 5.5.3 Algorithm Output

After $\gamma$ edge contractions, the algorithm outputs the simplicial complex of the simplified mesh structure $K^{p-\gamma}$ and the list of retained vertex indices $I^{p-\gamma}$. Using $I^{p-\gamma}$ we can obtain the shape and texture morphable models at the resolution $p-\gamma$ by simply indexing the relevant rows in $\mathbf{P}^p$, $\bar{\mathbf{s}}^p$, $\mathbf{R}^p$ and $\bar{\mathbf{t}}^p$ corresponding to each entry of $I^{p-\gamma}$. The simplified morphable model and mesh structure can then be used to generate lower resolution meshes, $\mathcal{M}(\mathbf{b},\mathbf{c})^{p-\gamma}$, using the same shape and texture parameters as the original model.

### 5.5.4 Computational Complexity

Similar to [73], our method adds an overhead to the computational complexity of the QSlim method for a single mesh [36]. A single edge contraction requires, $O(n)$ operations to recompute the quadric error in the neighborhood of the contracted edge, plus $O(\log p)$ operations to maintain the heap property. As the number of contractions is bounded by $p$, the iterative contraction phase runs in $O(np + p\log p)$ time.

## 5.6 Multiresolution Model Fitting

Applications which make use of a 3D morphable model for analysis or synthesis, in particular those which involve fitting the model to image data, are computationally expensive [15]. The main reason for this is the resolution at which the state-of-the-art models are built. For example, the Basel Face Model (BFM) [76] is built using scans with $p = 54390$ vertices. For this reason, there is strong motivation for developing multiscale fitting algorithms which vary the resolution of the model in a way which is appropriate to the data. Essentially, our simplified models can be plugged in as appropriate to any morphable model fitting algorithm with a resulting improvement in efficiency. Our approach to simplification of a morphable model lends itself well to multiscale fitting and super-resolution. Since the simplified model is a subset of the original high resolution model, shape parameter vectors can be used interchangeably between models at different resolutions.

Given an unseen 3D shape $(\mathbf{s}_{\text{test}}^{p-\gamma})$ having $p - \gamma$ vertices, using the eigenvectors $(\mathbf{P}^{p-\gamma})$ and mean shape $\bar{\mathbf{s}}^{p-\gamma}$ (the shape model at this resolution), we can obtain a parametric shape estimate that is optimum in a least squares sense:

$$\mathbf{b}^* = \left(\mathbf{P}^{p-\gamma}\right)^{-1}\left(\mathbf{s}_{\text{test}}^{p-\gamma} - \bar{\mathbf{s}}^{p-\gamma}\right), \tag{5.15}$$

where $(\mathbf{P}^{p-\gamma})^{-1}$ is the pseudo inverse of $\mathbf{P}^{p-\gamma}$. It should be noted that at a resolution $< p$, the reduced eigenvectors are no longer orthonormal, hence the use of the pseudo inverse rather than the transpose. We can now compute the corresponding shape at the highest resolution as:

$$\mathbf{s}_{\text{rec}}^{p}(\mathbf{b}^*) = \mathbf{P}^p \mathbf{b}^* + \bar{\mathbf{s}}^p. \tag{5.16}$$

Combining the above, the least squares estimate of a high resolution face given a low resolution observation (i.e. super-resolution) is given by:

$$\mathbf{s}_{\text{rec}}^{p}(\mathbf{b}^*) = \mathbf{P}^p\left(\mathbf{P}^{p-\gamma}\right)^{-1}\left(\mathbf{s}^{p-\gamma} - \bar{\mathbf{s}}^{p-\gamma}\right) + \bar{\mathbf{s}}^p. \tag{5.17}$$

In Section 5.7.2, we provide an analysis of the shape reconstruction error between the original and reconstructed meshes over a range of input resolutions (i.e. super-resolution). We compare

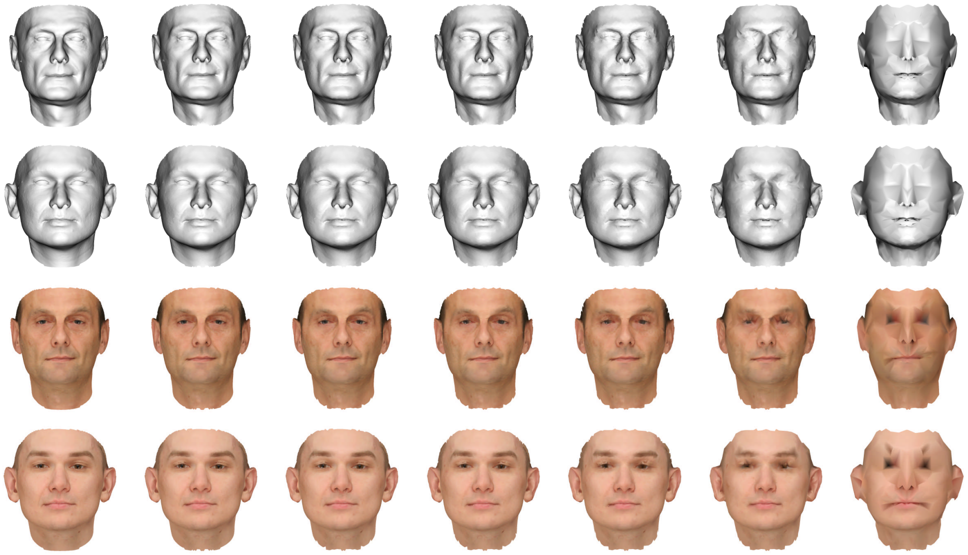

Figure 5.3: Simplification results for 2 BFM test meshes at different resolutions. The resolution from left to right: 53490, 25000, 20000, 15000, 10000, 5000 and 1000 . The first 2 rows show results for 3D shape and the last two rows show results for 3D shape and texture.

this with the model best fit to the original high resolution input (i.e. the generalisation error of the model). It should be noted that exactly the same derivations hold for the texture model.

Besides the linear least squares estimates, a more challenging application of our simplified models is to study the relationship between fitting quality and computational complexity when lower resolution models are used to optimise a more complex, nonlinear objective function. Many such objective functions have been proposed and our simplified model could be used in conjunction with any one of these. For our evaluation, we choose a simple exemplar objective function to highlight the effect of model simplification on the fitted result. Namely, the sum of squared angular errors between the input and model surface normals:

$$\mathbf{b}^* = \arg\min_{\mathbf{b}} \sum_{l=1}^{p-\gamma} \left(\arccos(\mathbf{n}^l(\mathbf{b}) \cdot \mathbf{N}^l)\right)^2, \tag{5.18}$$

where $\mathbf{N}^l$ is the input surface normal at vertex $l$ and $\mathbf{n}^l(\mathbf{b})$ is the $l$th surface normal corresponding to the shape estimate ($\mathbf{b}$). We choose this objective function for two reasons. First, the search landscape of the objective function is littered with local minima. Second, minimising the surface normal error is a nonlinear problem which is related to minimising appearance error,

| $\gamma$ | $p-\gamma$ | Our Method | | Method in [73] | |
|---|---|---|---|---|---|
| | | Mean | RMS | Mean | RMS |
| 28490 | 25000 | 0.0140 | 0.0369 | 0.0355 | 0.3947 |
| 33490 | 20000 | 0.0232 | 0.0626 | 0.0617 | 0.4775 |
| 38490 | 15000 | 0.0406 | 0.1230 | 0.0834 | 0.4987 |
| 43490 | 10000 | 0.0739 | 0.1956 | 0.1186 | 0.5443 |
| 48490 | 5000 | 0.2062 | 0.4788 | 0.2087 | 0.6828 |

Table 5.1: Mean and RMS distances between the highest resolution meshes and the simplified meshes. All distances are in mm.

as undertaken by analysis-by-synthesis of image data [15]. The problem also arises in fitting to photometric estimates of surface orientation. In Section 5.7.2 we show results of solving this objective function at different resolutions and compare results to the solution at the highest resolution.

## 5.7 Experimental Results

For our experimental evaluation, we use the BFM [76]. The BFM is a 3D morphable model trained on 200 faces and has a resolution of $p = 54390$ vertices. Using the method described in this chapter we can generate lower resolution models at any resolution with $p - \gamma$ vertices. We use the $n = 50$ most significant model eigenvectors (capturing 98.46% of the cumulative variance for the shape model) in our simplification algorithm. The BFM is a symmetric model and our simplification preserves this property.

### 5.7.1 Model Simplification

We begin by providing a quantitative analysis of how the low resolution meshes generated using the proposed algorithm compare to the highest resolution meshes. We compare our solution with the solution proposed in [73]. While our method uses the statistics of the trained morphable model to select an edge for collapse, in [73] the contraction edge is determined by summing the error metric over a set of meshes. We apply the method of [73], by summing over 50 meshes randomly generated using the BFM. The shape parameter vectors are generated by randomly

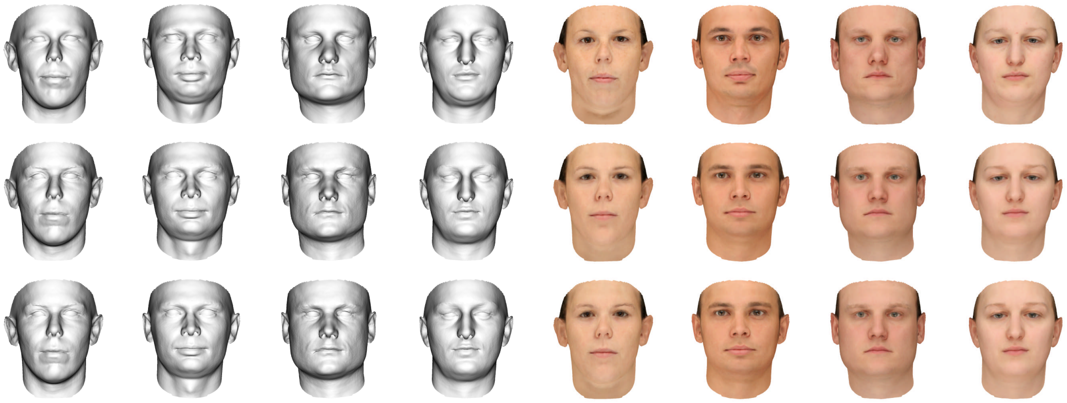

Figure 5.4: Least squares estimates for 4 unseen subjects. Row 1 shows the ground truth. Row 2 shows the estimates obtained using the model at the highest resolution. Row 3 shows the estimates obtained using the model at a resolution of 5000. The first 4 columns show 3D shape and the last 4 columns show 3D shape and texture.

sampling from the appropriate normal distribution for each parameter. Also for both methods we use the same vertex placement policy and symmetric implementation as described in Section 5.5.

To evaluate the simplified meshes, we use 10 out-of-sample test scans provided with the BFM. The scans were obtained using the structured light scanning system provided by ABW-3D [76] and are set into correspondence using a modified version of the Optimal Step Nonrigid ICP Algorithm [6]. We subselect the vertices corresponding to the simplified model and apply the corresponding low resolution mesh topology. The resulting low resolution mesh is compared to the original high resolution mesh. We provide mean and RMS distances computed using the Metro tool [23] in Table 5.1. We report the average of the forward and backward distances, averaged over the 10 meshes. It should be noted that the 10 test meshes are disjoint from the ones used to build the BFM or the 50 randomly generated meshes. The table shows that our solution results in the selection of more class-optimal edges to be contracted as compared to the solution strategy proposed in [73].

The effect of our vertex placement policy becomes evident when low resolution meshes in the model are visualised. In Figure 5.3 we show two of the BFM test meshes at various resolutions. Because we force edges to collapse to one of the vertices at either end, meshes at resolutions

| $\gamma$ | $p-\gamma$ | $\frac{p}{p-\gamma}$ | $\frac{E_s^h}{E_s}$ | $\frac{E_t^h}{E_t}$ |
|---|---|---|---|---|
| 28490 | 25000 | 2.1396 | 0.9950 | 0.9913 |
| 33490 | 20000 | 2.6745 | 0.9768 | 0.9871 |
| 38490 | 15000 | 3.5660 | 0.9509 | 0.9716 |
| 43490 | 10000 | 5.3490 | 0.9080 | 0.9365 |
| 48490 | 5000 | 10.6980 | 0.8935 | 0.8812 |

Table 5.2: Quantitative analysis for the least square estimates. Here $E_s = \frac{1}{p}\sum_{l=1}^{p}\|\mathbf{v}_{\text{test}}^l - \mathbf{v}_{\text{rec}}^l(\mathbf{b}^*)\|$ and $E_t = \frac{1}{p}\sum_{l=1}^{p}\|\mathbf{u}_{\text{test}}^l - \mathbf{u}_{\text{rec}}^l(\mathbf{c}^*)\|$. The units for $E_s$ and $E_t$ are mm and normalised RGB intensity respectively. $h$ denotes the analysis for the highest resolution morphable model, with $E_s^h = 0.9810$ and $E_t^h = 0.0372$. We use 199 parametric modes for the reported analysis.

$< 5000$ begin to lose salient facial details and start developing visual artefacts. Qualitatively, an acceptable perceptual output is possible for simplifications up to a factor of $\sim 10$.

### 5.7.2 Model Fitting

The second aspect of our evaluation investigates the trade off between quality of fit and efficiency (computational times) when fitting simplified models to data.

In the first experiment we provide results for the least squares fit of the model to unseen input meshes at a variety of resolutions (see Equation 5.17). We provide results for both shape and texture in comparison to a least squares fit at the highest resolution. The quantitative results are given in Table 5.2. All error measures are with respect to ground truth and we report error measures averaged over the BFM test set. Figure 5.4 shows the visual illustration of the least squares estimates for 4 unseen subjects. The results show that our low resolution morphable models can reliably estimate unseen data when the resolution is reduced by upto a factor of $\sim 10$.

Next we provide results for fitting the morphable models to optimise a more complex, non-linear objective function (Equation 5.18). For our experiment we use as input, out-of-sample normal maps obtained using the ABW-3D scanner [76]. Once again correspondence is established using a modified version of the algorithm of [6]. From the input 3D surface, we compute surface normals at each vertex [68]. We begin by optimising the objective function using the

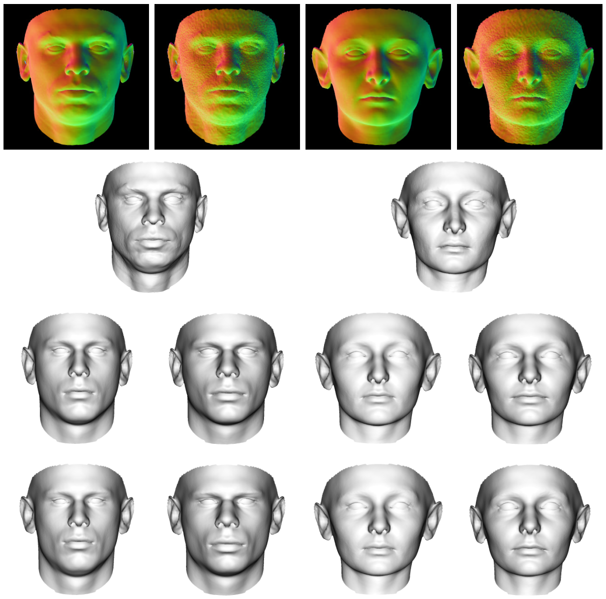

Figure 5.5: Model fitting estimates (optimising Equation 5.18). Row 1 shows the input bump maps (noise free and noisy) for 2 unseen subjects. Row 2 shows the ground truth. Row 3 shows the shape estimates obtained using the model at the highest resolution. Row 4 shows the shape estimates obtained using the model at a resolution of 5000.

original high resolution morphable model. We then use a morphable model at a much lower resolution (10000 and 5000) to optimise the same objective function, followed by generating the highest resolution output using Equation 5.16. We analyse the quality of fittings obtained and computational times required in the two cases. Note that at lower resolutions the input normals are obtained by subselecting the vertices corresponding to the simplified model and using the corresponding low resolution mesh topology. We use the coarse-to-fine model fitting strategy suggested in Chapter 4 to optimise an $n = 50$ parameter model (at all resolutions).

We also provide results for fitting to noisy normal maps. In this case we add noise to perturb the 3D geometry of the scans set in correspondence before computing the vertex normals. For our experiment, we randomly sample the noise from a normal distribution having zero mean. The standard deviation of the noise distribution is set proportional to the extent of the 3D geometry and ranges from 0.1 mm to 0.3 mm for the test data used. Table 5.3 reports the quantitative findings of our experiment. We report all results averaged over the BFM test set. Figure 5.5 shows the qualitative results for the experiment. In general, our low resolution models

| $\gamma$ | $p-\gamma$ | $\frac{E_a^h}{E_a}$ | $\frac{E_a^h}{E_a}$ (Noisy) | $\frac{\tau^h}{\tau}$ |
|---|---|---|---|---|
| 43490 | 10000 | 0.9558 | 0.9494 | 3.0888 |
| 48490 | 5000 | 0.9046 | 0.9176 | 3.7623 |

Table 5.3: Quantitative analysis for a model fitting application. $E_a$ is the mean angular error between the ground truth and fitted surface normals (in degrees) and $\tau$ is the computational time (in minutes) required for optimising the objective function (Equation 5.18). $h$ denotes the analysis for the highest resolution morphable model, with $E_a^h = 5.135$ (for noiseless input), $E_a^h = 6.203$ (for noisy input) and $\tau^h = 12.433$. All computations are done on a 1.78 GHz AMD Athlon processor.

(shape, texture or a combination of both) can be used to achieve a considerable reduction in computational times at a small compromise in the quality of fitted estimates.

## 5.8 Conclusions

In this chapter we have shown how to simplify a dense statistical 3D surface model by formulating the Quadric Error Metric as a function of the model statistics. Our simplified models can be used for super-resolution and to improve the efficiency of model fitting. In theory this work can be extended by including the texture model statistics in the formulation of the Quadric Error Metric (along the lines of [37]). However, doing so may be impractical from the point of view of implementing a closed form solution as it would result in a high degree polynomial in two variables (shape and texture parameters). Finally, since our vertex placement policy requires an edge to collapse to one of the vertices at either end, employing vertex decimation [90] could prove to be an effective strategy.

# Chapter 6

# Shape-from-Shading Driven Morphable Models

## 6.1 Introduction

Estimating 3D face shape from single images has several applications, primarily as a route to pose and illumination invariant face recognition. A considerable body of work has focused on this problem in recent years [8, 15, 39, 58, 85, 86, 103, 106]. Recent advances in this area can be divided into two distinct avenues of research. The first uses a morphable model of facial appearance [15, 103] while the second uses classical shape-from-shading techniques [81, 94].

The body of work which uses morphable models offers globally accurate and robust shape recovery from real images, but the recovered shape lacks local detail and only weakly satisfies image irradiance constraints (and hence photorealism). Recent results in shape-from-shading have shown that fine local surface detail can be recovered from face images but that the estimated global 3D shape is of inferior quality to that recovered using a morphable model. The aim in this chapter is to explore an approach which combines these currently divergent strands of research in the hope of retaining the benefits of both.

## 6.2 Related Work

### 6.2.1 3D Morphable Models

To solve the problem of face shape recovery using a morphable model, the challenge is to fit the model to images of previously unseen subjects. This amounts to solving a highly complex non-linear minimisation problem which requires estimation of: 1. the shape and texture parameters,

2. pose, scale and position of the subject, 3. camera and surface reflectance parameters and 4. the illumination conditions present in the scene. Such an approach provides 3D face shape estimates that are globally accurate. Importantly, the method is robust under real world conditions and explicitly models the underlying physical processes which give rise to the observed image. However, several key problems hinder further development of the technique. Model dominance is the most serious weakness. Morphable models provide a face shape recovery approach in which the quality of the reconstructed shape is entirely dependent on whether the model was trained using data which contained similar examples. Hence, fine surface detail or atypical surface features (often the most distinguishing elements of the face) are lost. No cues are exploited from within the image to directly recover shape. Instead, the only information extracted from the image is the cost value associated with the current parameter estimates. Minimising this cost function requires a highly complex and computationally expensive search, with no guarantee of obtaining the global minimum. In fact, Blanz and Vetter [15] had to employ stochastic minimisation techniques to avoid finding local minima.

Model dominance is also a problem in the estimation of surface texture or more precisely albedo. To date, albedo has been modelled, as for shape, using a linear statistical model [15, 71, 87, 103]. It is not clear how features such as moles, freckles and facial hair can be efficiently captured and hence recovered using such a model. In addition, obtaining meaningful albedo values with which to construct the training set is challenging. For example the Cyberware 3030PS scanner used to collect data [15] measures a texture map using digital cameras in the scan head. However, these texture measurements are illumination dependent and do not correspond to a physically meaningful albedo value.

More recently the morphable model of Blanz and Vetter [15] has been extended using multilinear face models to study and synthesize variations in faces along several axes, such as identity, expression and visemes [97] or to use statistical models for synthesis of detailed facial geometry [40]. Other methods for capturing fine scale details of human faces are also well researched

[99, 110]. Even though these methods would enable us to learn fine structural details and attributes, these methods require considerably more effort in the building of the models and the models would still be able to recover only what they were trained on.

The most recent work in this area of fitting morphable models has focused on developing more sophisticated morphable model fitting algorithms. Zhang and Samaras [103], use the fit of a morphable model to each gallery image to obtain a spherical harmonic basis (9D) for that subject. This is also done in a model based manner by learning a statistical model of spherical harmonic basis (assuming a Lambertian reflectance model). The proposed method can represent a face under arbitrary unknown lighting and pose simply by four low-dimensional vectors, i.e. shape parameters, pose parameters, spherical harmonic basis parameters and illumination coefficients. In order to improve performance when the images are taken under an extreme lighting condition, a subregion based framework that uses a Markov random field to model the statistical distribution and spatial coherence of face texture was used in [98]. In [57] the authors use a 30 basis Bilinear Illumination Model (BIM) to create a compact illumination subspace given 3D morphable model shape estimates.

All of these methods are based on explicitly modelling the underlying physical processes that give rise to an observed image, by rendering each hypothesised appearance. It is not clear that this is either necessary or the most practical approach. As already mentioned, the only information extracted from the image is the cost value associated with the current parameter estimates. On the other hand, both Romdhani and Vetter [87] and Moghaddam et al. [71] focus on improving the accuracy and efficiency of the fitting process respectively by using features derived from the input images rather than the intensity data itself. Romdhani and Vetter [87] used edges and specular highlights to obtain a smooth cost function, while Moghaddam et al. [71] used silhouettes computed from a large number of input images. In order to overcome model dominance, Knothe et al. [54] use local feature analysis to locally improve the fit of the model to a set of sparse feature points.

### 6.2.2 Shape-from-Shading

In contrast to fitting a morphable model, where the problem reduces to minimisation of a complex error functional, the idea in shape-from-shading is to make an estimate of the surface orientation at every pixel directly from the image intensities [105]. This approach is data-driven in that the intensity information at every pixel in the input image is used and by satisfying hard irradiance constraints, it can be ensured that the estimated shape recreates the input image exactly.

Unfortunately single image shape-from-shading has proved ineffective (compared to other methods such as photometric stereo [39]), in recovering realistic 3D shape. Several authors have shown that the accuracy of the recovered shape information can be greatly enhanced by using domain specific constraints. For instance, Prados and Faugeras [81] use the location of singular points to enforce convexity on the recovered surface. Zhao and Chellappa [107], on the other hand, have introduced a geometric constraint which exploits the approximate bilateral symmetry of faces. Dovgard and Basri [28] combined the statistical constraint of [7] and the geometric constraint of [107] into a single shape-from-shading algorithm.

More recently, Smith and Hancock [94] show how to recover a field of surface normals (a needle-map) using a statistical model of variations in surface normal direction. In contrast to a morphable model, their statistical model is view-dependent and does not consider correspondences between subjects. This makes projection of measured surface normals onto the model straightforward but limits the applicability of the model to registered fronto-parallel views. Without dense correspondence, it is also not clear how the model can efficiently characterise facial shape. Most recently, Ecker and Jepson [31] examine the shape-from-shading problem without boundary conditions as a polynomial system which they solve using semidefinite programming.

## 6.3 Contributions

In this chapter we propose a method for facial shape recovery from a single image which combines the strengths of both approaches (global robustness and recovery fine local detail) while tackling the weaknesses faced by current state-of-the-art methods that rely on fitting a 3D morphable model. In doing so, we make a number of contributions.

We begin by reviewing background theory of morphable models (Section 6.4) and shape-from-shading (Section 6.5). In Section 6.6, we then show how constraints on surface normal direction suggested by shading cues provide a route to exceeding the accuracy of face shape estimates obtainable using a morphable model. Our approach is based on linear optimisation of an error measure based on a combination of the geometric and surface normal errors. We show that reducing this error correlates with reduction in the appearance error of a recovered shape. Our method is able to reduce the error of a face reconstructed using a state-of-the-art analysis-by-synthesis algorithm and even improve upon the generalisation error induced by projecting an out-of-sample face onto the model. We also show that this refinement step can be used to improve upon the state-of-the-art recognition results.

We discuss the implications of this approach in Section 6.6.3 and consider how to incorporate the method into an iterative model fitting framework. We present our approach to do this in Section 6.7, where we describe an objective function which seeks morphable model parameters whose surface normals minimise the difference to those suggested by shape-from-shading. In addition to providing a new way in which to fit a morphable model, our algorithm also outputs per-vertex albedo and bump maps which are not constrained by the morphable model. These maps are able to capture fine detail of a face's surface shape and reflectance properties, without the lack of expressive power introduced by a statistical model. The extent to which shading variations are explained in terms of changes in surface orientation or albedo (i.e. 'shaded' versus 'painted' hypothesis) can be controlled by the degree of smoothness enforced on the albedo maps. We also show how the proposed method can be used for illumination insensitive face recognition. Finally, we present experimental results in Section 6.8.

## 6.4 3D Morphable Models

Consider a sample of $m$ 3D face meshes which are in dense correspondence. The $m$th shape is represented by a vector of $p$ vertices $\mathbf{x}_m \in \mathbb{R}^{3p}$. We use principal components analysis to obtain an orthonormal basis spanned by the $m$ eigenvectors $\mathbf{P}_m$. Any shape vector $\mathbf{x}$ may now be represented as a linear combination of the average shape and the model eigenvectors:

$$\mathbf{x} = \bar{\mathbf{x}} + \sum_m b_m \mathbf{P}_m, \tag{6.1}$$

where $\mathbf{b}$ is a vector of shape parameters. We stack the eigenvectors to form a matrix $\mathbf{P}$, such that we may write: $\mathbf{x} = \bar{\mathbf{x}} + \mathbf{Pb}$. The PCA eigenvalues $\lambda_m$ provide a measure of how much of the variance of the training data is captured by each eigenvector. We may choose to retain $n < m$ model dimensions, such that a certain percentage of the cumulative variance is captured.

The lengths of the parameter vectors (as measured by the square of the Mahalanobis distance from the mean) follow a chi-square distribution with $n$ degrees of freedom (Chapter 3):

$$D_M^2(\mathbf{b}) = \sum_n \left( \frac{b_n}{\sqrt{\lambda_n}} \right)^2 \sim \chi_n^2. \tag{6.2}$$

Such a distribution has a mean value of $n$ and variance $2n$. An interesting observation is that the chi-square distribution of parameter vector lengths implies that the parameter vectors lie approximately on the surface of a hyperellipsoid in parameter space. This observation suggests sensible constraints to enforce on the parameter vector lengths (Chapters 3 and 4).

## 6.5 Shape-from-Shading

The aim of computational shape-from-shading is to estimate 3D surface shape from single 2D intensity images. In order to recover surface orientation from image intensity measurements, the reflectance properties of the surface (human skin in our case) must be modelled. The bidirectional reflectance distribution function (BRDF) describes the ratio of the emitted surface radiance to the incident irradiance over all possible incident and exitant directions. Assuming the light enters and leaves the surface at the same point, the skin reflectance properties can be explained by a BRDF. A number of analytic reflectance models have been proposed that can capture a range of surface BRDFs. The simplest Lambertian model [16] predicts that light is scattered equally in all directions. The radiance function for such a surface is therefore:

$$g_{Lambertian}\left(\theta_i, \rho_d\right) = \rho_d \cos\left(\theta_i\right), \tag{6.3}$$

where $\rho_d$ is the diffuse albedo, which describes the intrinsic reflectivity of the surface. $\theta_i$ is the angle between the surface normal $\mathbf{n}$ and light source $\mathbf{s}$ vectors. Note that the reflected intensity is independent of the viewing direction. The Blinn-Phong reflectance model [16] is a phenomenological attempt to describe surfaces which reflect light both specularly and diffusely. It comprises a Lambertian diffuse term and a specular term controlled by the shininess parameter:

$$g_{Phong}\left(\theta_i, \theta_h, \rho_d, \rho_s, \eta_s\right) = \rho_d \cos\left(\theta_i\right) + \rho_s \cos^{\eta_s}\left(\theta_h\right), \tag{6.4}$$

where $\rho_s$ is the specular coefficient, $\eta_s$ is the shininess parameter and $\theta_h$ is the angle between the surface normal $\mathbf{n}$ and the vector $\mathbf{h} = \frac{\mathbf{s}+\mathbf{v}}{\|\mathbf{s}+\mathbf{v}\|}$ which bisects the light source $\mathbf{s}$ and viewer $\mathbf{v}$ directions.

The radiance function provides a succinct mapping between the reflectance geometry in the scene and the observed intensity ($I = g(\dots)$). This relationship is known as the image irradiance equation in the shape-from-shading literature. For an image in which the viewer and light source directions are fixed, the image irradiance equation reduces to a function of the surface normal direction. For typical reflectance models, this equation does not have an unique minimum and there are likely to be an infinite set of normal directions all of which minimise the equation.

The Lambertian reflectance model (Equation 6.3), provides a partial constraint on the direction of the surface normal, namely that the angle between the light source and the surface normal is given by:

$$\theta_i = \arccos\left(\mathbf{n}\cdot\mathbf{s}\right) = \arccos\left(\frac{I}{\rho_d}\right). \tag{6.5}$$

Geometrically this means that the surface normal must lie on a right circular cone whose axis is the light source direction and whose half angle is $\theta_i$. By constraining the surface normal to lie on the cone, the image irradiance equation is strictly satisfied, thereby ensuring that the information conveyed by the image is used to its fullest extent. Worthington and Hancock [100] show how to restore this constraint by rotating a surface normal to its closest on-cone position:

$$\acute{\mathbf{n}} = \Theta\left(\mathbf{a},\alpha\right)\tilde{\mathbf{n}}, \tag{6.6}$$

where $\tilde{\mathbf{n}}$ is an off-cone surface normal and $\Theta$ is a rotation matrix which rotates a unit vector about axis $\mathbf{a}$ by an angle $\alpha$. To restore a normal to the cone, we set $\mathbf{a} = \tilde{\mathbf{n}} \times \mathbf{s}$ and $\alpha = \theta_i - \arccos[\tilde{\mathbf{n}}\cdot\mathbf{s}]$. $\acute{\mathbf{n}}$ is the closest on-cone position that satisfies $\theta_i = \arccos\left(\frac{I}{\rho_d}\right)$.

For a surface which reflects light according to Equation 6.4, we can reformulate Equation 6.5 to provide an incident angle estimate which is consistent with the Blinn-Phong model:

$$\theta_i = \arccos(\mathbf{n}\cdot\mathbf{s}) = \arccos\left(\frac{I-\rho_s\cos^{\eta_s}\left(\theta_h\right)}{\rho_d}\right). \tag{6.7}$$

If we use this angle in the computation of the rotation in Equation 6.6, we strictly enforce the image irradiance constraint on a surface normal, by finding the closest surface normal direction that satisfies $\theta_i = \arccos\left(\frac{I-\rho_s\cos^{\eta_s}\left(\theta_h\right)}{\rho_d}\right)$.

## 6.6 Shading Constraints for Improving Morphable Model Shape Estimates

A morphable model allows us to represent a novel face using a linear combination of an orthonormal basis. Shape-from-shading enables us to modify an estimated set of surface normals such that they strictly satisfy constraints implied by the reflectance properties of the surface.

Given an estimate of the morphable model shape $\mathbf{x}$ (Equation 6.1), we can obtain the $p$th vertex normal $\mathbf{n}_p(\mathbf{x})$ [68]. Shading cues provide a constraint on the vertex normal direction which we can impose on the normals of a morphable model shape (hence directly utilising the image cue). Let:

$$\mathbf{n}_p^{\mathrm{oc}}(\mathbf{x}) = \Theta \mathbf{n}_p(\mathbf{x}), \tag{6.8}$$

be the rotated vertex normals. In the case a vertex is not visible (i.e we cannot sample the intensity value from the image), $\Theta$ is taken to be an identity matrix. The visibility of each vertex is tested with a z-buffer algorithm [34].

We would like to use the rotated vertex normals to improve the estimate of the shape and reduce the problem of model dominance. Using the approach suggested by Nehab et al. [74], a triangular mesh can be adjusted to match a set of target surface normals by minimising a sum of two error terms. The first measures the difference between the tangent plane normals of the refined shape and the target vertex normals:

$$E^{\mathrm{nc}} = \sum_p \sum_{u,w} [\mathbf{n}_p^{\mathrm{oc}}(\mathbf{x}) \cdot (\mathbf{x}_u^* - \mathbf{x}_w^*)]^2, \tag{6.9}$$

where $(u, w)$ are vertices of edges surrounding the vertex $p$. These edges are used to approximate the local tangent plane and the refined edges are encouraged to assume a direction which is perpendicular to the target vertex normal. The second term encourages a solution close to the original shape by penalising large departures:

$$E^{\mathrm{gc}} = \sum_p \|\mathbf{M}_p(\mathbf{x})(\mathbf{x}_p^* - \mathbf{x}_p)\|^2, \tag{6.10}$$

where $\mathbf{M}_p(\mathbf{x})$ is a $3 \times 3$ matrix that helps prevent self intersections in the optimised shape [74]. The refined shape $\mathbf{x}^*$ is given by:

$$\arg\min_{\mathbf{x}^*} \gamma E^{\mathrm{gc}} + (1 - \gamma) E^{\mathrm{nc}}. \tag{6.11}$$

This minimisation is linear in $\mathbf{x}$ and can be solved efficiently using sparse linear least squares. The parameter $\gamma \in [0, 1]$ controls the extent to which each error term is satisfied. A value of 0.1 gives stable performance which is not overly sensitive to either the intial shape or target normals [74]. The refined shape $\mathbf{x}^*$ is able to capture finescale local details of a face and does not suffer from model dominance.

### 6.6.1 An Error Measure for Evaluating Morphable Models

The ability of a morphable model to faithfully reconstruct an unseen shape is most commonly measured in terms of a geometric error [12], i.e. the average Euclidian distance between ground truth and reconstructed vertex positions. Such an error measure does not consider how well higher order properties of the surface are reconstructed, which determine the appearance of a mesh. It is this effect that has led to the development of many perceptual measures of error in the mesh processing literature. Using the measures described in Section 6.6, we can evaluate a reconstruction based on a combined term which penalises both geometric and surface normal errors. Let $\mathbf{x}^g$ be the ground truth shape of an unseen face. If $\mathbf{x}^e$ is the estimated shape, then we can evaluate the accuracy of the estimated shape with respect to the ground truth shape as:

$$E^{\text{comb}} = \kappa E^{\text{geo}} + (1 - \kappa) E^{\text{nor}}, \tag{6.12}$$

where

$$E^{\text{geo}} = \frac{1}{p} \sum_p \|\mathbf{x}_p^g - \mathbf{x}_p^e\|^2, \tag{6.13}$$

and

$$E^{\text{nor}} = \frac{1}{p} \sum_p \sum_{u,w} [\mathbf{n}_p(\mathbf{x}^g) \cdot (\mathbf{x}_u^e - \mathbf{x}_w^e)]^2. \tag{6.14}$$

These are analogous to Equations 6.9 and 6.10, with the term $\mathbf{M}_p$ no longer needed to evaluate $E^{\text{geo}}$ and the on-cone vertex normals replaced by the ground truth normals in $E^{\text{nor}}$. $\kappa \in [0, 1]$, is a scalar weight assigned to each error term.

We motivate this error measure by showing that it correlates well with appearance error. Appearance error measures how well the intensity at a vertex corresponds to ground truth when rendered under a variety of illumination conditions. We use the following appearance error:

$$E^{\text{app}} = \sum_{\zeta} \frac{1}{p} \sum_{p} \sum_{c} \left( \rho_d^c(\mathbf{n}_p(\mathbf{x}^g) \cdot \mathbf{s}_\zeta) \Upsilon(\mathbf{x}_p^g, \mathbf{s}_\zeta) - \rho_d^c(\mathbf{n}_p(\mathbf{x}^e) \cdot \mathbf{s}_\zeta) \Upsilon(\mathbf{x}_p^e, \mathbf{s}_\zeta) \right)^2, \tag{6.15}$$

where $c \in \{R, G, B\}$ and $\zeta$ is a set of randomly generated light source directions. We use $|\zeta| = 18$ directions to compute the appearance error. Our appearance error measure assumes a Lambertian reflectance model. $\Upsilon(\mathbf{x}_p, \mathbf{s}_\zeta) = 0$ if vertex $p$ is shadowed from direction $\mathbf{s}_\zeta$ and equals 1 otherwise. The measure can be modified to incorporate other reflectance models.

An interesting observation is that the appearance error also measures differences in geometry and surface normal direction. $\Upsilon$ is a function of the global geometry since a light source can be occluded by any other vertex in the mesh, while the Lambertian shading term is determined by the surface normal direction. In the following section we show that our combined error measure $E^{\text{comb}}$ correlates well with appearance error $E^{\text{app}}$ and hence any optimisation which reduces the combined error is likely to yield meshes with lower appearance error.

### 6.6.2 Quantative Analysis

We provide quantitative results on the Basel Face Model (BFM) [76]. The BFM contains a morphable model built using 200 faces and they provide an additional 10 unseen (out-of-sample) faces. In all our experiments we provide error measures with respect to ground truth data.

In our first experiment, we consider the optimal least squares shape estimate:

$$\mathbf{b}^* = \mathbf{P}^T(\mathbf{x} - \bar{\mathbf{x}}). \tag{6.16}$$

This represents the closest possible fit that the morphable model can make to the target vertex positions. We project out-of-sample faces onto the 199 parameter model and obtain the optimum

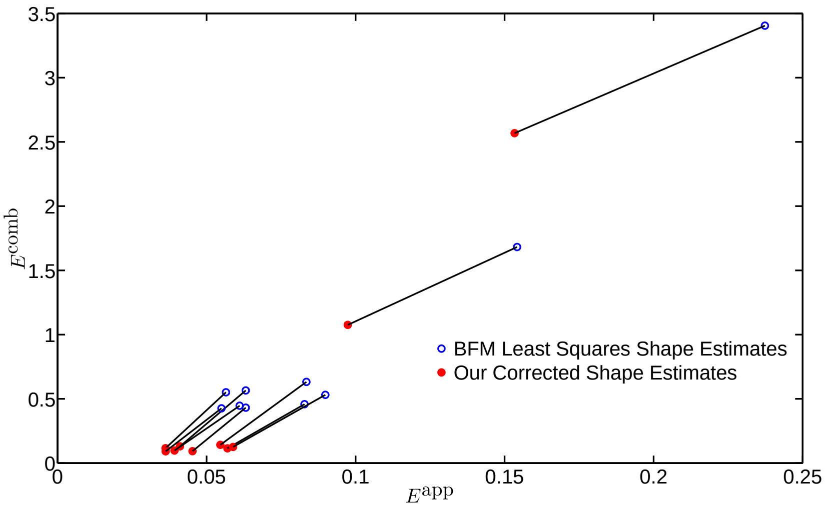

Figure 6.1: $E^{\text{comb}}$ (in mm$^2$) vs $E^{\text{app}}$ (in normalised squared pixel intensity).

least squares shape estimates using Equations 6.16 and 6.1. Using the morphable model shape estimates, we compute the on-cone vertex normals (Equation 6.8) and the refined geometry (Equation 6.11). The intensity (diffuse component) values which provide a constraint on the vertex normals are obtained from a rendering of the scan. Figure 6.1 shows a plot of $E^{\text{comb}}$ versus $E^{\text{app}}$ for each of the 10 test faces in the BFM. It is clear that refinement of the shape by minimising Equation 6.11 yields a decrease in $E^{\text{comb}}$ and $E^{\text{app}}$. The critical observation is that there is sufficient information in the surface orientation provided by shape-from-shading alone to obtain a mesh which is an improvement over the optimal morphable model shape according to two different errors measures. For the results reported in this chapter, $E^{\text{comb}}$ is computed using $\kappa = 0.1$. However, the shape refinement yields a decrease in $E^{\text{comb}}$ for any weighting of the two individual error terms.

Next we show experimental results for the problem of fitting a morphable model to a 2D image. For this we use the 270 BFM renderings. These consists of 27 renderings of the 10 unseen faces across varying lighting and pose. For each rendered image, we use the shape parameters obtained using a state-of-the-art morphable model fitting algorithm [76]. This analysis-by-synthesis technique optimises an objective comprising three error terms based on landmark points, contours and image intensity. For the purpose of our experiment we consider

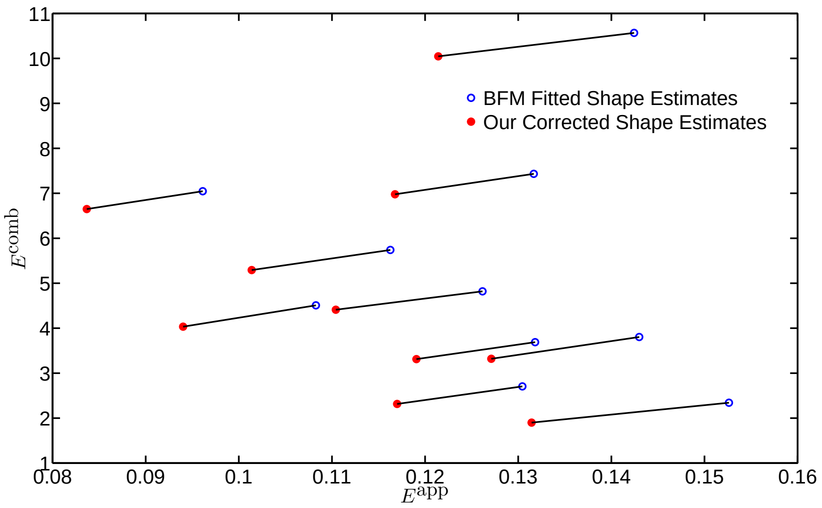

Figure 6.2: $E^{\text{comb}}$ (in mm$^2$) vs $E^{\text{app}}$ (in normalised squared pixel intensity).

the shape parameters provided for a 4 segment (eyes, nose, mouth and the rest) morphable model [76], with each segment having a face space of the 99 most significant modes. Hence the shape parameter vector $\mathbf{b} \in \mathbb{R}^{396}$. We take the fitted shape estimates and correct them using Equation 6.11. Figure 6.2 shows the plot of $E^{\text{comb}}$ vs $E^{\text{app}}$ for the 270 rendered images. We show the error measures averaged over the 27 renderings for each of the 10 subjects. We can see from Figure 6.2 that our refinement improves upon the shape estimate of the state-of-the-art morphable model fitting algorithm in all cases. Two of the refined results are shown in Figure 6.3. As can be seen in the figure, the corrected shapes for the examples clearly capture all the atypical features of the face (dimple on right cheek in row 1 and wrinkles on the cheek in row 2) in greater detail as compared to the BFM shape estimates.

In order to further validate this point, we use the BFM shape estimates and our corrected shape estimates in a recognition experiment (across varying pose and illumination). We divide the 270 rendering into a training set of 10 renderings (under frontal lighting and pose) and use the remaining 260 renderings as the test set. Thus we have 1 gallery image and 26 probe images per subject. We use the cosine of the angle between two shape parameter vectors (say $\mathbf{b}_1$ and

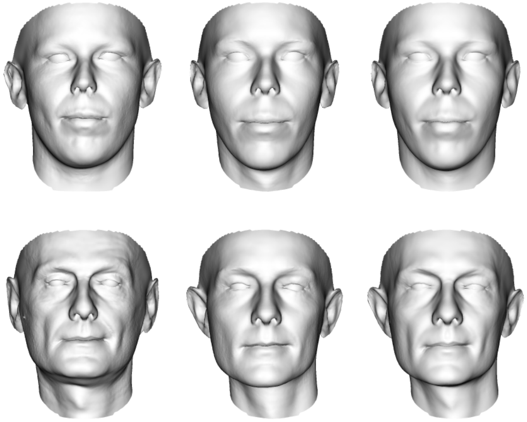

Figure 6.3: Ground truth out-of-sample faces (column 1), BFM fitted face shapes (column 2) and our corrected face shapes (column 3).

$\mathbf{b}_2$) as our similarity measure [15]:

$$\text{sim} = \frac{\mathbf{b}_1 \cdot \mathbf{b}_2}{\|\mathbf{b}_1\|\|\mathbf{b}_2\|}. \tag{6.17}$$

Using Equation 6.17, we obtain two sets of similarity matrices, one for the BFM shape parameters and one for the parameters obtained by projecting our refined shape back onto the morphable model. The recognition results are tabulated in Table 6.1. As we can see our correction step increases the recognition rate from 98.85 % to 99.62 %. Hence, our refinement step not only improves the recovered shape, but yields more informative parameter vectors when projected back into the model space.

These results show clearly that the additional constraint provided by shading cues helps exceed the optimum morphable model shape estimates and more importantly, improves upon the state-of-the-art analysis-by-synthesis results. Finally, the proposed method is very efficient. It requires $p$ vertex normal rotations (Equation 6.8) and a linear optimization (Equation 6.11). For a face with 53490 vertices the refinement process takes 0.16 sec to run on a 1.78 GHz AMD Athlon processor.

| Method | Recognition Rate (%) |
|---|---|
| BFM Parameters | 98.85 |
| Our Corrected Parameters | 99.62 |

Table 6.1: Face recognition results.

### 6.6.3 Discussion

The process described in this section is a one-shot post-processing step, which can be applied to improve upon the output of a morphable model fitting algorithm. The important implication of these results is that they demonstrate that shading information provides a powerful cue which can be used to exceed the accuracy possible with a morphable model.

Note that Equation 6.11 solves for vertex positions which are not constrained by the statistical model. Hence, iterating the refinement process turns out to be ineffective. If the recovered mesh is allowed to assume any shape, without the regularising effect of the morphable model, the process is prone to diverge and overfit to potentially noisy shape-from-shading normal estimates.

This drawback prevents the use of Equation 6.11 in model fitting applications. Nevertheless, shape-from-shading does provide useful cues which can potentially be used to guide the morphable model fitting process. For these reasons, in the next section we describe an iterative algorithm for fitting a morphable model to a single image. The aim is to strike a compromise between the stability of a purely model based approach and the improved accuracy afforded by allowing the shape to move outside the span of the statistical model according to shading constraints. We derive an objective function which penalises errors between model based surface normals and those implied by shape-from-shading, whilst simultaneously estimating albedo and bump maps which lie outside the span of the model and which strictly satisfy the image irradiance equation. This can be solved as a nonlinear optimisation and in doing so, robust 3D shape estimates can be obtained which capture the fine surface detail or atypical surface features of a face.

## 6.7 Fitting a Morphable Model using Shading Constraints

In this section we provide a framework to iteratively fit a morphable model based on minimizing the angular error between surface normals:

$$\mathbf{b}^* = \underset{D_M^2(\mathbf{b}) \leq D_{\max}^2}{\arg\min} \; \xi(\mathbf{b}), \tag{6.18}$$

where

$$\xi(\mathbf{b}) = \sum_p \left(\arccos\left(\mathbf{n}_p(\mathbf{b}) \cdot \Theta\mathbf{n}_p(\mathbf{b})\right)\right)^2. \tag{6.19}$$

In other words, Equation 6.18 seeks a morphable model shape estimate whose surface normals need minimal adjustment in order to strictly satisfy shading constraints. In contrast to Equation 6.11 (which simply adjusts the input vertex positions), this minimisation gives a new way to estimate model parameters in an iterative framework. $\mathbf{b}^*$ is the optimum shape parameter vector that minimises the objective function. $D_{\max}^2$ enforces a hard constraint on the maximum allowable parameter vector length. This controls the trade off between fitting quality and shape plausibility. Optimum performance occurs when $D_{\max}^2 \approx n$ (Chapter 3).

For each vertex in the mesh, we sample the image intensity by projecting the vertex to the image plane using an orthographic projection. We denote the intensity associated with the $p$th vertex as $I(\hat{r}_p)$. Perspective effects could in principle be modelled at the expense of estimating additional parameters. It is these intensity values which provide a constraint on the surface normal, which is strictly satisfied by applying the rotation $\Theta\mathbf{n}_p(\mathbf{b})$. The rotated vertex normals strictly satisfy Equation 6.7. These shape-from-shading normals possess two important qualities: 1. they will exactly recreate the input image 2. they are not constrained by the statistical model. The result of this is that they will capture fine surface detail. We therefore use these surface normals as a bump map in conjunction with the 3D mesh provided by the morphable model parameters.

The final ingredient in our method is to iteratively update the diffuse albedo map. Rearranging Equation 6.7, we can obtain a per-vertex estimate of the diffuse albedo:

$$\rho_d^p = \frac{I(\hat{r}_p) - \rho_s \cos^{\eta_s}(\theta_h^p(\mathbf{b}))}{\mathbf{n}_p(\mathbf{b}) \cdot \mathbf{s}}. \tag{6.20}$$

There is an additional constraint we impose here. The diffuse albedo cannot be greater than 1 (since a surface cannot reflect more light than was incident upon it) or less than the diffuse component of the sampled intensity. The diffuse albedo is estimated for each vertex at each iteration of the optimisation.

### 6.7.1 Implementation

Our implementation estimates morphable model shape parameters using an optimisation based on shape-from-shading constraints. The input to our algorithm is a single intensity image, the light source direction and the viewer direction. Following [38], we make some simplifying assumptions. The first is that we do not consider colour and the second is that the specular coefficient ($\rho_s$) and roughness parameter ($\eta_s$) are constant over the surface. We set these parameters as $\rho_s = 0.2$ and $\eta_s = 20$, which are reasonable for face images.

Allowing the albedo and surface normals to vary arbitrarily renders the problem underconstrained. Therefore, in practice we enforce an additional regularisation constraint which requires the albedo to be piecewise smooth. This has a long history in the albedo estimation literature, as the underpinning assumption of the Retinex algorithm [19]. To do the regularisation, we can apply any edge sensitive smoothing filter. For simplicity, we use a median filter on a small region around each vertex $p$:

$$\rho_d^p = \Psi_{q\times q}(\rho_d^p), \tag{6.21}$$

where $\Psi_{q\times q}$ represents a median filter applied on a region of size $q \times q$. Although the region size adds a parameter to our approach, it turns out to have an useful interpretation. When $q$ is large, the smoothing effect is large and high frequency variation is removed from the albedo

**Algorithm 1** Steps in evaluating the objective function

```
Constrain the shape parameter vector length (Equation 6.18),
for all visible $p$ do
    Compute the diffuse albedo using $I(\hat{r}_p)$ and $\mathbf{n}_p(\mathbf{b})$ (Equation 6.20),
end for
Enforce local consistency (regularisation constraint) on the diffuse albedo map (Equation 6.21),
err = 0,
for all visible $p$ do
    Compute $\Theta$ using $I(\hat{r}_p)$, $\mathbf{n}_p(\mathbf{b})$ and the regularised diffuse albedo (Section 6.5),
    Obtain $\Theta\mathbf{n}_p(\mathbf{b})$ the $p$th bump normal,
    $\text{err} = \text{err} + \arccos(\mathbf{n}_p(\mathbf{b}) \cdot \Theta\mathbf{n}_p(\mathbf{b}))$,
end for
return err
```

map and must be explained as changes in surface orientation, i.e. the 'shaded' hypothesis. On the other hand, when $q$ is small the albedo is free to retain high frequency variation, so sharp changes in intensity are explained as albedo variation, i.e. the 'painted' hypothesis. Hence, the size parameter provides a user tuneable interface to smoothly vary between the two hypotheses. We discuss good choices of value for this parameter in our experimental results.

We minimise Equation 6.18 (a non-linear objective function) using the Levenberg-Marquardt algorithm [63] to find optimal estimates of the shape parameter vector $\mathbf{b}^*$. Our convergence criterion is based on the total angular error between the model based and shape-from-shading normals:

$$\xi(\mathbf{b}) < \epsilon. \tag{6.22}$$

Evaluating the objective function involves a number of steps which are described in Algorithm 1. This process is executed at each iteration of the optimisation. We initialise our optimisation by fitting the model to a sparse set of manually selected feature points (based on Chapter 3). The initialisation also provides an estimate of the 3D pose. A similar initialisation strategy was used by Blanz and Vetter [15] for their stochastic optimisation procedure. Note that simply using this initial shape estimate to compute the albedo and bump maps (i.e. one pass of Algorithm 1, without optimising Equation 6.18) will not yield optimal results. We provide experimental validation of this in Section 6.8.3.

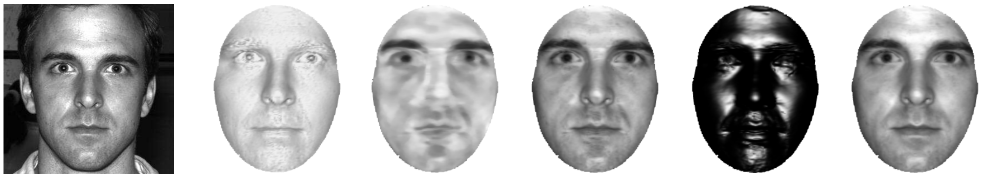

Figure 6.4: Given an input 2D image (leftmost). The figure shows (from left to right) the estimated a) bump map, b) diffuse albedo map, c) Lambertian reflectance map, d) specular reflectance map and e) Blinn-Phong reflectance map.

Upon convergence we output the final per-vertex bump and albedo maps. When combined with the estimated mesh, these maps ensure that the input image is recreated exactly.

### 6.7.2 Face Recognition

We follow the approach of Basri and Jacobs [8] based on spherical harmonics in which a low-dimensional subspace is derived analytically from a model. They show that under any lighting conditions, at least 98% of the variability in the reflectance function is captured by the first 9 harmonic images. Their analysis therefore suggests that images of a convex Lambertian surface will lie close to a 9D subspace. This subspace can be derived exactly from the estimated albedo and bump maps without being dependent on the quantity or variability of a sample of training images.

Let $\boldsymbol{\rho_d}$ denote a vector of length $l$ containing the albedo values across a face's surface, such that $\rho_d^l$ is the albedo at the $l$th vertex. Here $l \leq p$ is the number of visible vertices. Similarly, the $x$, $y$ and $z$ components of the surface normals are stacked to form a further three vectors of length $l$: $\mathbf{n_x}$, $\mathbf{n_y}$ and $\mathbf{n_z}$, such that $n_x^l$ is the $x$ component of the $l$th vertex normal. We define: $\mathbf{n_{x^2}} = \mathbf{n_x} .* \mathbf{n_x}$ (where the operator $.*$ denotes the component-wise product of two vectors of the same length). Similarly for $\mathbf{n_{y^2}}$, $\mathbf{n_{z^2}}$, $\mathbf{n_{xz}}$, $\mathbf{n_{yz}}$ and $\mathbf{n_{xy}}$. The first nine harmonic images for

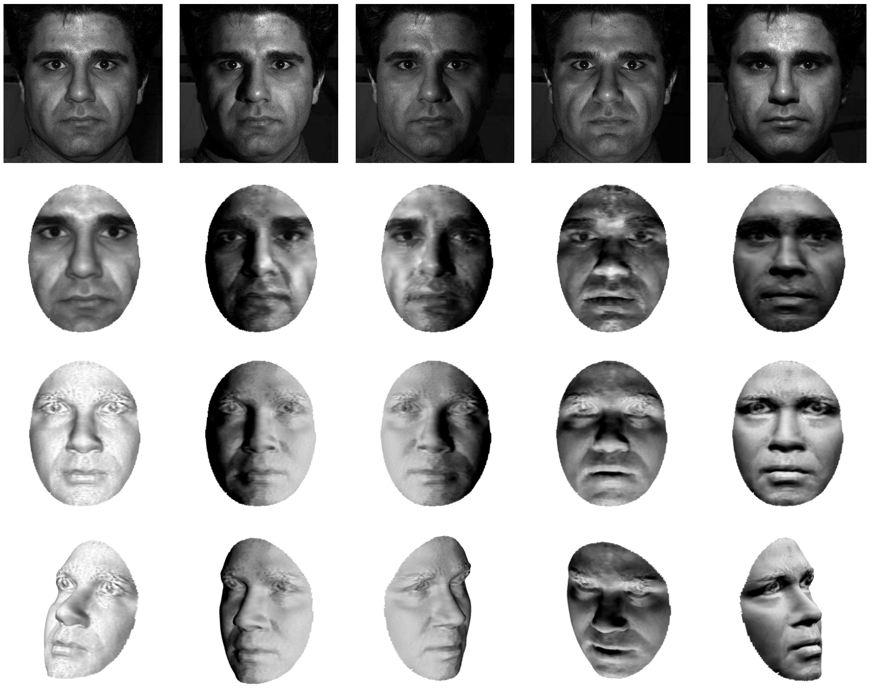

Figure 6.5: The first image in row 1 is the image under frontal illumination ($\mathbf{s} = (0,0,1)^T$). This image is used as the input image for synthesis. The remaining 4 images in row 1 (from left to right) are the actual images corresponding to the light source direction $\mathbf{s} = (-1,0,1)^T$, $\mathbf{s} = (1,0,1)^T$, $\mathbf{s} = (0,-1,1)^T$ and $\mathbf{s} = (0,1,1)^T$ respectively. Row 2 shows the synthesised image under the 5 different light source directions. Row 3 shows similar results for the the estimated bump map (i.e. $\cos(\theta_i)$). Row 4 shows the results of row 3 in a novel pose.

a surface with known normals and albedo are given by:

$$
\begin{aligned}
\mathbf{e}_{00} &= \tfrac{1}{\sqrt{4\pi}}\boldsymbol{\rho_d}, & \mathbf{e}_{10} &= \sqrt{\tfrac{3}{4\pi}}\boldsymbol{\rho_d}.*\mathbf{n_z}, \\
\mathbf{e}^o_{11} &= \sqrt{\tfrac{3}{4\pi}}\boldsymbol{\rho_d}.*\mathbf{n_y}, & \mathbf{e}^e_{11} &= \sqrt{\tfrac{3}{4\pi}}\boldsymbol{\rho_d}.*\mathbf{n_x}, \\
\mathbf{e}_{20} &= \tfrac{1}{2}\sqrt{\tfrac{3}{4\pi}}\boldsymbol{\rho_d}.*(2\mathbf{n_{z^2}} - \mathbf{n_{x^2}} - \mathbf{n_{y^2}}), \\
\mathbf{e}^o_{21} &= 3\sqrt{\tfrac{5}{12\pi}}\boldsymbol{\rho_d}.*\mathbf{n_{yz}}, & \mathbf{e}^e_{21} &= 3\sqrt{\tfrac{5}{12\pi}}\boldsymbol{\rho_d}.*\mathbf{n_{xz}}, \\
\mathbf{e}^o_{22} &= 3\sqrt{\tfrac{5}{12\pi}}\boldsymbol{\rho_d}.*\mathbf{n_{xy}}, & \mathbf{e}^e_{22} &= \tfrac{3}{2}\sqrt{\tfrac{5}{12\pi}}\boldsymbol{\rho_d}.*(\mathbf{n_{x^2}} - \mathbf{n_{y^2}}).
\end{aligned}
\tag{6.23}
$$

We form a $l \times 9$ matrix $\mathbf{B}$ containing the basis images as columns . However, this basis is not orthonormal. Using a QR decomposition, we find the $l \times 9$ orthonormal basis $\mathbf{Q}$ and $9 \times 9$ matrix $\mathbf{R}$, such that $\mathbf{QR} = \mathbf{B}$. Given a vector of sampled image intensities $\mathbf{I}$, we may now compute the distance to the subspace using: $\|\mathbf{QQ}^T\mathbf{I} - \mathbf{I}\|$ and perform recognition.

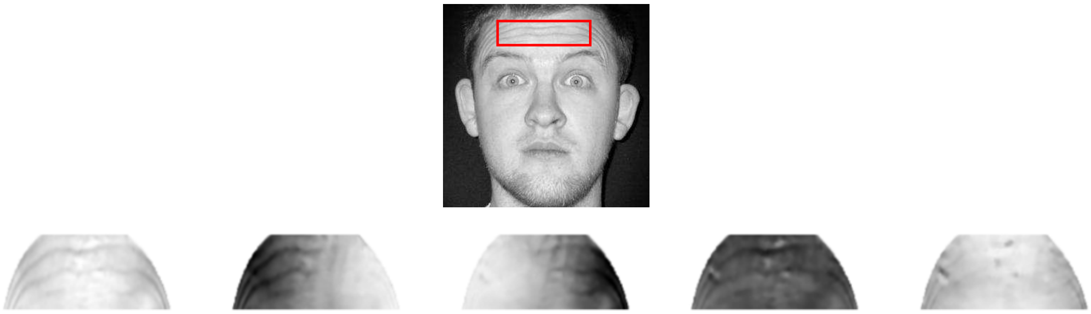

Figure 6.6: Given an input image with prominent wrinkles (red bound box). The second row shows the estimated bump map (i.e. $\cos(\theta_i)$) under 5 light source directions ($\mathbf{s} = (0,0,1)^T$, $\mathbf{s} = (-1,0,1)^T$, $\mathbf{s} = (1,0,1)^T$, $\mathbf{s} = (0,-1,1)^T$ and $\mathbf{s} = (0,1,1)^T$).

## 6.8 Experimental Results

We now present results for the fitting algorithm proposed in Section 6.7. For these experiments the 3D morphable model was built using 100 face scans obtained from a Cyberware 3030PS laser scanner. The data was set into correspondence using the principle of thin-plate splines and the model was constructed according to Kendall's notion of a shape space (Chapter 3). We retain 99 modes for all our experiments. The morphable model fitting process takes 2 to 2.5 mins to run on a 1.78 GHz AMD Athlon processor. We provide results on the data obtained from the Yale Face Database B [39].

### 6.8.1 Face Synthesis

Figure 6.4 shows the shape and reflectance parameters estimated from a single input image. Note that the albedo map contains no residual shading information, while the bump map clearly contains fine-scale surface detail present in the original image. Figure 6.5 shows the synthesis results under different light source directions.

Figure 6.6 shows the ability of our algorithm to recover shape not present in the training set. To demonstrate this we captured a 2D image of a subject with clearly visible wrinkles. In the second row we show a zoomed portion of the mesh in which we have reilluminated the bump map with a varying light source direction. The appearance of the wrinkles under the different light source directions is realistic. Note in particular when $\mathbf{s} = (0,1,1)^T$ (i.e the face is lighted

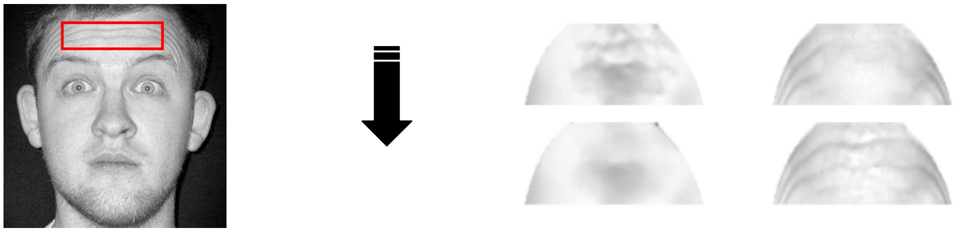

Figure 6.7: Given an input image with prominent wrinkles (red bound box). The rows (on the right) show the estimated albedo (left) and bump map (right) for different amounts of regularisation constraint applied. The black block arrow indicates an increasing $q$ (from $q = 3$ to $q = 5$).

from the top), the wrinkles disappear altogether. This would not be the case if the wrinkles were explained as variation in albedo.

Figure 6.7 shows how our algorithm can be used to explain shading variations as either changes in surface orientation (the 'shaded' hypothesis) or albedo (the 'painted' hypothesis). Our experimentation shows, using $q = 5$ provides the right trade off between allowing the shading variations to be explained as changes in surface orientation and not allowing too severe a regularisation constraint. This choice of $q$ works well in critical applications like recognition.

In Figure 6.8 we show synthesis results for 5 subjects in the Yale Face Database B. For each subject we show the estimated Blinn-Phong reflectance (rows 2 and 3) and bump maps (rows 4 and 5). In all cases the shape recovery process is stable (even with large variations in albedo caused by facial hair). Note in particular that the bump maps successfully capture discriminating facial shape details, i.e. the identity of each subject is clearly visible in the shape-only images. Since the specular term is dependent on the viewer direction, we can observe the specularity motion when the viewpoint changes (row 3). Finally, in Figure 6.9 we show synthesis results for input images in non-frontal poses. We show the estimated Blinn-Phong reflectance and bump maps for both images.

### 6.8.2 Sensitivity to Non-Ideal Illumination Conditions

Our reflectance model assumes a single directed light. In practice, this assumption may be too strong for many real world applications. Nevertheless, because of the regularising effect of the

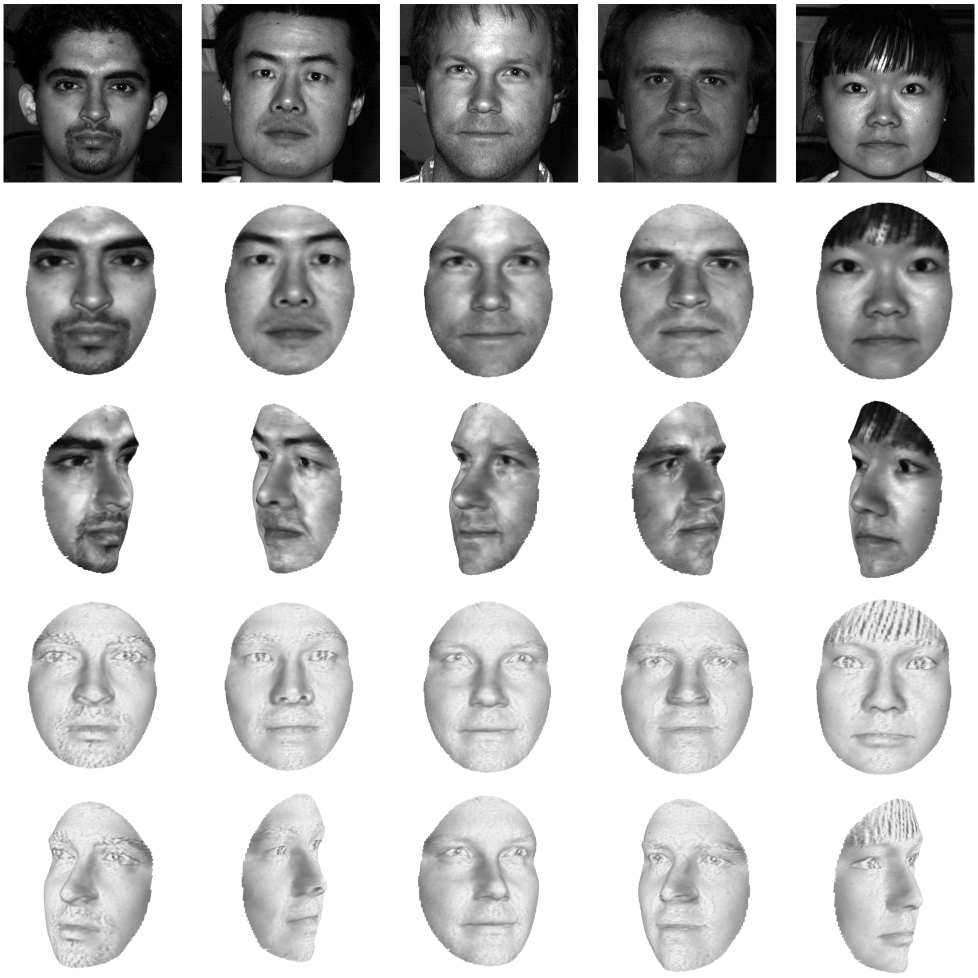

Figure 6.8: Face synthesis results for 5 subjects in the Yale Database B.

morphable model on the shape estimate, our algorithm is robust to some degree of deviation from the assumed conditions. We provide evaluation for the case when there is more than one source present in the scene. To do this, we generate images which simulate the presence of 3 and 5 directed light sources by taking weighted combinations of the single source images. The light source directions are given in Table 6.2.

We consider the shape estimates from images taken under light source $\mathbf{s}_1$ as our baseline. Mean angular error from the baseline shape to that recovered from images containing multiple light sources are shown in Table 6.3. The results show that our estimated surface normals deviate by only $\sim 6^\circ$ when as many as 4 distracting sources are included. These results demonstrate that our method is robust to non-ideal illumination conditions, i.e. those likely to be encountered in

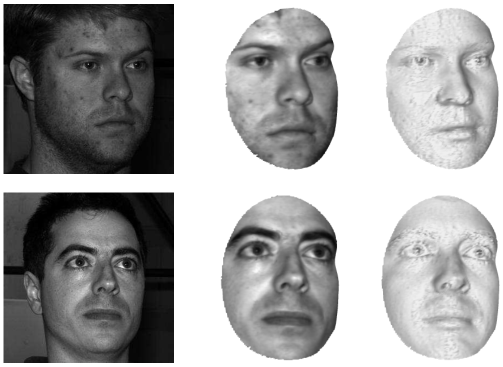

Figure 6.9: Face synthesis results for non-frontal input images. The pose variation is about 24 ° from the camera optical axis [39].

| Light<br>Source | Azimuthal Angle<br>in ° | Elevation Angle<br>in ° |
|---|---|---|
| $\mathbf{s}_1$ | 0 | 0 |
| $\mathbf{s}_2$ | -70 | 0 |
| $\mathbf{s}_3$ | 85 | -20 |
| $\mathbf{s}_4$ | 20 | -40 |
| $\mathbf{s}_5$ | 0 | 45 |

Table 6.2: List of five directed light sources considered for our evaluation. Baseline images are taken under light source $\mathbf{s}_1$. 3 light source images result by combining images taken under light sources $\mathbf{s}_1$, $\mathbf{s}_2$ and $\mathbf{s}_3$ (with a weighting 0.8,0.1,0.1). 5 light source images result by combining images taken under light sources $\mathbf{s}_1$,$\mathbf{s}_2$,$\mathbf{s}_3$,$\mathbf{s}_4$ and $\mathbf{s}_5$ (with a weighting 0.6,0.1,0.1,0.1,0.1). See Figure 6.10 for an example of the generated images. Refer [39] for details about the original images under each tabulated light source.

images taken in an uncontrolled environment. Figure 6.10 provides the qualitative analysis for our experiment.

### 6.8.3 Illumination Invariant Face Recognition

We perform illumination invariant face recognition on data obtained from the Yale Face Database B [39], which contains images of 10 individuals (disjoint from the morphable model training data) under 45 different illumination and 9 pose conditions. In our experiment we consider images having frontal pose (pose 1). We group the lighting variation into 4 subsets of differing extremity (refer [39] for details).

We use a single training image with frontal lighting (azimuth and elevation angles equal to zero degrees). Our test set consists of the remaining 440 varying illumination images. We apply

| Subject Number | 3 light source images | 5 light source images |
|---|---|---|
| 1 | 5.8 | 5.93 |
| 2 | 5.36 | 5.71 |
| 3 | 7.08 | 7.36 |
| 4 | 5.30 | 5.71 |
| 5 | 6.80 | 7.13 |
| 6 | 5.43 | 5.65 |
| 7 | 6.52 | 6.55 |
| 8 | 5.56 | 5.88 |
| 9 | 6.28 | 6.30 |
| 10 | 5.02 | 5.31 |
| Average | 5.92 | 6.15 |

Table 6.3: Mean angular error (in °) between the estimated bump maps for each subject in the Yale Database B. All reported errors are with respect to the baseline estimates (for images taken under a single directed light source $\mathbf{s}_1$).

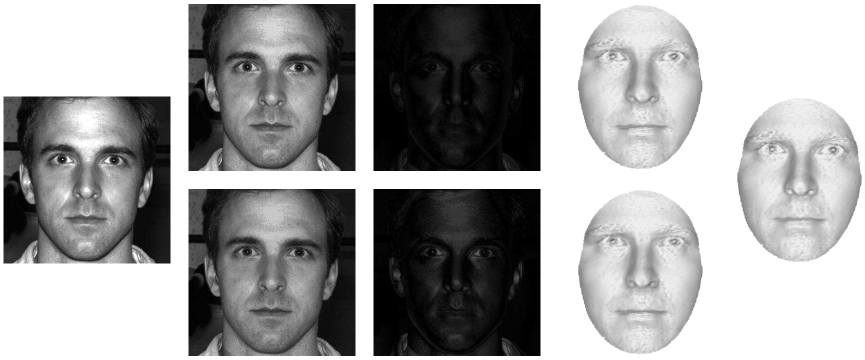

Figure 6.10: Estimated bump maps (fourth column) for images taken under 3 and 5 directed light sources. The first column show the baseline image (taken under $\mathbf{s}_1$). The second column shows the images simulating 3 and 5 directed light sources (refer Table 6.2). The third column shows the difference images with respect to the baseline image. The last column shows the estimated bump map for the baseline image.

our algorithm to each training image and use the estimated albedo and bump maps to perform recognition (as described in Section 6.7.2).

Figure 6.11 shows the estimated albedo map, bump map and spherical harmonic basis for a subject in the database. Table 6.4 shows the recognition results for our method and compares them with other methods. Our method is most similar to that proposed by Zhang and Samaras [103], who use the fit of a morphable model to each gallery image to obtain a spherical harmonic basis for that subject. In contrast to the method proposed in [103], our albedo and bump maps not are constrained by a statistical model and are therefore free to capture atypical,

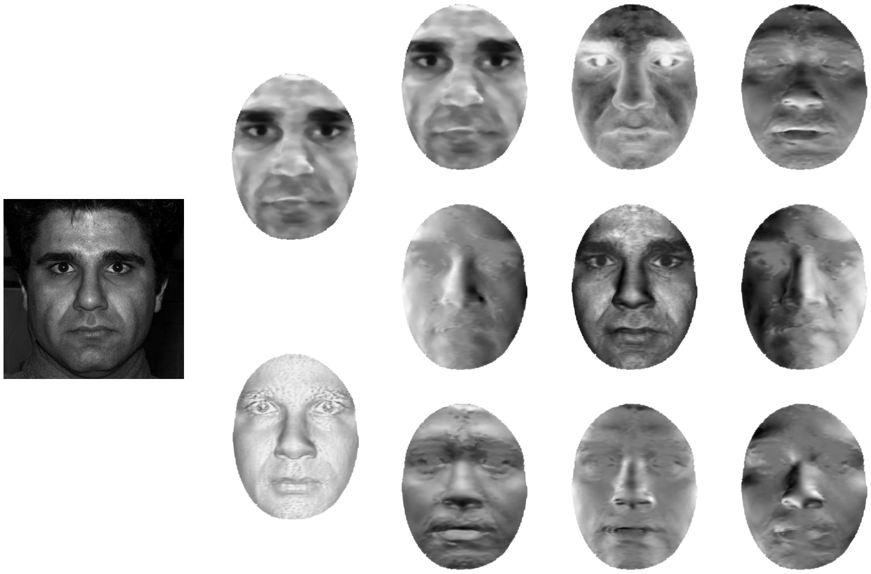

Figure 6.11: Given an input 2D image under frontal illumination (first column). The figure shows the estimated albedo and bump maps (second column). The last 3 columns show the spherical harmonic subspace derived from the estimated albedo and bump maps.

discriminating facial features. The advantage of non-model based albedo and bump maps is evident from our recognition performance as compared to that of [103]. The MRF based method [98] reports the best recognition results using a spherical harmonic model and single training image. This method involves a more complicated optimization process, with a greater number of parameters to be optimized as compared to the method of [103] or our method. The 30 basis BIM also achieves a better recognition rate compared to our method (using 9 spherical harmonic basis). However, for a 9 basis BIM the subset 4 error rate increases to 7.1% [57].

Finally, as stated in Section 6.7.1, using the initial feature point based estimates will not result in optimal performance. For the face recognition experiment described above, our optimal estimates result in an error rate of 1.4% on subset 4, whereas using the initial feature points based estimates result in an error rate of 10% on the same subset. This point clearly illustrates the importance of iteratively minimising Equation 6.18.

| Comparison of Recognition Methods | | | | |
|---|---|---|---|---|
| Method | Number of Training Images | Error Rate(%) vs. Illum | | |
| | | Subset 1 & 2 | Subset 3 | Subset 4 |
| Correlation | 6-7 | 0.0 | 23.3 | 73.6 |
| Eigenfaces | 6-7 | 0.0 | 25.8 | 75.7 |
| Linear subspace | 6-7 | 0.0 | 0.0 | 15 |
| Illum. Cones-attached | 6-7 | 0.0 | 0.0 | 8.6 |
| Illum. Cones-cast | 6-7 | 0.0 | 0.0 | 0.0 |
| 9 Points of Light (9PL) | 9 | 0.0 | 0.0 | 2.8 |
| Zhang & Samaras | 1 | 0.0 | 0.0 | 3.1 |
| BIM (30 basis) | 1 | 0.0 | 0.0 | 0.7 |
| MRF Based Method | 1 | 0.0 | 0.0 | 0.1 |
| Our Method | 1 | 0.0 | 0.0 | 1.4 |

Table 6.4: Recognition results on the Yale Face Database B. Except for our method the data was summarized from [98].

### 6.8.4 Pose and Illumination Invariant Face Recognition

It is straightforward to extend our method to perform recognition on faces in a non-frontal pose. The fitting algorithm can be applied to a non-frontal training set, a spherical harmonic basis (for each training image) constructed and subsequently used to perform illumination insensitive recognition.

However, performing recognition across poses using a training image in a single pose does not result in robust performance. There are multiple reasons for this. The first is that the number of visible vertices varies with pose and hence the length of the spherical harmonic basis also varies. This means that a basis constructed in one pose (say frontal) and transformed to an orthonormal basis using a QR decomposition will no longer be orthonormal when vertices visible in another pose are sub-selected. The second problem is that vertices which are not visible in the training pose will have no albedo and bump map estimates. They will therefore be missing in other poses where those vertices may be visible.

Due to these obstacles, robust pose and illumination invariant recognition (using the proposed method) can only be performed by using one training image per pose.

## 6.9 Conclusions

In this chapter we have shown how constraints on surface normal direction suggested by shading cues provide a route to fitting a morphable model to single images and refining model based shape estimates.

We began by demonstrating that surface normals estimated using shape-from-shading could be used to refine morphable model face shape estimates, even exceeding the accuracy of a least squares model fit. We then proposed an algorithm for iteratively fitting a 3D morphable model using constraints suggested by shape-from-shading. Our approach provides a robust means of estimating facial albedo and bump maps which are not constrained by a statistical model. They are therefore free to capture atypical, discriminating facial features. We also show how this data can used to perform illumination insensitive face recognition from a single gallery image and obtain results which are competitive with the state-of-the-art.

# Chapter 7

# Conclusions

In recent years, 3D morphable models have attracted considerable research attention within the computer vision community (and related areas). Since the models were first introduced about a decade ago, substantial headway has been made and they now provide a viable research tool for face processing applications ranging from forensics to analysis of art. They have also been shown to offer a potential solution to the problem of pose and illumination invariance in face recognition [80]. However, the problem of fitting a morphable model to image data in a general setting is still an unsolved problem and they are therefore some way away from being viable as a commercial tool. Next, we summarise the contributions made in this thesis, highlight some weaknesses of the work presented in this thesis and end by discussing directions for future research.

## 7.1 Summary of Contributions

In this thesis, we have proposed a number of advances in the training and application of morphable models which go some way to addressing the outstanding issues identified in Chapter 1. The contributions made in this work are summarised in the following list.

- We have shown how to simplify both training data (Chapter 2) and trained models (Chapter 5) in a way which yields class-optimal low resolution morphable models. These simplified models can be plugged into any existing fitting algorithm to reduce computational expense.

- We have shown (in Chapter 3) how the tools of thin-plate spline warping and Procrustes analysis can be used to construct a morphable model that is both more efficient and generalises to novel face surfaces more accurately than previous models.

- We have shown (in Chapter 4) how a hard constraint on facial distinctiveness ensures plausible fitting results without having to employ global optimisation strategies to avoid local minima close to the mean. The constraint also helps alleviate problems of model dominance by ensuring all results are equally distinctive without having to tune a regularisation parameter.
- We have shown (in Chapter 6) how incorporation of shading constraints into the morphable model fitting process allows model dominance to be overcome by moving outside the span of the model. This allows the recovery of fine surface detail.

## 7.2 Critical Analysis

There are some criticisms which may be levelled at the work presented in this thesis. We now discuss these weaknesses and then provide potential avenues to address them in the next section.

- **Enforced Hard Constraint on the Parameter Vector Length**: In Chapter 4 we showed how we can constrain faces to lie on a manifold of equal distinctiveness, with only identity allowed to vary. The enforced distinctiveness was chosen as the expected value of the distribution which follows from the statistics of the model parameters. This hard constraint could potentially be deemed as limiting or restrictive.
- **Iterative Optimisation**: The model fitting algorithms described in this thesis involve a nonlinear optimisation over the model parameters. The drawback of this is that iterative procedures are ill-suited to applications which must operate in real time.
- **Lack of Local Statistical Models**: In Chapter 6 we showed high frequency details can be captured by incorporating shading constraints. However, we had no statistical model to efficiently characterise the high frequency variations in an individuals face shape

and albedo. Incorporating such local models would help regularise the shape/albedo estimation process and may significantly improve the results presented in Chapter 6.

- **Lack of Photorealism**: In Chapter 6 we modelled human skin using a Blinn-Phong reflectance model. It can be argued that incorporating more realistic reflectance and illumination models would greatly enhance quality of rendered estimates. Additionally, biophysical constraints could have an important role to play in obtaining meaning parametric descriptions of skin histology.

## 7.3 Directions for Future Research

We conclude the thesis by providing directions for future research. Whilst most of the stated directions address the points raised in Section 7.2, we also ask interesting questions about the training data and statistical modelling.

- **Relaxing the Manifold Based Constraint**: In the future it would be interesting to relax the enforced hard constraint on the parameter vector length. The decomposition into identity and distinctiveness may still prove useful as different priors and constraints could be enforced on direction and magnitude in parameter space.

- **Convex Objective Functions**: One of the major challenges of using a 3D morphable models to fit to data (such as images, photometric normals or noisy range scans), is that it requires solving a highly complex nonconvex optimisation problem. This is computationally expensive and also there is no guarantee of obtaining the globally optimal solution. There has been significant recent progress in recognising the way in which convex (but potentially nonlinear) objective functions can be constructed through disciplined convex programming [42]. Such systems can be efficiently solved. It would be interesting to approach the problem of morphable model fitting from the perspective of designing a convex objective function. One approach we have begun to experiment with involves deriving a linear system of equations which minimises both surface orientation errors and positional errors in the image plane. This is particularly well suited to fitting the model to 2.5D data, such as might be obtained by photometric stereo. We believe the whole problem could be formulated using a bilinear system with a linear objective to estimate the camera parameters (along the lines of the Gold Standard Algorithm [44]), followed by a linear objective to estimate the shape parameters (extending the method of [74]).

- **Local/Global Statistical Models**: Global statistical models cannot efficiently characterise the high frequency variations in individuals face shape and texture. In a sense, projection onto a global model acts like a low-pass filter. A critical question is therefore how to best represent the remaining high frequency information. One approach would be to augment the global model with local models of high frequency variation. For example, Mohammed et al. [72] demonstrated a data-driven approach along these lines which operates in 2D. The challenge is whether such high frequency (and potentially stochastic) variation can be efficiently modelled. To this end Golovinskiy et al. [40] introduced a statistical technique for the analysis and synthesis of small 3D facial features, such as wrinkles and pores. Another approach would be to enforce biophysical constraints. For example, only certain skin types can exhibit freckles. Freckles themselves follow spatial distributions which can be learnt from data and their appearance is determined by well understood skin histology. Hence, skin type may be stored as a global parameter (which is in turn constrained by ethnicity), while freckle distribution can be explained using spatially varying local models.

- **Enhancing Realism of Appearance Modelling**: Photorealistic rendering of faces is now possible when sufficiently accurate shape and reflectance information is available [64]. Important advances have included skin reflectance models which provide parametric descriptions of skin histology that can generalise to any skin type [51] and realistic models of environment illumination [84]. By incorporating more information about the reflectance properties of faces into the model and using more realistic reflectance and illumination models in the fitting process, much higher quality results are likely to be possible [3].

The challenge here is that this additional information increases the dimensionality of the training data and hence the difficulty of learning a low dimensional model.

- **Training Data and Statistical Modelling**: In 2D face analysis, the vast availability of data has led to a whole subfield dedicated to developing manifold learning techniques to efficiently and accurately describe face space. Because of the much higher cost and technical challenge of capturing 3D data and obtaining dense correspondence for large datasets, a similar literature for 3D data does not exist. For this reason, there are many open questions related to statistical modelling and the empirical nature of 3D face space. These include:
  - How much training data is required to build a model which can accurately generalise to any face?
  - What is the true dimensionality of face space?
  - Is a unimodal multivariate Gaussian distribution an accurate or efficient model of the true underlying face space? Or would a better model result from a nonlinear or multilinear analysis?
  - Can face space be learnt in an unsupervised manner? Or should information about gender, ethnicity, age etc be used to label training samples and guide the manifold learning process?
  - Is the sparsity of training samples with respect to the dimensionality of the data a limiting problem?

# References

[1] USF HumanID 3D Face Database, Courtesy of Sudeep. Sarkar, University of South Florida, Tampa, FL.

[2] R. Adler, J.-P. Dedieu, J. Margulies, M. Martens, and M. Shub. Newton's method on Riemannian manifolds and a geometric model for the human spine. *IMA J. of Numerical Analysis*, 22:359–390, 2002.

[3] O. Alexander, M. Rogers, W. Lambeth, M. Chiang, and P. Debevec. Creating a photoreal digital actor: The Digital Emily project. In *Proc. CVMP*, pages 176–187, 2009.

[4] B. Allen, B. Curless, and Z. Popović. The space of human body shapes: reconstruction and parameterization from range scans. *ACM Trans. Graphic. (Proceedings of SIGGRAPH)*, 22(2):587–594, 2003.

[5] B. Amberg, A. Blake, A. Fitzgibbon, S. Romdhani, and T. Vetter. Reconstructing high quality face-surfaces using model based stereo. In *Proc. ICCV*, 2007.

[6] B. Amberg, S. Romdhani, and T. Vetter. Optimal step nonrigid ICP algorithms for surface registration. In *Proc. CVPR*, pages 1–8, 2007.

[7] J. J. Atick, P. A. Griffin, and A. N. Redlich. Statistical approach to SFS: Reconstruction of 3D face surfaces from single 2D images. *Neural Comp.*, 8(6):1321–1340, 1996.

[8] R. Basri and D. W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(2):218–233, 2003.

[9] M. Ben-Chen, C. Gotsman, and G. Bunin. Conformal flattening by curvature prescription and metric scaling. *Comput. Graph. Forum*, 27(2):449–458, 2008.

[10] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet. Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering. In *Proc. NIPS*, pages 177–184, 2004.

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

[12] V. Blanz, A. Mehl, T. Vetter, and H-P. Seidel. A statistical method for robust 3d surface reconstruction from sparse data. In *Proc. 3DPVT*, pages 293–300, 2004.

[13] V. Blanz, K. Scherbaum, and H.-P. Seidel. Fitting a morphable model to 3D scans of faces. In *Proc. ICCV*, 2007.

[14] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Proc. SIGGRAPH*, pages 187–194, 1999.

[15] V. Blanz and T. Vetter. Face recognition based on fitting a 3D morphable model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(9):1063–1074, 2003.

[16] J. F. Blinn. Models of light reflection for computer synthesized pictures. *SIGGRAPH Comput. Graph.*, 11:192–198, 1977.

[17] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(6):567–585, 1989.

[18] F. L. Bookstein. Landmark methods for forms without landmarks: Morphometrics of group differences in outline shape. *Med. Image Anal.*, 1(3):225–243, 1997.

[19] D. H. Brainard and B. A. Wandell. Analysis of the retinex theory of color vision. *J. Opt. Soc. Am. A*, 3(10):1651–1661, 1986.

[20] C. G. Broyden. The convergence of a class of double-rank minimization algorithms. *Journal of the Institute of Mathematics and Its Applications*, 6(1):76–90, 1970.

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

[22] P. Cignoni, C. Montani, C. Rocchini, and R. Scopigno. A general method for preserving attribute values on simplified meshes. In *IEEE Visualization 1998*, pages 59–66, 1998.

[23] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: Measuring error on simplified surfaces. *Comput. Graph. Forum*, 17(2):167–174, 1998.

[24] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *Proc. ECCV*, pages 484–498, 1998.

[25] T. F. Cootes, C. J. Taylor, D. Cooper, and J. Graham. Active shape models – their training and application. *Comput. Vis. Image Underst.*, 61:38–59, 1995.

[26] T. F. Cox and M. A. Cox. *Multidimensional Scaling.* CRC Press, 1994.

[27] C. DeCoro and S. Rusinkiewicz. Pose-independent simplification of articulated meshes. In *Proc. Symposium on Interactive 3D Graphics*, 2005.

[28] R. Dovgard and R. Basri. Statistical symmetric shape from shading for 3D structure recovery of faces. In *Proc. ECCV*, volume 2, pages 99–113, 2004.

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

[30] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* John Wiley and Sons, 1977.

[31] A. Ecker and A. D. Jepson. Polynomial shape from shading. In *Proc. CVPR*, pages 145–152, 2010.

[32] B. Fink, J. T. Manning, N. Neave, and K. Grammer. Second to fourth digit ratio and facial asymmetry. *Evol. Hum. Behav.*, 25(2):125–132, 2004.

[33] M. S. Floater. Mean value coordinates. *Comput. Aided Geom. D.*, 20(1):19–27, 2003.

[34] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Huges. *Computer Graphics: Principles and Practice.* Addison–Wesley, 1996.

[35] Y. Furukawa and J. Ponce. Dense 3d motion capture from synchronized video streams. In *Proc. CVPR*, pages 1–8, 2008.

[36] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. In *Proc. SIGGRAPH*, pages 209–216, 1997.

[37] M. Garland and P. S. Heckbert. Simplifying surfaces with color and texture using quadric error metrics. In *Visualization*, pages 263–269, 1998.

[38] A. Georghiades. Recovering 3–D shape and reflectance from a small number of photographs. In *Eurographics Symposium on Rendering*, pages 230–240, 2003.

[39] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(6):643–660, 2001.

[40] A. Golovinskiy, W. Matusik, H. Pfister, S. Rusinkiewicz, and T. Funkhouser. A statistical model for synthesis of detailed facial geometry. In *Proc. SIGGRAPH*, pages 1025–1034, 2006.

[41] A. Golovinskiy, J. Podolak, and T. Funkhouser. Symmetry-aware mesh processing. In *Proc. Mathematics of Surfaces*, pages 170–188, 2009.

[42] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In *Recent Advances in Learning and Control*, pages 95–110. Springer-Verlag Limited, 2008.

[43] `http://alice.loria.fr/index.php/software/3-platform/22-graphite.html`, June 2010.

[44] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2000.

[45] N. Hasler, C. Stoll, M. Sunkel, B. Rosenhahn, and H.-P. Seidel. A statistical model of human pose and body shape. *Comput. Graph. Forum*, 2(28):337–346, 2009.

[46] U. Helmke, K. Hüper, and J. Moore. Quadratically convergent algorithms for optimal dextrous hand grasping. *IEEE Trans. Robotic. Autom.*, 18(2):138–146, 2002.

[47] H. Hoppe. Efficient implementation of progressive meshes. *Comput. Graph.*, 22(1):27–36, 1998.

[48] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America. A*, 4(4):629–642, 1987.

[49] C. Hu, J. Xiao, I. Matthews, S. Baker, J. Cohn, and T. Kanade. Fitting a single active appearance model simultaneously to multiple images. In *Proc. BMVC*, 2004.

[50] K. Huper and J. Trumpf. Newton-like methods for numerical optimization on manifolds. In *Proc. IEEE Asilomar Conference on Signals Systems and Computers*, pages 136–139, 2004.

[51] J. Jimenez, T. Scully, N. Barbosa, C. Donner, X. Alvarez, T. Vieira, P. Matts, V. Orvalho, D. Gutierrez, and T. Weyrich. A practical appearance model for dynamic facial color. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 29(6):141:1–141:10, 2010.

[52] J. Kiefer. Sequential minimax search for a maximum. *Proceedings of the American Mathematical Society*, 4(3):502–506, 1953.

[53] R. Kimmel and J. A. Sethian. Computing geodesic paths on manifolds. *Proc. Natl. Acad. Sci.*, 95(15):8431–8435, 1998.

[54] R. Knothe, S. Romdhani, and T. Vetter. Combining PCA and LFA for surface reconstruction from a sparse set of control points. In *Proc. Int. Conf. on Automatic Face and Gesture Recognition*, pages 637–644, 2006.

[55] V. Kraevoy and A. Sheffer. Cross-parameterization and compatible remeshing of 3D models. *ACM Trans. Graphic. (Proceedings of SIGGRAPH)*, 23(3):861–869, 2004.

[56] E. Landreneau and S. Schaefer. Simplification of articulated meshes. *Comput. Graph. Forum (Proceedings of EUROGRAPHICS)*, 28(2):347–353, 2009.

[57] J. Lee, B. Moghaddam, H. Pfister, and R. Machiraju. A bilinear illumination model for robust face recognition. In *Proc. ICCV*, pages 1177–1184, 2005.

[58] K. Lee, J. Ho, and D. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(5):1–15, 2005.

[59] P. Y. Lee and J. B. Moore. Gauss-Newton-on-manifold for pose estimation. *J. Ind. Manag. Optim.*, 1(4):565–587, 2005.

[60] M. J. Leotta and J. L. Mundy. Predicting high resolution image edges with a generic, adaptive, 3-D vehicle model. In *Proc. CVPR*, pages 1311–1318, 2009.

[61] B. Lévy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic texture atlas generation. *ACM Trans. Graphic. (Proceedings of SIGGRAPH)*, 21(3):362–371, 2002.

[62] L. Liu, L. Zhang, Y. Xu, C. Gotsman, and S. J. Gortler. A local/global approach to mesh parameterization. *Comput. Graph. Forum*, 27(5):1495–1504, 2008.

[63] M. I. A. Lourakis. levmar: Levenberg-Marquardt nonlinear least squares algorithms in C/C++. `http://www.ics.forth.gr/~lourakis/levmar/`.

[64] W-C. Ma, T. Hawkins, P. Peers, C-F. Chabert, M. Weiss, and P. Debevec. Rapid acquisition of specular and diffuse normal maps from polarized spherical gradient illumination. In *Proc. Eurographics Symposium on Rendering*, pages 183–194, 2007.

[65] Y. Ma, J. Košecká, and S. Sastry. Optimization criteria and geometric algorithms for motion and structure estimation. *Int. J. Comput. Vis.*, 44(3):219–249, 2001.

[66] J. Manton. Optimization algorithms exploiting unitary constraints. *IEEE Trans. Signal Proces.*, 50(3):635–650, 2002.

[67] I. Matthews and S. Baker. Active appearance models revisited. *Int. J. Comput. Vis.*, 60(2):135–164, 2004.

[68] N. Max. Weights for computing vertex normals from facet normals. *J. Graph. Tools*, 4(2):1–6, 1999.

[69] V. E. McGee. Multidimensional scaling of n sets of similarity measures: A nonmetric individual differences approach. *Multivariate Behavioral Research*, 3(2):233–248, 1968.

[70] M. Meytlis and L. Sirovich. On the dimensionality of face space. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(7):1262–1267, 2007.

[71] B. Moghaddam, J. Lee, H. Pfister, and R. Machiraju. Model–based 3–D face capture with shape–from–silhouettes. In *Proc. IEEE Work. Analysis and Modeling of Faces and Gestures*, pages 20–27, 2003.

[72] U. Mohammed, S.J.D. Prince, and J. Kautz. Visio-lization: Generating novel facial images. *ACM Trans. Graphic. (Proceedings of SIGGRAPH)*, 28(3), 2009.

[73] A. Mohr and M. Gleicher. Deformation sensitive decimation. Technical report, University of Wisconsin, 2003.

[74] D. Nehab, S. Rusinkiewicz, J. Davis, and R. Ramamoorthi. Efficiently combining positions and normals for precise 3D geometry. In *Proc. SIGGRAPH*, pages 536–543, 2005.

[75] A. J. O'Toole, T. Vetter, H. Volz, and E. M. Salter. Three-dimensional caricatures of human heads: Distinctiveness and the perception of facial age. *Perception*, 26(6):719–732, 1997.

[76] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3D face model for pose and illumination invariant face recognition. In *Proc. AVSS*, pages 296–301, 2009.

[77] X. Pennec. Probabilities and statistics on Riemannian manifolds: Basic tools for geometric measurements. In *Proc. IEEE Workshop on Nonlinear Signal and Image Processing*, 1999.

[78] G. Peyré and L. D. Cohen. Geodesic computations for fast and accurate surface remeshing and parameterization. *Progress in Nonlinear Differential Equations and Their Applications*, 63:157–171, 2005.

[79] G. Peyré and L. D. Cohen. Geodesic remeshing using front propagation. *Int. J. Comput. Vis.*, 69(1):145–156, 2006.

[80] P. Phillips, W. Scruggs, A. O'Toole, P. Flynn, K. Bowyer, C. Schott, and M. Sharpe. FRVT 2006 and ICE 2006 large-scale results. Technical Report NISTIR 7408, NIST, 2007.

[81] E. Prados and O. Faugeras. A generic and provably convergent shape-from-shading method for orthographic and pinhole cameras. *Int. J. Comput. Vision*, 65(1-2):97–125, 2005.

[82] E. Praun, W. Sweldens, and P. Schrder. Consistent mesh parameterizations. In *Proc. SIGGRAPH*, pages 179–184, 2001.

[83] N. Ray and B. Lévy. Hierarchical least squares conformal maps. In *Proc. Pacific Conference on Computer Graphics and Applications*, pages 263–270, 2003.

[84] E. Reinhard, W. Heidrich, P. Debevec, S. Pattanaik, G. Ward, and K. Myszkowski. *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting.* Morgan Kaufmann, 2010.

[85] S. Romdhani, V. Blanz, and T. Vetter. Face identification by fitting a 3D morphable model using linear shape and texture error functions. In *Proc. ECCV*, pages 3–19, 2002.

[86] S. Romdhani, J. Ho, T. Vetter, and D. J. Kriegman. Face recognition using 3–D models: Pose and illumination. *Proc. of the IEEE*, 94(11):1977–1999, 2006.

[87] S. Romdhani and T. Vetter. Estimating 3D shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In *Proc. CVPR*, volume 2, pages 986–993, 2005.

[88] J. Rossignac and P. Borrel. Multi-resolution 3D approximations for rendering complex scenes. In *Modeling in Computer Graphics: Methods and Applications*, pages 455–465. Springer, 1993.

[89] P. Sander, J. Snyder, and H. Hoppe. Texture mapping progressive meshes. In *Proc. SIGGRAPH*, pages 409–416, 2001.

[90] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen. Decimation of triangle meshes. *SIGGRAPH Comput. Graph.*, 26(2):65–70, 1992.

[91] A. Sheffer, B. Lévy, M. Mogilnitsky, and A. Bogomyakov. ABF++: Fast and robust angle based flattening. *ACM Trans. Graphic. (Proceedings of SIGGRAPH)*, 24(2):311–330, 2005.

[92] A. Sheffer, E. Praun, and K. Rose. Mesh parameterization methods and their applications. *Foundations and trends in Computer Graphics and Vision*, 2(2):105–171, 2006.

[93] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression database. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(12):1615–1618, 2003.

[94] W. A. P. Smith and E. R. Hancock. Recovering facial shape using a statistical model of surface normal direction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(12):1914–1930, 2006.

[95] A. N. Tychonoff and V. Y. Arsenin. *Solution of Ill-posed Problems.* Winston and Sons, 1977.

[96] T. Valentine. A unified account of the effects of distinctiveness, inversion, and race in face recognition. *Quarterly Journal of Experimental Psychology A*, 43(2):161–204, 1991.

[97] D. Vlasic, M. Brand, H. H-P. Pfister, and J. Popović. Face transfer with multilinear models. *ACM Trans. Graphic. (Proceedings of SIGGRAPH)*, 24(3):426–433, 2005.

[98] Y. Wang, L. Zhang, Z. Liu, G.Hua, Z.Wen, Z.Zhang, and D.Samaras. Recovering facial shape using a statistical model of surface normal direction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(11):1968–1984, 2009.

[99] T. Weyrich, W. Matusik, H. Pfister, B. Bickel, C. Donner, C. Tu, J. McAndless, J. Lee, A. Ngan, H. Wann Jensen, and M. Gross. Analysis of human faces using a measurement-based skin reflectance model. *ACM Trans. Graphic. (Proceedings of SIGGRAPH)*, 25(3):1013–1024, 2006.

[100] P. L. Worthington and E. R. Hancock. New constraints on data–closeness and needle map consistency for shape–from–shading. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(12):1250–1267, 1999.

[101] J. Xiao, S. Baker, I. Matthews, and T. Kanade. Real–time combined 2D+3D active appearance models. In *Proc. CVPR*, pages 535–542, 2004.

[102] Y. Yang, J. Kim, F. Luo, S. Hu, and X. Gu. Optimal surface parameterization using inverse curvature map. *IEEE Trans. Vis. Comp. Gr.*, 14(5):1054–1066, 2008.

[103] L. Zhang and D. Samaras. Face recognition from a single training image under arbitrary unknown lighting using spherical harmonics. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(3):351–363, 2006.

[104] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz. Spacetime faces: High resolution capture for modeling and animation. *ACM Trans. Graphic. (Proceedings of SIGGRAPH)*, 23(3):548–558, 2004.

[105] R. Zhang, P. S. Tsai, J. E. Cryer, and M. Shah. Shape–from–shading: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(8):690–706, 1999.

[106] W. Y. Zhao and R. Chellappa. Illumination–insensitive face recognition using symmetric SFS. In *Proc. CVPR*, pages 286–293, 2000.

[107] W. Y. Zhao and R. Chellappa. Symmetric shape–from–shading using self–ratio image. *Int. J. Comput. Vision*, 45:55–75, 2001.

[108] K. Zhou, J. Snyder, B. Guo, and H.-Y. Shum. Iso-charts: Stretch-driven mesh parameterization using spectral analysis. In *Proc. Eurographics Symposium on Geometry Processing*, pages 47–56, 2004.

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

[110] J. Zivanov, P. Paysan, and T. Vetter. Facial normal map capture using four lights – An effective and inexpensive method of capturing the fine scale detail of human faces using four point lights. In *Proc. GRAPP*, pages 13–20, 2009.