Statistical Models of Shape

Rhodri Davies • Carole Twining • Chris Taylor

Statistical Models of Shape

Optimisation and Evaluation

Springer

Rhodri Davies
Division of Imaging Science
and Biomedical Engineering (ISBE)
University of Manchester
UK
rhodri.davies@manchester.ac.uk

Carole Twining
Division of Imaging Science
and Biomedical Engineering (ISBE)
University of Manchester
UK
carole.twining@manchester.ac.uk

Chris Taylor
Division of Imaging Science
and Biomedical Engineering (ISBE)
University of Manchester
UK
chris.taylor@manchester.ac.uk

ISBN: 978-1-84800-137-4 e-ISBN: 978-1-84800-138-1
DOI: 10.1007/978-1-84800-138-1

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Control Number: 2008933222

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

Springer Science+Business Media
springer.com

If fy rhieni, diolch am eich holl gefnogaeth.
Rhodri

To my wife Jo, and to the memory of my father.
Carole

To Jill, Andrew, and Alison, for keeping me sane.
Chris

Acknowledgements

Whilst undertaking the research that formed the basis for this book, Carole Twining and Rhodri Davies were funded by the MIAS[1] Inter-disciplinary Research Consortium (IRC) project, EPSRC grant No. GR/N14248/01, UK Medical Research Council Grant No. D2025/31.

During his doctoral training, which also contributed to that research, Rhodri Davies was funded on a BBSRC Industrial CASE studentship with AstraZeneca.[2] During the writing of this book, Rhodri Davies was fortunate to receive financial support from the Foulkes Foundation.[3]

A special thank you to Tomos Williams, Anna Mills, Danny Allen, and Tim Cootes (ISBE, Manchester) for their assistance, humour, and forebearance. Thanks also to John Waterton (AstraZeneca) and Alan Brett (Optasia Medical[4]).

A special mathematical thank you to Stephen Marsland (Massey University, New Zealand) for his many contributions to the development of this work, and for many fruitful arguments.

The hippocampus dataset was kindly provided by Martin Styner, Guido Gerig, and co-workers from the University of North Carolina, Chapel Hill, USA; the schizophrenia study from which this data was drawn was supported by The Stanley Foundation. The surfaces of the distal femur were provided by Tomos Williams, Chris Wolstenholme, and Graham Vincent of Imorphics.[5] The authors also wish to extend their thanks to their colleagues within the MIAS IRC, and to their ISBE colleagues at Manchester, for helping to provide the sort of intellectual environment in which excellent research can flourish.

---

[1] "From Medical Images and Signals to Clinical Information."

[2] AstraZeneca Pharmaceuticals, Alderley Park, Macclesfield, Cheshire, SK10 4TF.

[3] Foulkes Foundation Fellowship, 37 Ringwood Avenue, London, N2 9NT.

[4] Optasia Medical Ltd, Haw Bank House, High Street, Cheadle, SK8 1AL.

[5] Imorphics Ltd, Kilburn House, Lloyd Street North, Manchester Science Park, Manchester, M15 6SE.

Contents

Chapter 1
Introduction

The goal of image interpretation is to convert raw image data into meaningful information. Images are often interpreted manually. In medicine, for example, a radiologist looks at a medical image, interprets it, and translates the data into a clinically useful form. Manual image interpretation is, however, a time-consuming, error-prone, and subjective process that often requires specialist knowledge. Automated methods that promise fast and objective image interpretation have therefore stirred up much interest and have become a significant area of research activity.

Early work on automated interpretation used low-level operations such as edge detection and region growing to label objects in images. These can produce reasonable results on simple images, but the presence of noise, occlusion, and structural complexity often leads to erroneous labelling. Furthermore, labelling an object is often only the first step of the interpretation process. In order to perform higher-level analysis, a priori information must be incorporated into the interpretation process. A convenient way of achieving this is to use a flexible model to encode information such as the expected size, shape, appearance, and position of objects in an image.

The use of flexible models was popularized by the active contour model, or 'snake' [98]. A snake deforms so as to match image evidence (e.g., edges) whilst ensuring that it satisfies structural constraints. However, a snake lacks specificity as it has little knowledge of the domain, limiting its value in image interpretation.

More sophisticated models based on the physical properties of an object have also been proposed (e.g., [134]). However, the expected patterns of variation of the model are usually estimated from only a single prototype, which requires many assumptions to be made. A more promising approach – and that followed in this book – is to use statistical models that attempt to *learn* the actual patterns of variability found in a class of objects, rather than making arbitrary assumptions. The idea is to estimate the population statistics from a *set* of examples instead of using a single prototype. The pattern of variation for a given class of object is established from a training set and

R. Davies et al., *Statistical Models of Shape*,
DOI: 10.1007/978-1-84800-138-1_1,

statistical analysis is used to give an efficient parameterisation of this variability, providing a compact and efficient representation.

The starting point in the construction of a statistical model is usually a training set of segmented images. In order to calculate statistics across the training set, a correspondence must be established between each member. It is important to choose the correct correspondence, otherwise a poor representation of the modelled object will result.

Correspondence is often established manually, but this is a time-consuming process that presents a major bottleneck in model construction. The manual definition of correspondence is also restricted to two-dimensional objects, which limits their use in interpreting medical images, since many of these are three dimensional. Other approaches to model-building have also been proposed, but these do not produce correspondences that are correct in any obvious way and the models that they produce are of limited utility.

This book presents a generic solution to this correspondence problem by treating it as part of the learning process. We will see that the key is to treat model construction as an optimisation problem, thus automating the process and guaranteeing the effectiveness of the resulting models. The other subject covered in this book is the evaluation of statistical models. This is also an important aspect of modelling since it allows us to quantify the likely utility of the model in practical applications. Model evaluation methods are established for cases with ground truth or in its absence.

In the remainder of this first chapter, we will take a look at some practical problems where statistical models have been applied before an overview of the rest of the book is presented.

1.1 Example Applications of Statistical Models

Statistical models have been used to successfully solve a wide range of practical problems, from Chinese character recognition [163] to cardiac modelling [72]. The number of applications is vast, but here we will focus on a few interesting examples and concentrate on the properties of statistical models that have allowed them to be successfully applied.

1.1.1 Detecting Osteoporosis Using Dental Radiographs

Statistical shape models were originally conceived as a basis for automatic image segmentation [31] – the process of labelling an image so that the labels correspond to real-world objects. A big advantage of using a shape model for this task is that it can produce extremely accurate and objective segmentations. They can therefore be used to detect changes that might otherwise be

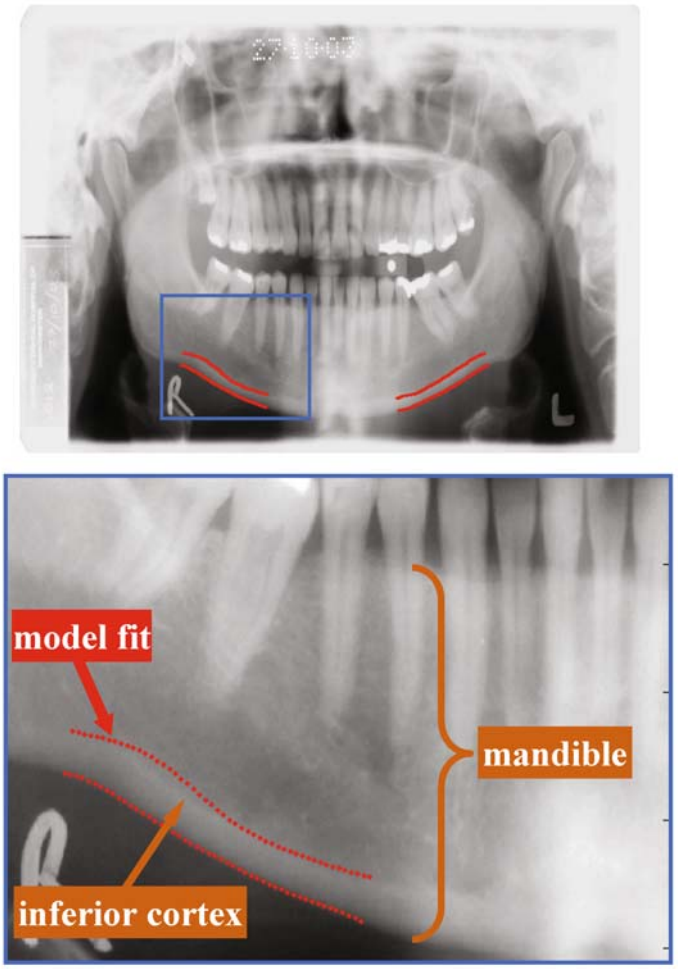

Fig. 1.1 An example of a segmentation of the inferior mandibular cortex on a panoramic dental radiograph using the method described in [1]. **Top**: a panoramic radiograph, where the red lines represent the segmentation of the left and right inferior mandibular cortex using a shape model. **Bottom**: Detail of the segmentation on the right side of the patient's mandible. Figure courtesy of P.D. (Danny) Allen, University of Manchester.

missed by human annotation. An example of where this additional accuracy has proved to be critical is in detecting osteoporosis in dental radiographs [1, 59].

Osteoporosis is a common disorder that causes a reduction in bone tissue density, leading to brittle bones that are prone to fracture. The standard method of diagnosis involves a dual energy x-ray absorbiometry (DXA) scanner, but these are dedicated machines with limited accessibility. Although

access is improving, only patients with a high index of clinical suspicion are referred for DXA scans, resulting in missed diagnoses.

It has been reported that osteoporosis can also be detected by careful measurement of the width of the inferior mandibular cortex on panoramic dental radiographs – a common investigation in dental practice. However, if the cortical width is measured manually, the time taken impedes the dental practitioner from performing the test during a routine consultation. Manual measurement also introduces considerable inter- and intra-operator variability, resulting in reduced sensitivity and specificity of detection. However, it has been shown [1, 59] that this variability can be reduced by using a statistical shape model for segmentation – an example of a segmentation of the inferior mandibular cortex using a statistical shape model is shown in Fig. 1.1. The accuracy of the resulting segmentations was shown to be sufficient to diagnose skeletal osteoporosis with good diagnostic ability and reliability [59]. Furthermore, measurement was performed in real time with minimal human interaction. This application thus promises another means of detecting osteoporosis, or at least of flagging high-risk cases for referral to DXA scanning.

1.1.2 Detecting Vertebral Fractures

Statistical models provide a compact and efficient basis for describing the variation of a class of object. Model parameter values can therefore be used as effective features in classifier systems. We already know a bit about osteoporosis from the previous section, so we will now look at an example of where complications of the disease can be detected using a model-based classifier.

A common complication of osteoporosis is a fractured vertebra. Although many of these fractures are asymptomatic, they are an important indicator of more harmful fractures in the future. Diagnosis of a fracture is usually made by a radiologist, but, as with any human operator, they are liable to report subjective results. Also, vertebral fracture assessment by DXA scanners is becoming common in places other than radiology units (e.g., general practice), and may not be carried out by an expert radiologist. Therefore, it is desirable to establish quantitative criteria that capture some of the subtle information used by an expert, since current (vertebral) height-based quantitative methods are insufficiently specific, especially in diagnosing the mild fractures that occur in the early stages of the disease. These height-based measures do not capture subtle shape information, nor other features present in the image. A recent body of work [167, 145, 146] has shown that using statistical models of shape and appearance can offer substantial improvements in diagnostic accuracy over conventional quantitative methods.

The first step in the system is to segment the vertebrae of interest using a statistical model – an example of a typical segmentation result is given in Fig. 1.2. As with the example of the inferior mandibular cortex given

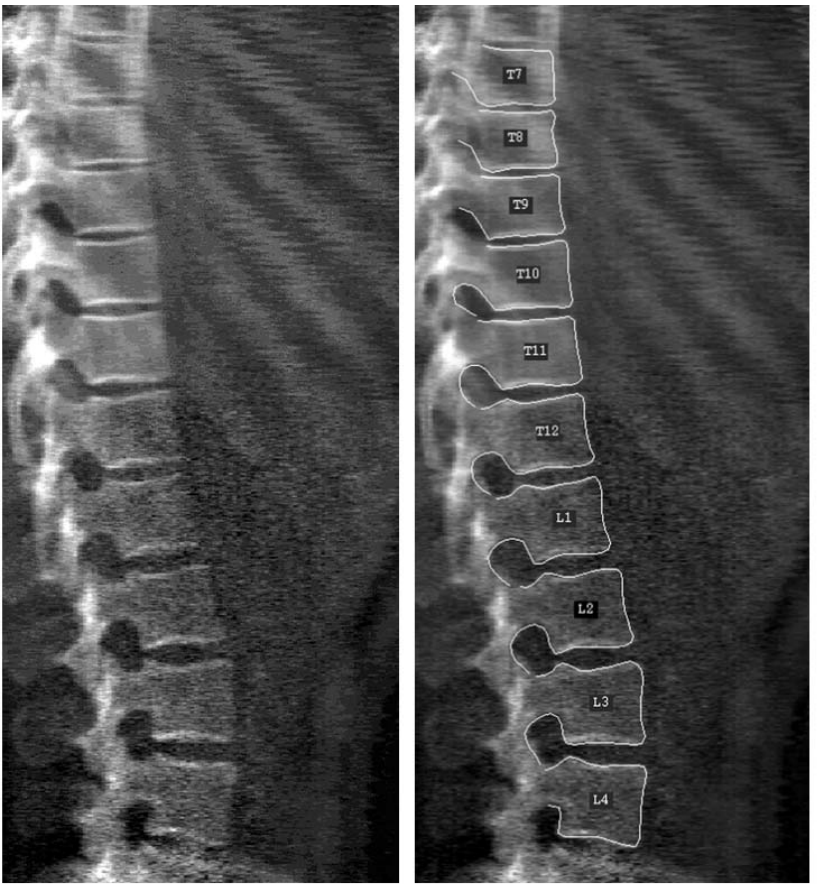

Fig. 1.2 An example of a segmentation of human vertebrae using the method described in [146]. **Left**: a DXA image showing the spine from the seventh thoracic vertebra (T7) down to the fourth lumbar vertebra (L4). **Right**: the segmentation achieved by the model, overlaid on the image. Figure courtesy of Martin Roberts, ISBE, University of Manchester.

above, this segmentation is quick, requires minimal user interaction, and, most importantly, produces more objective and reproducible results than manual segmentation.

The model parameters found in segmentation can then be fed into a trained classifier, which will return a positive or negative diagnosis of vertebral fracture. The advantage of using the model parameters, rather than other established measures (such as vertebral height), is that they capture much more information about the state of the vertebrae such as their shape, appearance, and pose. This information forms a much stronger basis for the classifier system.

1.1.3 Face Identification, Tracking, and Simulation of Ageing

We have said in the introduction that many non-statistical modelling approaches infer object properties from a single prototype. This approach has several disadvantages over the statistical approach – one of these is that it lacks the flexibility to model objects that exhibit significant variability. An example of where this is particularly evident is in human face analysis.

Face analysis, which includes tasks such as finding, identifying, and tracking someone's face in an image, has application in many fields – from aiding human-computer interaction to security surveillance monitoring. However, face analysis is a difficult task, not least because of the huge inherent variability of a face – not just between individuals, but variation in the appearance of the same person's face due to changes in expression, pose, lighting, etc.

A statistical model has the ability to learn this variability from a set of examples – as long as enough examples are presented the model will be generic enough to deal with these variations. Such a model of facial appearance was presented in [64] and was shown to perform well in finding, identifying, and tracking faces in images.

Statistical models can also be used in generative mode, where the model is used to synthesize new examples of the class of object. This opens the possibility of using the models in different types of applications to those that we have seen so far. A good example is described in [104], which shows how facial ageing can be simulated artificially. So, given a photograph of a

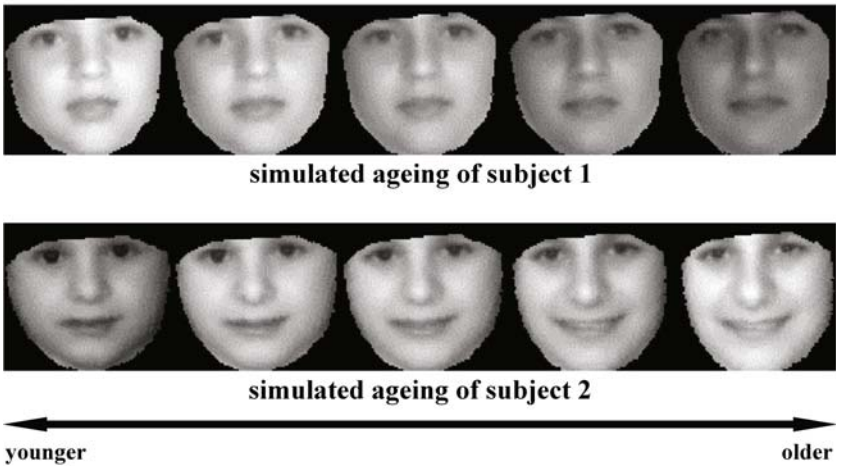

Fig. 1.3 Simulated ageing using the method described in [104]. The figure shows the effect of simulating ageing in photographs of two individuals. Figure reproduced from Fig. 4 in [104] with permission (©2002 IEEE).

person's face, the model can generate impressions of that person at different ages – some examples are shown in Fig. 1.3. This has application in ageing photographs of missing or wanted persons as well as improving robustness of face analysis to ageing.

1.2 Overview

The organization of the remainder of this book is as follows.

We start with a theoretical treatment of statistical shape models in Chap. 2. This contains a comprehensive description of model-building, and considers both discrete and continuous representations of shape. Several other aspects of model-building, such as shape alignment and practical applications, are also covered.

Chapter 3 covers a fundamental problem in shape modelling in greater detail: that of establishing a correspondence between a set of shapes. The chapter begins by illustrating the importance of establishing a suitable correspondence, before looking at various ways in which this can be achieved. The last part of the chapter introduces our approach of model-building within an optimisation framework. This allows correspondence to be established by viewing it as an integral part of the learning process.

One essential component of this optimisation approach to model-building is an objective function that quantifies what is meant by the quality of a model. The subject of objective functions is covered in Chap. 4; we look at various objective functions that can be used to establish correspondence, including a full derivation of the Minimum Description Length objective function (and various approximations to it).

In order to minimise our chosen objective function, we must be able to manipulate correspondence across the set of shapes. At the same time, we must ensure that only valid correspondences are generated. In Chap. 5, we show how this can be achieved by re-parameterising each shape. Several representations of curve re-parameterisation are described in Chap. 5. The extension to surface re-parameterisation is complicated by the need for explicit surface parameterisation, but a generic method of achieving this along with several representations of re-parameterisation for surfaces is given in Chap. 6.

The final component of the optimisation approach is a method of finding the configuration of re-parameterisation functions that lead to the optimal value of the objective function. This problem is explored in Chap. 7, which presents a generic optimisation approach, as well as looking at how it can be tailored for certain situations.

Chapter 8 explores an alternative approach to representing and manipulating correspondence, using a non-parametric approach – the goal being to produce more robust results in less time. A fluid-based regularizer is described as well as an efficient method of optimisation.

Finally, Chap. 9 considers the question of how we should evaluate models of shape or models of appearance. In particular, we address the question of whether ground truth data should be used for evaluation, and how models can be evaluated in the absence of such ground truth data.

Chapter 2
Statistical Models of Shape and Appearance

As was explained in the Introduction, our aim is to start with a set of example shapes (the *training set*), and learn from this the patterns of variability of the shape of the class of objects for which the training set can be considered a representative sample. We will first consider this problem in a rather abstract sense, and illustrate how the question of correspondence *between* different shapes is inextricably linked to the question of representing a set of shapes.

Mathematically, the usual approach is to construct a mapping from an example shape to a point in some shape space. This is the process of constructing a *representation* of shape. The idea is that every physical shape corresponds to a point in shape space, and conversely, each point in shape space corresponds to some physical shape. There are many ways of constructing such representations, but whichever method is chosen, what is then obtained is a mapping from our training set of shapes to a set of points in shape space.

Modelling can then be considered as the process of modelling the distribution of our training points in shape space. However, before we can begin to talk about the distribution of such points, we first need to define a notion of distance on shape space.

A definition of a distance on shape space then leads directly to the notion of correspondence between the physical shapes themselves. Consider two distinct physical shapes, and the two points in shape space which represent those two shapes. We can then imagine a continuous path between the two points in shape space, and, given that we have a definition of distance, the shortest such path between the two shapes. When we map this construction back to the space of physical shapes, what we obtain is a continuous sequence of physical shapes that interpolates between our two original shapes. If we now consider a single point on one shape, we can then follow it through this continuous sequence of shapes, and hence locate the physical point on the other shape to which this point corresponds. This is what is meant by a dense correspondence between shapes.

Let us now return to our physical shapes, and imagine translating and rotating a physical shape (that is, altering the *pose* of the shape). In many

R. Davies et al., *Statistical Models of Shape*,
DOI: 10.1007/978-1-84800-138-1_2,

cases, the pose of the physical shape is unimportant, and what we mean by *change* of shape is not such a transformation.

But there is another transformation we could consider. Suppose we imagine two *distinct* points in shape space, which give the same *physical* shape, but different correspondence when compared to some other (reference) shape (using the construction defined above). If such a construction is possible, we see that it is possible (at least in theory) to manipulate the correspondence between shapes (moving the point in shape space), whilst leaving the physical shape unaltered. Which then means that we have to answer the question as to what correspondence we should use for our analysis of shape variability (the *correspondence problem* – see also Sect. 3.1).

We hence see that the issue of shape correspondence naturally arises as soon as we consider the steps necessary to represent a set of shapes, and analyse their distribution. Some methods of shape representation do not allow correspondence to be manipulated independently of shape, and in these cases, the correspondence they generate can be considered as *implicit* (for example, the SPHARM method [76], or early M-Rep methods [137]). However, there are other methods of shape representation for which the correspondence is *explicit*, which allow correspondence to be manipulated independently of physical shape.

In the remainder of this book, we will restrict ourselves to such a shape representation, the shape representation which leads to the class of deformable models known as *Statistical Shape Models*[1] (SSMs). We will now describe this shape representation in detail, beginning with the finite-dimensional case.

2.1 Finite-Dimensional Representations of Shape

Let us consider first building a finite-dimensional representation of a single shape S. The most intuitive and simplest way to represent such a shape is a join-the-dots approach.

We take a set of n_P points which lie on the shape S, with positions:

$$\mathbf{x}^{(i)} \in S, \;\; i = 1, \ldots n_P. \tag{2.1}$$

The coordinates of each point position can be concatenated to give a single *shape vector* $\mathbf{x} = \{\mathbf{x}^{(i)}\}$. For example:

$$\mathbf{x} \doteq (x^{(1)}, y^{(1)}, z^{(1)}, x^{(2)}, y^{(2)}, z^{(2)}, \ldots, x^{(n_P)}, y^{(n_P)}, z^{(n_P)}), \;\; S \subset \mathbb{R}^3, \tag{2.2}$$

[1] Note that these were initially called *Point Distribution Models (PDMs)*. However, due to a clash with nomenclature in the statistics literature, they were later re-christened *Statistical Shape Models (SSMs)*. Both terms can be found in the literature.

where $(x^{(i)}, y^{(i)}, z^{(i)})$ are the Cartesian coordinates of the i^{th} point on the shape. For a shape S in $\mathbb{R}^d$, this gives a $d \times n_P$-dimensional representation. In most cases, Cartesian coordinates are sufficient, but in cases where parts of shapes can rotate, it may be useful to instead use angular coordinates [87].

The final representation of the shape is then generated from the shape vector by interpolation. For shapes in $\mathbb{R}^2$ (curves), this is a simple join-the-dots approach, using either straight-line segments (polygonal representation), or by spline interpolants if a smoother shape is preferred. For shapes in $\mathbb{R}^3$ (surfaces), interpolants can similarly also be linear (planes), or a higher-order spline interpolant.

What we have not considered so far is the connectivity of the points, and the topology of the shape. For the case of shapes in $\mathbb{R}^2$, the simplest case is where the shape has only one connected component, with the topology of either an open or closed line. The points are usually numbered so that they are connected consecutively – for the closed shapes, we must also form a loop by connecting the last point to the first. For more complicated multi-part shapes, the points which are members of each part, and the connectivity within each part have to be specified separately.

Similar considerations holds for shapes in $\mathbb{R}^3$. The simplest case is then single-part shapes in $\mathbb{R}^3$, with the topology of either open surfaces or spheres, with the points being part of a triangulated mesh.

Once we have a finite-dimensional representation of a single shape S, we can easily see how this can be extended to form a *common* representation of a *set* of shapes. To be specific, let us take a set of n_S shapes:

$$S_i : i = 1, \ldots n_S. \tag{2.3}$$

We suppose that each shape is then represented by a set of n_P points, such that the individual points are placed in *corresponding* positions across the set of shapes. This then gives us a set of initial shape vectors $\{\mathbf{x}_i : i = 1, \ldots n_S\}$ which form a representation of the whole set of shapes in a common shape space $\mathbb{R}^{dn_P}$.

2.1.1 Shape Alignment

In many cases, the size, placement, and orientation of an object is arbitrary, and has nothing to do with the actual variation of shape that we are interested in. In mathematical terms, there are degrees of freedom (scaling, translation, and rotation) associated with each shape example, which we wish to factor out of our shape analysis.

Consider a fixed shape $\mathbf{y}$, and a second moving shape $\mathbf{x}$, which we wish to align with the first by means of a similarity transformation. A general similarity transformation acting on $\mathbf{x}$ can be written as:

$$\mathbf{x} \mapsto s\mathbf{R}(\mathbf{x} - \mathbf{t}), \tag{2.4}$$

where $\mathbf{t}$ represents a translation in $\mathbb{R}^d$, $\mathbf{R}$ is a $dn_P \times dn_P$ representation of a rotation in $\mathbb{R}^d$, and $s \in \mathbb{R}^+$ is a scaling. Note that these elements of the representation of a similarity transformation are such that they act on the concatenated set of shape points in the shape vector. They are constructed from a representation that acts on single points in the obvious way, although the exact details depend on the way in which the coordinates of the shape points have been concatenated.

We wish to find the similarity transformation which brings the moving shape $\mathbf{x}$ as close as possible to the fixed shape $\mathbf{y}$. The simplest way to define proximity is just the magnitude of the Euclidean norm of the difference between the two shape vectors in $\mathbb{R}^{dn_P}$:

$$\mathcal{L} \doteq \|\mathbf{y} - s\mathbf{R}(\mathbf{x} - \mathbf{t})\|^2, \tag{2.5}$$

which is the square of the Procrustes distance between the shapes [78]. In terms of the positions of individual points, this expression can be rewritten as:

$$\mathcal{L} = \sum_{i=1}^{n_P} \|\mathbf{y}^{(i)} - s\mathbf{R}(\mathbf{x}^{(i)} - \mathbf{t})\|^2, \tag{2.6}$$

where $\mathbf{t}$ is now just a vector in $\mathbb{R}^d$, and $\mathbf{R}$ is a $d \times d$ rotation matrix.

If we define our origin so that it lies at the centre of mass of the fixed shape:

$$\frac{1}{n_P} \sum_{i=1}^{n_P} \mathbf{y}^{(i)} = \mathbf{0}, \tag{2.7}$$

with rotation defined about this origin, the optimal translation can then be calculated as:

$$\left.\frac{\partial \mathcal{L}}{\partial \mathbf{t}}\right|_{s,\mathbf{R}} = 0 \implies \left.\frac{\partial}{\partial \mathbf{t}}\right|_{s,\mathbf{R}} \sum_{i=1}^{n_P} \|s\mathbf{R}(\mathbf{x}^{(i)} - \mathbf{t})\|^2 = \mathbf{0}, \tag{2.8}$$

$$\implies \left.\frac{\partial}{\partial \mathbf{t}}\right|_{s,\mathbf{R}} \sum_{i=1}^{n_P} \|(\mathbf{x}^{(i)} - \mathbf{t})\|^2 = \mathbf{0}, \tag{2.9}$$

$$\implies \boxed{\mathbf{t} = \frac{1}{n_P} \sum_{i=1}^{n_P} \mathbf{x}^{(i)}} = \frac{1}{n_P} \sum_{i=1}^{n_P} \left(\mathbf{x}^{(i)} - \mathbf{y}^{(i)}\right). \tag{2.10}$$

That is, the centroid/centre of mass of the original moving shape is translated so that it coincides with the centre of mass of the fixed shape.

Once the shapes have been centred, we can then calculate the combined scaling and rotation:

$$\frac{\partial \mathcal{L}}{\partial s\mathbf{R}} = 0 \implies \boxed{\sum_{i=1}^{n_P} y_\mu^{(i)} x_\beta^{(i)} = sR_{\mu\alpha} \sum_{j=1}^{n_P} x_\alpha^{(j)} x_\beta^{(j)},} \tag{2.11}$$

where $\mathbf{x}^{(i)} = \{x_\alpha^{(i)} : \alpha = 1, \dots d\}$ and $\mathbf{y}^{(i)} = \{y_\alpha^{(i)} : \alpha = 1, \dots d\}$ are the Cartesian components of the point positions. This can then be solved for the matrix $s\mathbf{R}$ (for example, see [102] for further details).

Rather than aligning just a pair of shapes, we wish to mutually align an entire set of shapes $\{S_i : i = 1, \dots n_S\}$, $S_i = \{\mathbf{x}_j^{(i)} : j = 1, \dots n_P\}$. We use a similar criterion to that considered above, either by considering the squared Procrustes distances between all pairs of shapes, or between all shapes and the mean shape. This is known as generalized Procrustes analysis. The translations are obtained as before, centering each shape on the origin. However, the general problem of finding the optimal rotations and scalings is not well-posed unless further constraints are placed on the mean [174], as will be explained below.

For statistical shape analysis and statistical shape models, a simple iterative approach is usually sufficient. After first centering all the shapes, a typical algorithm then proceeds [38] as Algorithm 2.1.

Algorithm 2.1 : Mutually Aligning a Set of Shapes.

Initialize:

- Choose one shape as the reference frame, call it $\mathbf{x_{ref}}$, and retain this.
- Normalize the scale so that $\|\mathbf{x_{ref}}\| = \mathbf{1}$.
- Set the initial estimate of the mean shape to be $\mathbf{x_{ref}}$.

Repeat:

- Perform pairwise alignment of all shapes to the current estimate of the mean shape.
- Recompute the mean of the set of shapes:

  $\bar{\mathbf{x}} \doteq \{\bar{\mathbf{x}}^{(i)} : i = 1, \dots n_P\}$, $\bar{\mathbf{x}} \doteq \dfrac{1}{n_S} \displaystyle\sum_{j=1}^{n_S} \mathbf{x_j}$.
- Align $\bar{\mathbf{x}}$ to the initial reference frame $\mathbf{x_{ref}}$.
- Normalize the mean so that $\|\bar{\mathbf{x}}\| = 1$.

Until convergence.

Note that it is necessary to retain the initial reference frame to remove the global degree of freedom corresponding to rotating all the shapes by the same amount. Setting $\|\bar{\mathbf{x}}\| = 1$ similarly removes the degree of freedom associated with scaling all the shapes by the same factor. The degrees of freedom associated with a uniform translation have already been removed by centering all the shapes before we began the rest of the alignment.

There remains the question of what transformations to allow during the iterative refinement. A common approach is to scale all shapes so that $\|\mathbf{x}_i\| = 1$, and allow only rotations during the alignment stage. This means that from the original shape space $\mathbb{R}^{dn_P}$, all shapes have been projected onto the surface of a hypersphere $\|\mathbf{x}\| = 1$. This means that the submanifold of $\mathbb{R}^{dn_P}$ on which the aligned shapes lie is curved, and if large shape changes occur, significant non-linearities can appear. This may be problematic when we come to the next stage of building a statistical model of the distribution of shapes. An alternative is to allow both scaling and rotation during alignment, but this can also introduce significant non-linearities. If this is a problem, the non-linearity can be removed by projecting the aligned shapes onto the tangent hyperplane to the hypersphere at the mean shape. That is:

$$\mathbf{x}_i \mapsto s_i \mathbf{x}_i, \;\; s_i \in \mathbb{R}^+ \text{ such that } (\bar{\mathbf{x}} - s_i \mathbf{x}_i) \cdot \bar{\mathbf{x}} = 0. \tag{2.12}$$

See [38] for further details and explicit examples.

2.1.2 Statistics of Shapes

To summarize our progress so far, we have mapped our initial shape vectors (2.2) in $\mathbb{R}^{dn_P}$ to a new set of mutually aligned shape vectors, by factoring out uninteresting degrees of freedom corresponding to pose (scale, orientation, and position). We now wish to analyse the statistics of this distribution of shape vectors. To do this, we first need to find a set of axes specific to the particular set of shapes. We have in some sense already started to perform this, since we have a mean shape $\bar{\mathbf{x}}$ that can be used as an origin.

To see that this is a necessary procedure, consider the extreme case where there is a shape point, $\mathbf{x}^{(i)}$ say, which does not change its position across the set of examples. Since this point does not vary, there is no value in retaining the axes corresponding to the coordinates of this point $\{x_\alpha^{(i)} : \alpha = 1, \dots d\}$. We wish instead to find a new set of axes in $\mathbb{R}^{dn_P}$ that span the subspace which contains the (aligned) shapes. One simple procedure for performing this task is Principal Component Analysis (PCA).

2.1.3 Principal Component Analysis

We start from our set of shape vectors $\{\mathbf{x}_i : i = 1, \dots n_S\}$ (we will assume from now on that we are only considering sets of shape vectors which have been aligned), with components relative to our original axes:

$$\mathbf{x}_i = \{x_{i\mu} : \mu = 1, \dots d \times n_P\}. \tag{2.13}$$

These are the components and axes defined by those in $\mathbb{R}^d$, the original space in which the input shapes reside.

We wish to find a new set of orthogonal axes in $\mathbb{R}^{dn_P}$ that better reflects the actual distribution of the set. The origin of this new set of axes will be set to the mean shape $\bar{\mathbf{x}}$. Let these new axes be described by a set of orthonormal vectors:

$$\{\mathbf{n}^{(a)}\} \text{ such that } \mathbf{n}^{(a)} \cdot \mathbf{n}^{(b)} = \delta_{ab}, \tag{2.14}$$

where δ_{ab} is the Kronecker delta.

We then have the following theorem:

Theorem 2.1. PCA.
The set of orthonormal directions $\{\mathbf{n}^{(a)}\}$ *that maximises the quantity:*

$$\mathcal{L} \doteq \sum_a \sum_{i=1}^{n_S} \left((\mathbf{x}_i - \bar{\mathbf{x}}) \cdot \mathbf{n}^{(a)} \right)^2, \tag{2.15}$$

are given by the eigenvectors of the data covariance matrix $\mathbf{D}$ *for the shapes, where we define* $\mathbf{D}$ *of size* $dn_P \times dn_P$ *with components:*

$$\boxed{D_{\mu\nu} \doteq \sum_{i=1}^{n_S} (\mathbf{x}_i - \bar{\mathbf{x}})_\mu (\mathbf{x}_i - \bar{\mathbf{x}})_\nu.} \tag{2.16}$$

Then the eigenvectors are defined by:

$$\mathbf{D}\mathbf{n}^{(a)} = \lambda_a \mathbf{n}^{(a)}, \;\; a = 1, \ldots n_S - 1. \tag{2.17}$$

Proof. Suppose we are extracting these vectors in some sequential manner, so that having found an acceptable subset $\{\mathbf{n}^{(a)} : a = 1, \ldots b-1\}$*, we now wish to make the optimum choice of the next vector* $\mathbf{n}^{(b)}$*. Optimality is then given by maximising:*

$$\mathcal{L} \doteq \sum_{i=1}^{n_S} \left((\mathbf{x}_i - \bar{\mathbf{x}}) \cdot \mathbf{n}^{(b)} \right)^2, \tag{2.18}$$

with respect to $\mathbf{n}^{(b)}$*, subject to the orthonormality constraints:*

$$\mathbf{n}^{(a)} \cdot \mathbf{n}^{(b)} = \delta_{ab}, \;\; a = 1, \ldots b. \tag{2.19}$$

Using Lagrange multipliers $\{c_{ba} : a = 1, \ldots b\}$*, the solution to this constrained optimisation problem corresponds to the stationary point of the function:*

$$\mathcal{L} = \sum_{i=1}^{n_S} \left((\mathbf{x}_i - \bar{\mathbf{x}}) \cdot \mathbf{n}^{(b)} \right)^2 + \sum_{a=1}^{b-1} c_{ba} \mathbf{n}^{(a)} \cdot \mathbf{n}^{(b)} + c_{bb} \left(\mathbf{n}^{(b)} \cdot \mathbf{n}^{(b)} - 1 \right). \quad (2.20)$$

$$\frac{\partial \mathcal{L}}{\partial c_{ba}} = 0 \implies \mathbf{n}^{(a)} \cdot \mathbf{n}^{(b)} = \delta_{ab}, \text{ which are the required constraints.} \quad (2.21)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{n}^{(b)}} = 0 \implies 2 \sum_{i=1}^{n_S} (\mathbf{x}_i - \bar{\mathbf{x}})_\nu (\mathbf{x}_i - \bar{\mathbf{x}})_\mu n_\mu^{(b)} + \sum_{a=1}^{b-1} c_{ba} n_\nu^{(a)} + 2 c_{bb} n_\nu^{(b)} = 0, \quad (2.22)$$

where we use the Einstein summation convention[2] *that the repeated index* μ *is summed from* $\mu = 1$ *to* dn_P*. Using the definition of the covariance matrix* $\mathbf{D}$ *(2.16), we can rewrite the condition as:*

$$2\mathbf{D}\mathbf{n}^{(b)} + \sum_{a=1}^{b-1} c_{ba} \mathbf{n}^{(a)} + 2 c_{bb} \mathbf{n}^{(b)} = \mathbf{0}. \quad (2.23)$$

For the case $b = 1$ *(the first direction we choose), this reduces to:*

$$\mathbf{D}\mathbf{n}^{(1)} + c_{11} \mathbf{n}^{(1)} = 0 \quad (2.24)$$

$$\implies \boxed{\mathbf{D}\mathbf{n}^{(1)} = \lambda_1 \mathbf{n}^{(1)}} \;\&\; \mathbf{n}^{(1)}\mathbf{D} = \lambda_1 \mathbf{n}^{(1)}, \; c_{11} \doteq \lambda_1. \quad (2.25)$$

That is, the vector $\mathbf{n}^{(1)}$ *is a left and right eigenvector of the (symmetric) shape covariance matrix* $\mathbf{D}$*, with eigenvalue* λ_1*. The condition for the second axis can then be written as:*

$$2\mathbf{D}\mathbf{n}^{(2)} + c_{21} \mathbf{n}^{(1)} + 2 c_{22} \mathbf{n}^{(2)} = \mathbf{0}. \quad (2.26)$$

Taking the dot product of this expression with $\mathbf{n}^{(1)}$*, we obtain:*

$$2\mathbf{n}^{(1)}\mathbf{D}\mathbf{n}^{(2)} + c_{21} = 0 \quad (2.27)$$

$$\implies 2\lambda_1 \mathbf{n}^{(1)} \cdot \mathbf{n}^{(2)} + c_{21} = 0 \implies c_{21} = 0. \quad (2.28)$$

$$\therefore \; \mathbf{D}\mathbf{n}^{(2)} + c_{22} \mathbf{n}^{(2)} = \mathbf{0} \implies \boxed{\mathbf{D}\mathbf{n}^{(2)} = -c_{22} \mathbf{n}^{(2)} \doteq \lambda_2 \mathbf{n}^{(2)}.} \quad (2.29)$$

It then follows by induction that the required set of axes $\{\mathbf{n}^{(a)}\}$ *are the orthonormal set of eigenvectors of the shape covariance matrix* $\mathbf{D}$*.* □

The sum of the squares of the components of the shape vectors along each of the PCA directions $\mathbf{n}^{(a)}$ is then given by:

[2] Note that, in general, indices that appear in brackets $.^{(a)}$ will *not* be summed over unless explicitly stated. See Glossary.

$$\sum_{i=1}^{n_S} \left((\mathbf{x}_i - \bar{\mathbf{x}}) \cdot \mathbf{n}^{(a)} \right)^2 = n_\mu^{(a)} D_{\mu\nu} n_\nu^{(a)} = \lambda_a \geq 0. \tag{2.30}$$

This means that the set of axes can be ordered in terms of relative importance by sorting the eigenvalues in terms of decreasing size. Since there are n_S shapes, there are at most $n_S - 1$ non-zero eigenvalues. This means that for the case $n_S - 1 < dn_P$, we have performed dimensionality reduction by locating the directions with zero eigenvalue that are orthogonal to the subspace spanned by the data.

In practice, we retain not just the directions corresponding to non-zero eigenvalues, but instead that ordered set which encompasses a certain amount of the total variance of the data.

$$\text{Ordered set of eigenvalues:} \quad \lambda_1 \geq \lambda_2, \ldots \geq \lambda_{dn_P}, \tag{2.31}$$

$$\text{Total variance:} \quad \sum_{a=1}^{n_S - 1} \lambda_a, \tag{2.32}$$

$$\text{Variance up to } n_m: \quad \sum_{a=1}^{n_m} \lambda_a. \tag{2.33}$$

The number of modes n_m retained is then chosen to be the lowest value such that the variance up to n_m is some specified fraction of the total variance.

We can also transform coordinates to the system defined by the directions $\{\mathbf{n}^{(a)}\}$, with origin $\bar{\mathbf{x}}$. For each shape $\mathbf{x}_i$ this then defines a new vector of shape parameters $\mathbf{b}^{(i)} \in \mathbb{R}^{n_m}$ thus:

$$\mathbf{b}^{(i)} = \{b_a^{(i)} : a = 1, \ldots n_m\}, \;\; b_a^{(i)} \doteq \left(\mathbf{n}^{(a)} \cdot (\mathbf{x}_i - \bar{\mathbf{x}}) \right), \tag{2.34}$$

where the covariance in this frame is now given by the diagonal matrix:

$$\boxed{D_{ab} \doteq \sum_{i=1}^{n_S} (\mathbf{n}^{(a)} \cdot \mathbf{b}^{(i)})(\mathbf{n}^{(b)} \cdot \mathbf{b}^{(i)}) = \lambda_a \delta_{ab}.} \tag{2.35}$$

We define the matrix of eigenvectors:

$$\boxed{\mathbf{N}, \;\; N_{\mu a} \doteq n_\mu^{(a)},} \tag{2.36}$$

which is then of size $dn_P \times n_m$. We can then form an approximate reconstruction of the shape vector $\mathbf{x}_i$ from the corresponding parameter vector $\mathbf{b}^{(i)}$ thus:

$$\mathbf{x}_i \approx \bar{\mathbf{x}} + \mathbf{N}\mathbf{b}^{(i)}. \tag{2.37}$$

The reconstruction is only approximate, since we have only retained the first n_m eigenvectors, rather than all eigenvectors with non-zero eigenvalue.

The matrix $\mathbf{N}$ performs a mapping from the coordinate axes defined in (shape) parameter space to the original shape space. The mean shape simply performs a translation of the origin, since the origin of parameter space is taken to correspond to the mean shape. The corresponding backwards mapping, from shape space to parameter space, is performed by the matrix $\mathbf{N}^T$. For a general parameter vector $\mathbf{b} \in \mathbb{R}^{n_m}$ and shape vector $\mathbf{x} \in \mathbb{R}^{dn_P}$:

$$\boxed{\mathbf{b} \mapsto \bar{\mathbf{x}} + \mathbf{N}\mathbf{b}, \;\; \mathbf{x} \mapsto \mathbf{N}^T (\mathbf{x} - \bar{\mathbf{x}}) .} \tag{2.38}$$

Note however that the mappings are not the inverse of each other, even if all the variance is retained, since the dimensionality of parameter space is less than the dimensionality of shape space. For a shape vector $\mathbf{x}$ which is not part of the original training set, the action of $\mathbf{N}^T$ first projects the shape vector into the subspace spanned by the training set, then forms an (approximate) representation of this using the n_m available modes.

If we suppose that the parameter vectors for our original set of shapes are drawn from some probability distribution $p(\mathbf{b})$, then we can sample parameter vectors $\mathbf{b}$ from this distribution. We can then construct the corresponding shapes for each parameter vector $\mathbf{b}$ as above (2.38). This gives us an arbitrarily large set of generated shapes, sharing the same distribution as the original set. This is usually referred to as applying the SSM in a generative mode.

The remaining task is to learn this distribution $p(\mathbf{b})$, given our original set of shapes – in this context, we refer to this set as a *training set*.

2.2 Modelling Distributions of Sets of Shapes

For a simple unimodal distribution of shapes in shape space, PCA generates a coordinate system centred on the distribution, whose axes are aligned with the significant directions of the distribution, and represent modes of variation of that data. If the distribution is not simple, PCA will still enable us to discard dimensions which are orthogonal to the data, that is, perform dimensional reduction. The individual directions $\mathbf{n}^{(a)}$ will not however necessarily correspond to modes of variation of the data.

In the following sections, we consider various methods for studying and representing the distribution of the training data in shape space. We start with the simplest case of a single multivariate Gaussian, where the data is unimodal and the PCA axes do correspond to real modes of variation of the input data. For the case of multimodal or non-linear data distributions, we discuss two types of kernel methods, the classical method of kernel density estimation, and the more recent technique of kernel principal component analysis.

2.2.1 Gaussian Models

We will consider modelling the distribution of the data by a multivariate Gaussian. Having already applied PCA, we now model the parameter space containing the vectors $\{\mathbf{b}^{(i)}\}$ defined above (2.34).

We consider a multivariate Gaussian distribution centred on the origin in parameter space, that is, centred on the mean of the data in shape space.

Theorem 2.2. Maximum Likelihood Method.
Consider a centred Gaussian probability density function (pdf) of the form:

$$p(\mathbf{b}) \propto \left(\prod_{c=1}^{n_m} \frac{1}{\sigma_c}\right) \exp\left(-\frac{1}{2}\sum_{a=1}^{n_m}\left(\frac{\mathbf{b}\cdot\mathbf{m}^{(a)}}{\sigma_a}\right)^2\right), \tag{2.39}$$

where $\{\mathbf{m}^{(a)} : a = 1, \ldots n_m\}$ *are some orthonormal set of directions:*

$$\mathbf{m}^{(a)} \cdot \mathbf{m}^{(b)} = \delta_{ab}, \tag{2.40}$$

and $\{\sigma_a\}$ *are the set of width parameters. The fitted Gaussian which maximises the quantity:*

$$\prod_{i=1}^{n_S} p(\mathbf{b}^{(i)}), \tag{2.41}$$

is then given by $\{\mathbf{m}^{(a)}\}$ *equal to the eigenvectors of the covariance matrix of* $\{\mathbf{b}^{(i)}\}$*. If these eigenvectors have corresponding eigenvalues* $\{\lambda_a\}$*, then the optimum width parameters are:*

$$\sigma_a^2 = \frac{1}{n_S}\lambda_a. \tag{2.42}$$

Proof. We are required to maximise:

$$\prod_{i=1}^{n_S} p(\mathbf{b}^{(i)}). \tag{2.43}$$

Equivalently, we can maximise instead the logarithm of this:

$$\mathcal{L} = -n_S \sum_{c=1}^{n_m} \ln \sigma_c - \frac{1}{2}\sum_{i=1}^{n_S}\sum_{a=1}^{n_m}\left(\frac{\mathbf{b}^{(i)}\cdot\mathbf{m}^{(a)}}{\sigma_a}\right)^2 + \textit{(constant terms)}, \tag{2.44}$$

with the orthonormality constraints as above. For the case of the directions $\{\mathbf{m}^{(a)}\}$*, if we compare this to (2.20), we see that it is essentially the same optimisation problem as the one we encountered previously. Hence we can deduce that the directions* $\{\mathbf{m}_a\}$ *are just the eigenvectors of the covariance*

matrix of the $\{\mathbf{b}^{(i)}\}$*. And since this covariance matrix is diagonal in the PCA coordinate frame (2.35), we finally have that* $\mathbf{m}^{(a)} = \mathbf{n}^{(a)} \ \forall \ a = 1, \ldots n_m$*.*

For the parameters $\{\sigma_a\}$*, we then have to optimise:*

$$\mathcal{L} = -n_S \sum_{c=1}^{n_m} \ln \sigma_c - \frac{1}{2} \sum_{a=1}^{n_m} \frac{\lambda_a}{\sigma_a^2} + \textit{(constant terms)}, \tag{2.45}$$

$$\implies \frac{\partial \mathcal{L}}{\partial \sigma_a} = -\frac{n_S}{\sigma_a} + \frac{\lambda_a}{\sigma_a^3}, \tag{2.46}$$

$$\therefore \ \frac{\partial \mathcal{L}}{\partial \sigma_a} = 0 \implies \boxed{\sigma_a^2 = \frac{1}{n_S}\lambda_a = \frac{1}{n_S} \sum_{i=1}^{n_S} \left((\mathbf{x}_i - \bar{\mathbf{x}}) \cdot \mathbf{n}^{(a)} \right)^2}, \tag{2.47}$$

which is just the mean variance across the set of shapes in the direction $\mathbf{n}^{(a)}$*.* □

In many cases, where the shape variation is linear, a multivariate Gaussian density model is sufficient. A single Gaussian cannot however adequately represent cases where there is significant non-linear shape variation, such as that generated when parts of an object rotate, or where there are changes to the viewing angle in a two-dimensional representation of a three-dimensional object. The case of rotating parts of an object can be dealt with by using polar coordinates for these parts, rather than the Cartesian coordinates considered previously [87]. However, such techniques do not deal with the case where the probability distribution is actually multimodal, and in these cases, more general probability distribution modelling techniques must be used. In what follows, we consider kernel-based techniques, the first being classical kernel density estimation, and the second based on the technique of kernel principal component analysis.

2.2.2 Kernel Density Estimation

As before, we start from the set of n_S centred points $\{\mathbf{b}^{(i)}\}$ in shape space $\mathbb{R}^{n_m}$. Kernel density estimation [165] estimates a pdf from data points by essentially smearing out the effect of each data point, by means of a kernel K:

$$p(\mathbf{b}) = \frac{1}{n_S h^{n_m}} \sum_{i=1}^{n_S} K\left(\frac{\mathbf{b} - \mathbf{b}^{(i)}}{h} \right), \tag{2.48}$$

where h is a scaling parameter. In the trivial case where the kernel K is a Dirac δ-function, we obtain the empirical distribution of the data, a pdf $p(\mathbf{b})$ which is zero everywhere except at a data point. A non-trivial choice of kernel would be a multivariate Gaussian:

$$K(\mathbf{b}) \doteq \mathcal{N}(\mathbf{b}; \mathbf{0}, \mathbf{D}), \tag{2.49}$$

where the covariance $\mathbf{D}$ of the kernel can be chosen to match the covariance of the data $\{\mathbf{b}^{(i)}\}$.

A slightly more sophisticated approach is the sample smoothing estimator [15, 175]. Rather than a single global scale parameter h, there is now a local scale parameter, which reflects the local density about each data point, allowing wider kernels in areas where data points are sparse, and narrower kernels in more densely populated areas. Similarly, the kernel covariance can also vary locally [152].

Such kernel methods can give good estimates of the shape distribution. However, the large number of kernels can make them too computationally expensive in an application such as the Active Shape Model (ASM) (Sect. 2.4.1). Cootes et al. [35, 36] developed a method of approximating the full kernel density estimate using a smaller number of Gaussians within a Gaussian mixture model:

$$p_{mix}(\mathbf{b}) \doteq \sum_{i=1}^{n_{mix}} w_i \mathcal{N}(\mathbf{b}; \boldsymbol{\mu}_i, \mathbf{D}_i), \tag{2.50}$$

where n_{mix} is the number of Gaussians within the mixture model, w_i is the weight of the i^{th} Gaussian, with center $\boldsymbol{\mu}_i$ and covariance $\mathbf{D}_i$. The fitting of the parameters can be achieved using a modification [36] to the standard Expectation Maximisation (EM) algorithm method [117].

2.2.3 Kernel Principal Component Analysis

The previous method aims to fit a non-linear or multimodal shape distribution by constructing a parametric non-linear and multimodal distribution within the original shape space.

The Kernel Principal Component Analysis (KPCA) method takes a different approach. KPCA [156, 157] is a technique for non-linear feature extraction, closely related to methods applied in Support Vector Machines [194, 188] (SVMs). Rather than working within the original data space with non-linear and multimodal distributions, KPCA seeks to construct a non-linear mapping of input space $\mathcal{I}$ to a new feature space.

Let $\mathbf{b}$ represent a point in our input data space[3] $\mathcal{I} = \mathbb{R}^{n_m}$, which is mapped to a feature space $\mathcal{F}$:

$$\boldsymbol{\Phi} : \mathbb{R}^{n_m} \mapsto \mathcal{F}, \ \mathbb{R}^{n_m} \ni \mathbf{b} \mapsto \boldsymbol{\Phi}(\mathbf{b}) \in \mathcal{F}, \tag{2.51}$$

[3] Here, we start from the dimensionally reduced space $\mathbb{R}^{n_m}$ rather than the original shape space $\mathbb{R}^{dn_P}$ in order to also include the infinite-dimensional case $n_P \mapsto \infty$ that is considered in Sect. 2.3.