Learning Shape: Optimal Models for Analysing Natural Variability

Rhodri Huw Davies

Division of Imaging Science and Biomedical Engineering,
University of Manchester

Contents

Abstract

Statistical shape models have wide application in biomedical image analysis – both for image segmentation and morphometry. This thesis addresses an important issue in shape modelling, that of establishing correspondence between a set of shapes. Current methods involve either manual annotation of the data (the current 'gold standard') or establishing correspondences in an essentially arbitrary manner. The thesis establishes a principled framework for establishing correspondences completely automatically by treating this as part of the learning process. Ideas from information theory are used to develop an objective function that measures the utility of a model, based on the minimum description length principle. Model-building can then be posed as the problem of finding the set of correspondences that optimise the objective function. Efficient methods are presented for manipulating correspondences via re-parameterisation and for optimising the objective function. Practical results are presented for both 2D and 3D training sets of shapes from medical images. A quantitative evaluation shows that the resulting models have better compactness, generalisation ability and specificity than those obtained using existing methods. A 3D model is used in a practical application to explore the possibility of using 3D magnetic resonance images to detect differences in shape between the hippocampi of schizophrenic patients and normal controls. A more significant effect is demonstrated using the new method than that obtained using the best previous approach.

Copyright

Acknowledgements

I would like to thank Prof Chris Taylor OBE for being such an outstanding supervisor over the course of the project; he was an invaluable source of ideas, support, and guidance. My "mathematical mentor" Dr Carole Twining also made a significant contribution to the work in this thesis – especially in the later formulation of the objective function. Dr Tim Cootes deserves thanks for his critical role in the original derivation of the objective function.
My industrial supervisor Dr John Waterton gave priceless advice and ensured that my interests remained broad. Alan Brett, Danny Allen and Neil Thacker contributed to some of the ideas presented here. I would also like to express appreciation to other members of ISBE for making my stay there such an enjoyable experience.
The hippocampus data and SPHARM model were provided by Guido Gerig, Martin Styner and co-workers from the University of North Carolina. The hippocampus study was sponsored by the Stanley Foundation. Adrian Barret and co-workers from Imperial College supplied the segmented prostate data.
The project was funded by a BBSRC Industrial CASE studentship with AstraZeneca.
Finally, I would like to thank my friends, my family, and Edwina.

"Cell and tissue, shell and bone, leaf and flower, are so many portions of matter, and it is in obedience to the laws of physics that their particles have been moved, moulded and conformed. They are no exceptions to the rule that God always geometrizes. Their problems of form are in the first instance mathematical problems, their problems of growth are essentially physical problems, and the morphologist is, *ipso facto*, a student of physical science."
– D'Arcy Thompson [102].

Chapter 1

Introduction

"An ounce of action is worth a ton of theory."
– *Friedrich Engels.*

This thesis describes a principled approach to constructing 'optimal' models of shape variation. The work is motivated by applications in image segmentation and morphological analysis. Although the approach is generic, the thesis concentrates on biomedical applications – first in 2D, then in 3D. The aim is to show that, by treating model construction as an optimisation problem, the process can be automated and the effectiveness of the resulting models guaranteed.

1.1 Model-based Image Segmentation

Segmentation is a fundamental step in computer-based image interpretation. It involves labelling an image so that the labels correspond to real world objects. For example, figure 1.1 shows a magnetic resonance image (MRI) of a brain that has been segmented to label some of the subcortical structures.

Segmentation can be performed manually by tracing the outline of the object of interest, but this requires expert knowledge and reproducible results are difficult to achieve. Manual segmentation is also a time-consuming and tedious process that is prone to operator bias and fatigue.

Low-level methods, such as edge detection and region growing (see [93] for a review), are often used on their own or to aid manual segmentation. These can provide plausible segmentations of simple images but the presence of noise, occlusion and structural complexity may lead to incomplete segmentation and/or

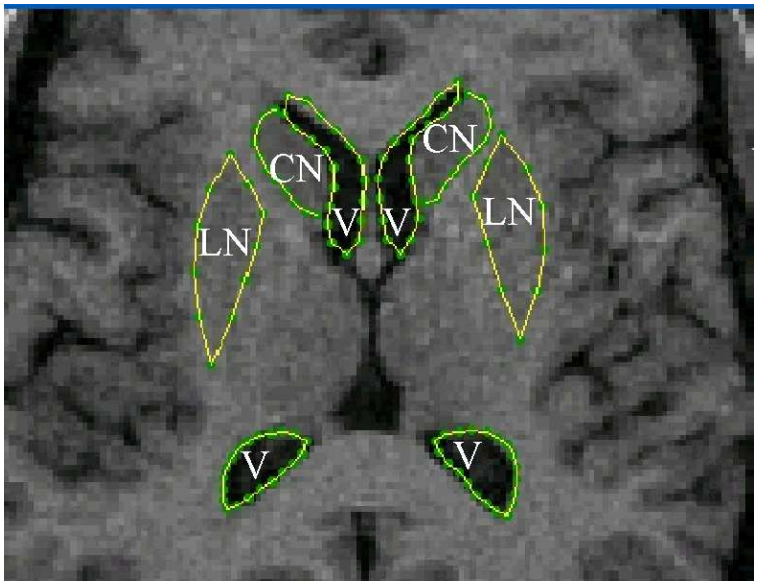


Figure 1.1. A magnetic resonance image (MRI) of a brain.
Segmentation key: CN – Caudate Nucleus; LN – Lentiform Nucleus; V – Ventricle.

erroneous labelling. Intensive user intervention is often required to correct the resulting segmentation.

A promising approach to segmentation - and that followed in this thesis – is to incorporate *a priori* knowledge in a *model* of the class of objects of interest, encoding high-level information such as expected size, position, shape and appearance. This information allows examples of the object to be segmented even in the face of noise, occlusion and structural complexity.

1.2 Models for Morphological Analysis

Morphology is the study of biological form and structure. There are many examples where a connection has been suggested between abnormal structure and function (e.g. the shape of the hippocampus in schizophrenia [69]). Morphological analysis can give a quantitative description of shape differences between the anatomy of normal and affected patients providing invaluable insight into the disease process and a basis for diagnosis. For example, figure 1.2 shows the result of a morphological analysis of the hippocampal shape of healthy and schizophrenic patients. The figure shows the shape variation that discriminates

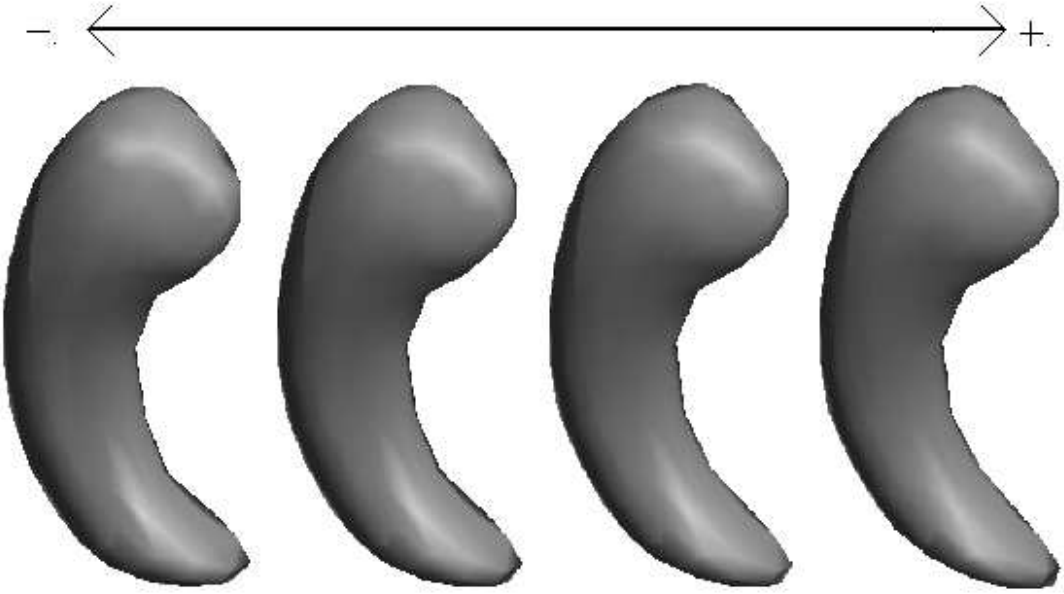


Figure 1.2. The shape variation that discriminates between the hippocampal shape of a group of healthy (−) and schizophrenic (+) patients. The main effect is a lengthening and narrowing of the 'tail'.

between the hippocampi of a group of healthy and schizophrenic patients[1].

Although many morphological tools exist (see [32] for an overview), apart from some recent advances (e.g. [7] and pp 307–311 of [32]), they are mainly applicable to sets of 'landmark points' manually annotated on 'important features'. We wish to extend this work by analysing the shape of the *entire* object as this should provide more complete information. Although this is a relatively new area of research, promising results have already been reported (e.g. [40, 41]). Closely related is deformation-based morphometry which analyses 'warp' fields between different shape instances [45, 103]. Also related is voxel-based morphometry, which analyses the spatial occupancy of different tissues [2].

1.3 Approach to Constructing Models

The modelling approach followed in this thesis involves 'learning' the shape variation of a class of objects from a set of training examples. If the model is constructed correctly, it can generalise to unseen instances of the class. A key issue in constructing a shape model is establishing the correct correspondence

[1]The method used to calculate this shape difference is described in detail in chapter 13.

between different instances of the class of shapes. In previous work, correspondence has often been established using manual 'landmarks', but this is a tedious, time-consuming and error-prone process that becomes impractical for 3D objects. Although several approaches have been proposed for automatic model-building, the resulting correspondences are essentially arbitrary. This thesis describes an approach to model-building based on the idea of selecting the set of correspondences that build the 'best' model. In other words, model-building is posed as an optimisation problem in which the correspondences are manipulated so as to optimise an objective function. This involves the following components:

A method of manipulating correspondence : based on re-parameterising each training example; this allows arbitrary patterns of correspondence whilst ensuring that the resulting model is valid.

An objective function : based on the minimum description length (MDL) principle [82], that directly promotes desirable properties of the model.

A method of optimisation : to build the 'best' model by finding the set of re-parameterisations of the training shapes that minimises the objective function.

The proposed method is evaluated on synthetic and biomedical objects, showing that it produces models with substantially better specificity, generalisation ability and compactness than models built using alternative methods.

To illustrate its use, the approach is applied to modelling the hippocampus in a set of control and schizophrenic patients. Discriminant analysis is applied and the MDL model is shown to capture more significant shape differences than a model built using a widely recognised alternative approach.

1.4 Thesis Organisation

The remainder of the thesis is organised as follows:

Chapter 2 reviews methods of constructing deformable models of variable objects.

Chapter 3 describes how a statistical shape model can be constructed from a training set of shapes.

Chapter 4 demonstrates the importance of choosing an appropriate set of correspondences and reviews previous attempts to define correspondences for a set of shapes automatically.

Chapter 5 outlines the chosen model-building approach and gives an overview of the work in chapters 6 to 9.

Chapter 6 reviews previous work on measuring model 'quality' and gives a detailed derivation of the MDL objective function.

Chapter 7 defines a piecewise linear representation of re-parameterisation and shows how the set of training shapes can be re-parameterised to optimise the MDL objective function.

Chapter 8 describes an improved representation of re-parameterisation and shows how it can be employed in a multiresolution scheme to optimise the objective function.

Chapter 9 investigates various approaches to improving the optimisation process, leading to better shape models.

Chapter 10 presents the results of applying the proposed method to a number of 2D sets of training data.

Chapter 11 shows how the proposed method can be extended to build 3D shape models by re-parameterising sets of closed surfaces.

Chapter 12 presents the results of applying the 3D method to several sets of training data.

Chapter 13 demonstrates the value of the method in a practical application, comparing shape discrimination results with those obtained using a competing method of model-building.

Chapter 14 draws conclusions, presents ideas for future work and summarises the contribution of the thesis.

Chapter 2

Deformable Models

"The purpose of models is not to fit the data but to sharpen the questions."
– Samuel Karlin.

Deformable models have been used extensively as a basis for segmenting medical images and performing morphological analysis. This chapter reviews the most important approaches and compares their properties to those of an ideal model. A more detailed (but rather dated) survey can be found in [71].

2.1 Ideal Properties of a Model

The utility of a model depends on its ability to characterise the class of objects modelled. Ideally, a model should be:

- **General** – able to represent *any* instance of the class.
- **Specific** – *only* capable of representing legal instances of the class.
- **Compact** – with as few parameters as possible.

Of these, specificity is particularly important, especially for segmentation where it determines the extent to which prior knowledge of expected structure can be applied to achieve robust results. From a practical point of view, it is also important that the approach selected should be computationally tractable using a desktop computer.

2.2 Snakes

The use of deformable models in image segmentation was popularised by Kass *et al.* [54]. They describe an active contour model, or 'snake', that deforms inside an image. The snake's deformation is governed by a combination of forces: an image force, an internal force and an external force. The image force is minimised when the snake matches some image evidence (e.g. edges); the internal forces ensure that the snake maintains a smooth shape; and the external forces can be controlled by a human operator to attract the snake to particular image features. The snake converges when the forces achieve equilibrium.

Although many enhancements have been proposed to the method (e.g. adding a 'balloon' force [18] and allowing topological changes [70]), a snake lacks specificity as it has no *a priori* knowledge of the domain. It can consequently deform to match *any* smooth contour. The method is of limited use in morphology since it provides no convenient parameterisation; it is best suited to aid in manual segmentation.

2.3 Object-specific Models

Many deformable models are custom built for the task at hand (e.g. [110], [64], [47] and [68]) . Yuille *et al* [110], for example, build a model of an eye for video image segmentation. Object-specific parameters, such as the radius of the iris, are used to deform the model. Although this approach works for structures with obvious modes of deformation, many classes of objects cannot be parameterised so simply. The expense of hand-crafting a new model for each class of object is also overly restrictive.

2.4 Physical Models

Physical models incorporate *a priori* knowledge about the expected 'physical' variation of an object. A single example is analysed to construct a parametric shape basis.

Pentland and Sclaroff [76] and Nastar and Ayache [75] describe a method of building shape models from a prototype represented by a set of nodes attached to each other by springs. The mass of each node and the stiffness of each spring are specified by two matrices. These matrices are used to solve a generalised eigenproblem to obtain the 'modes of vibration' of the structure.

Terzopulos and Metaxas [101] describe 'deformable superquadric' models that change shape according to physical properties. Deformable superquadrics combine the global shape properties of a superquadric ellipsoid with the local shape properties of a spline. The superquadric ellipsoid has six global variables: one scaling parameter, three parameters to define the aspect ratios and two parameters that control the model's 'squareness'. The spline represents free form deformations to accommodate for local changes that cannot be adequately described by the superquadric ellipsoid. The models are used in segmentation by deforming the model to match image evidence (e.g. edges) in a manner constrained by physical properties.

Although physical modelling techniques seem intuitive and exploit a convenient mathematical framework, the resulting models are unlikely to provide an accurate representation of *true* shape variation leading to non-specific models that can represent invalid instances of the class of objects.

2.5 Statistical Shape Models

Statistical models attempt to capture the actual patterns of variability found in a class of objects, rather than making arbitrary assumptions. The basic idea is to estimate the population statistics from a *set* of examples instead of using a single prototype.

Staib and Duncan [95] use a Fourier decomposition to form an orthogonal shape basis for a training set of closed contours. Normalisation is performed to achieve invariance to similarity transformations and starting point. The Fourier coefficients are recorded over the training set and modelled using a set of distributions. New examples are generated by sampling independently from the distributions and reconstructing the shape. In practice, different coefficients are often correlated over the training set, so independent sampling from the individual distributions can lead to illegal examples – the model is non-specific. Also, there may be important information in the boundary over a large range of frequencies, leading to a non-compact model.

Cootes *et al.* [24] construct 'point distribution models' (PDMs) from training sets of 2D boundaries. A discrete number of points are sampled on each boundary and their coordinates are concatenated to form a shape vector. Corresponding points are chosen to lie at equivalent positions on different instances of the shape. By registering the shapes to a common co-ordinate frame, the variation can be represented by a multivariate Gaussian model. Principal component analysis is used to define a coordinate frame aligned to the principal axes of the data. New shapes are generated by sampling independently from the distribution along each axis (these are, by definition, linearly independent) and recon-

structing using the principal vectors. Thus the models are specific. As the positions of the points tend to be highly correlated, and only components with significant variance need be retained, the resulting models can be quite compact. More details are given in chapter 3.

2.6 Deformable Atlases

An alternative approach to modelling shape boundaries is to consider the space in which they are embedded. Models constructed from this representation can be regarded as 'atlases' that deform to match a target image. By labelling (i.e. segmenting) the atlas, the labels can be transferred onto the target image after matching.

Bajcsy and Kovacic [3] propose a physics-based deformable atlas. A multiresolution approach is used to deform the atlas to match a target image. The model is deformed according to elastic properties ensuring that the atlas does not 'tear'. Christensen *et al.* [15] describe a similar framework where a two-step process is used to register a source image (the atlas) onto a target image. First, the atlas deforms according to elastic properties. This is followed by a non-linear deformation, based on the properties of a viscous fluid that accommodates for finer deformations. The application of these physics-based models is limited to images that are similar to the prototype. Like the physical boundary and surface models, the deformations allowed by such models are unlikely to reflect the true variation in the class of objects.

Several atlas-based methods have been described specifically for face images, but are equally applicable to medical images. Sirovich and Kirby [91] describe a statistical approach to atlas building. 'Eigenfaces' are constructed by performing a statistical analysis on a set of rigidly registered face images. A model is constructed by performing principal component analysis on the spatially-corresponding intensity values of a training set of images. Because rigid registration does not generally align individual features, deformation is treated as a spatially varying intensity change, leading to non-specific models. Craw and Cameron [26] establish spatial correspondence by warping each training image into a shape-normalised frame. Principal component analysis is performed on these shape-normalised images to obtain a model of grey-level variation. Edwards *et al.* [34] build a combined appearance model that describes the correlated variation of shape and intensity. A grey-level model is constructed from a set of annotated images using the method of Craw and Cameron and combined with a point distribution model (see above) of the annotated points. A further statistical analysis is performed to eliminate any correlation between the shape and grey-levels parameters.

Although atlas-based models can sometimes be used to achieve robust segmentation [20], they are sensitive to imaging modality and image acquisition parameters. A shape boundary model, on the other hand, can be used with any medical imaging modality. For example, a model could be trained on a set of segmented magnetic resonance images and used to segment ultrasound images.

2.7 Conclusions

Although there are pros and cons for all the deformable model approaches outlined above, the point distribution models of Cootes *et al.* [24] satisfy the criteria set out in section 2.1 (general, specific, compact) most closely and can be used in both boundary-only [21] and deformable atlas representations [34]. Efficient algorithms already exist to apply these models in image segmentation [25, 19].

PDMs are, however, difficult to construct. The problem of corresponding a large number of 'equivalent' points on all the examples in a training set is ill-defined and often solved, in practice, by manual annotation. This is subjective and labour intensive for 2D images and impractical in 3D. The main contribution of this thesis is to solve the problem of constructing models automatically from a training set of pre-segmented contours (in 2D) or surfaces (in 3D).

Chapter 3

Learning Shape

"Science is always wrong. It never solves a problem without creating ten more"
– George Bernard Shaw.

The automated approach to learning shape developed in this thesis builds on the statistical shape models of Cootes *et al.* [24]. This chapter outlines their approach, which is based on representing each shape by a set of boundary points, and presents an illustrative example of building a model from a set of 2D training shapes. The method is extended to deal with continuous boundaries in 2D and surfaces in 3D, allowing it to work directly from the training data. The importance of choosing appropriate correspondences between the members of the training set is discussed and demonstrated using the illustrative example.

3.1 Constructing Statistical Shape Models

3.1.1 Representing Shapes

A statistical shape model is constructed from a training set of n_s shapes, aligned to a common coordinate frame. Each shape, S_i ($i = 1, \ldots n_s$), can – without loss of generality – be represented by a dense set of n points, sampled on each example boundary. The way these points are sampled is important and is discussed later in this chapter (§3.2). By concatenating the (Cartesian) coordinates of the sample points, each example can be represented by an n_p-dimensional shape vector $\mathbf{x}_i$. For example, in 2D:

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{in}, y_{i1}, y_{i2}, \ldots, y_{in})^T, \tag{3.1}$$

and in 3D:

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{in}, y_{i1}, y_{i2}, \ldots, y_{in}, z_{i1}, z_{i2}, \ldots, z_{in})^T. \tag{3.2}$$

3.1.2 Statistical Analysis

Following the approach of Cootes *et al.* [24], the set of shape vectors can be modelled by a multivariate Gaussian distribution[1]. Principal Component Analysis (PCA) is performed to define a set of axes that are aligned with the principal directions of the data. The approach can be summarised as follows:

1. calculate the mean shape:
$$\bar{\mathbf{x}} = \frac{1}{n_s} \sum_{i=1}^{n_s} \mathbf{x}_i; \tag{3.3}$$

2. calculate the normalised covariance matrix:
$$\mathbf{D} = \frac{1}{n_s n_p} \sum_{i=1}^{n_s} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T; \tag{3.4}$$

3. solve for the eigenvalues $\{\lambda^m\}$ and unit eigenvectors $\mathbf{P} = \{\mathbf{p}^m\}$ of $\mathbf{D}$:
$$\mathbf{D}\mathbf{p}^m = \lambda^m \mathbf{p}^m. \tag{3.5}$$

The eigenvalues (and corresponding eigenvectors) are sorted in descending order and scaled such that:

$$\lambda^m \rightarrow n_p \lambda^m. \tag{3.6}$$

λ^m now gives the variance of the training data in the m^{th} principal direction, $\mathbf{p}^m$.

[1] If there is an intrinsic non-linearity in the variation of shape, a more complex model may be required [23] but for the remainder of this thesis, we will assume that a single multivariate Gaussian is sufficient to model the data.

3.1.3 Modelling Shape

Following the statistical analysis described above, a shape vector $\mathbf{x}_i$ can be described by a linear model of the form:

$$\mathbf{x}_i = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b}_i = \bar{\mathbf{x}} + \sum_m \mathbf{p}^m b_i^m, \tag{3.7}$$

where the unit eigenvectors $\{\mathbf{p}^m\}$ represent a set of axes that define the new co-ordinate system and $\mathbf{b} = \{b^m\}$ are shape parameters. A change in shape (often referred to as a 'mode' of shape variation) can be induced by independently varying one of the shape parameters.

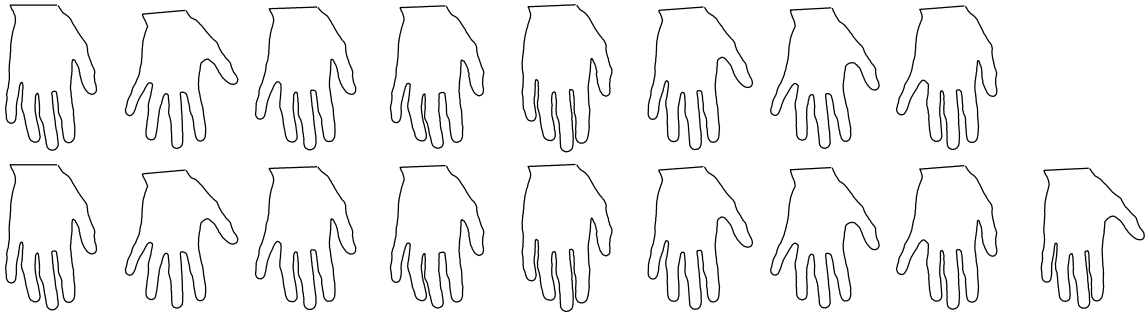


Figure 3.1. A training set of hand outlines.

New examples from the class of shapes can be generated by selecting different values of $\{b^m\}$. One can ensure that the model can only represent 'valid' instances by constraining the values of $\{b^m\}$ to be similar to those found in the training set (e.g. $b^m = \pm 3\sqrt{\lambda^m}$). For example, figure 3.3 shows a model constructed from the training set of hand outlines shown in figure 3.1. The figure shows the first three modes of variation by independently varying the values of the first three shape parameters, $(b^m, m = 1 \ldots 3)$ in (3.7), by $\pm 2\sqrt{\lambda^m}$. The boundaries were sampled manually by placing a number of 'landmarks' on the salient features of each shape (such as the tips of the fingers) as shown in figure 3.2. A fixed number of points were equally-spaced in between the landmarks resulting in $n = 400$ sample points. The examples were aligned into a common co-ordinate frame by minimising the squared distance between corresponding points (as described in [24]). The re-aligned points were concatenated into a shape vector and the model was built as described above. It can be seen in figure 3.3 that the model captures considerable variability (is general) but only reconstructs 'legal' examples of hands (is specific). Five parameters are required to capture 96% of the variation of the training set (is compact).

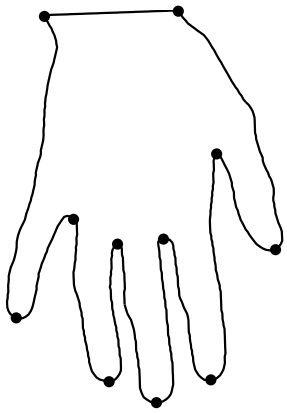


Figure 3.2. How manual landmarks were placed on the training boundaries.

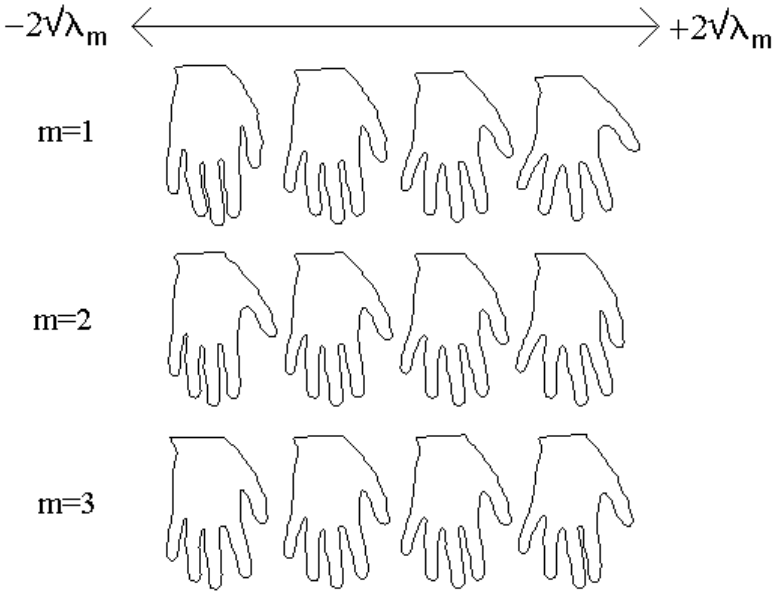


Figure 3.3. The first three modes of variation of the manually constructed hand outline model. Each parameter $(b_m, m = 1 \ldots 3)$ is varied independently by $\pm 2\sqrt{\lambda_m}$.

3.1.4 Modelling Continuous Shapes

In this work, we wish to construct shape models *directly* from the training data which are in the form of continuous boundaries or surfaces. Thus we are interested in the limit $n_p \to \infty$, which leads to an infinitely large covariance matrix. We note, however, that there can be at most $n_s - 1$ non-zero eigenvalues hence the index m in (3.7) has an upper limit of $n_s - 1$.

If we construct an $n_p \times n_s$ data matrix $\mathbf{W}$ from the set of vectors $\{(\mathbf{x}_i - \bar{\mathbf{x}}) : i = 1, \ldots n_s\}$; the $n_p \times n_p$ covariance matrix in (3.4) can be written as $\mathbf{D} = \frac{1}{n_p n_s}\mathbf{W}\mathbf{W}^T$ with eigenvectors and eigenvalues $\{\mathbf{p}^m, \lambda^m\}$:

$$\mathbf{D}\mathbf{p}^m = \lambda^m \mathbf{p}^m. \tag{3.8}$$

If we define the $n_s \times n_s$ matrix, $\mathbf{D}' = \frac{1}{n_p n_s}\mathbf{W}^T\mathbf{W}$ with eigenvalues and eigenvectors $\{\mathbf{p}'^m, \lambda'^m, \}$ then:

$$\begin{aligned} \mathbf{D}'\mathbf{p}'^m &= \lambda'^m \mathbf{p}'^m, \\ \text{From (3.8)}: \ \mathbf{D}\mathbf{p}^m &= \lambda^m \mathbf{p}^m, \\ \Rightarrow \frac{1}{n_p n_s}\mathbf{W}\mathbf{W}^{\mathbf{T}}\mathbf{p}^m &= \lambda^m \mathbf{p}^m, \end{aligned} \tag{3.9}$$

pre-multiplying by $\mathbf{W}^{\mathbf{T}}$ gives:

$$\begin{aligned} \Rightarrow \ \mathbf{D}'(\mathbf{W}^{\mathbf{T}}\mathbf{p}^m) &= \lambda^m(\mathbf{W}^{\mathbf{T}}\mathbf{p}^m), \\ \text{Similarly:} \quad \mathbf{D}(\mathbf{W}\mathbf{p}'^m) &= \lambda'^m(\mathbf{W}\mathbf{p}'^m). \end{aligned} \tag{3.10}$$

Therefore, for all $\lambda^m \neq 0$, we can assign indices such that:

$$\lambda^m = \lambda'^m \ \text{ and } \ \mathbf{p}^m = \mathbf{W}\mathbf{p}'^m. \tag{3.11}$$

Hence, all $n_s - 1$ eigenvalues of $\mathbf{D}$ can be obtained directly from $\mathbf{D}'$, and the eigenvectors are a weighted sum of the training shapes. It is clear how $\mathbf{D}'$ can be calculated when n_p is very large, but not in the limit that $\{S\}$ is continuous ($n_p \to \infty$).

We choose to represent the training examples parametrically[2]:

$$S = S(\mathbf{u}) = (S^x(\mathbf{u}), S^y(\mathbf{u}), S^z(\mathbf{u}))^T \tag{3.12}$$

where $\mathbf{u}$ is a parameterisation of the shape ($\mathbf{u} = u$ in 2D and $\mathbf{u} = (u, v)$ in 3D) and $S^x(\mathbf{u}), S^y(\mathbf{u}), S^z(\mathbf{u})$ are the Cartesian coordinates of the boundary/surface at $\mathbf{u}$. The mean of a set of shapes, $\bar{S}(\mathbf{u}) = \frac{1}{n_s}\sum_i S_i(\mathbf{u})$, is also defined. It can be shown [58] that $S(\mathbf{u})$ forms an infinite dimensional vector space.

The summation over all the boundary/surface points, implicit in (3.9) now becomes an integral and the ij^{th} element of $\mathbf{D}'$ can be calculated from:

[2]We describe how this can be achieved practically in later chapters.

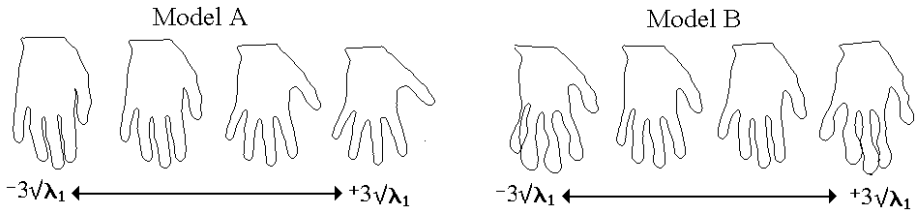


Figure 3.4. The first mode of variation of models A and B. The first parameter (b_1) is varied by $\pm 3\sqrt{\lambda_m}$.

$$\mathbf{D}'_{ij} = \int d\mathbf{u}(S_i(\mathbf{u}) - \bar{S}(\mathbf{u}))^T(S_j(\mathbf{u}) - \bar{S}(\mathbf{u})), \tag{3.13}$$

which is the inner product $(S_i(\mathbf{u}) - \bar{S}(\mathbf{u})) \cdot (S_i(\mathbf{u}) - \bar{S}(\mathbf{u}))$ of shapes i and j relative to the mean. Where for shapes ψ and φ, the inner product is defined as:

$$\psi \cdot \varphi = \int d\mathbf{u}\, \psi^x(\mathbf{u})\varphi^x(\mathbf{u}) + \psi^y(\mathbf{u})\varphi^y(\mathbf{u}), \quad \text{in 2D},$$

and:

$$\psi \cdot \varphi = \int d\mathbf{u}\, \psi^x(\mathbf{u})\varphi^x(\mathbf{u}) + \psi^y(\mathbf{u})\varphi^y(\mathbf{u}) + \psi^z(\mathbf{u})\varphi^z(\mathbf{u}), \quad \text{in 3D}. \tag{3.14}$$

In practice the integrals can be calculated by numerical integration.

3.2 The Correspondence Problem

We now demonstrate the importance of selecting a suitable set of correspondences. Figure 3.4 shows two models, A and B, constructed from the same training set of 10 hand outlines (shown in figure 3.1) but with different correspondences. Model A was constructed as before using manual landmarks; model B used a single manual landmark with the remaining points equally spaced around the boundary. All the example shapes generated by model A (using values of $\{b^m\}$ within the range found in the training set) are 'legal' examples of hands, whilst model B generates implausible examples. Model B is thus a non-specific model with limited utility for imposing shape constraints. It is also less compact: the variances of the first three modes are $(2.23, 2.01, 0.49)$ and $(4.64, 2.19, 1.21)$ for models A and B respectively. This demonstrates that the problem of establishing the 'correct' correspondences is fundamental to the idea of learning shape.

Previous attempts at solving this correspondence problem are reviewed in the following chapter.

Chapter 4

Solving the Correspondence Problem

"I have not failed. I've just found 10,000 ways that won't work."
– *Thomas Edison.*

The previous chapter outlined an approach to building deformable models based on statistical learning. It was demonstrated that good results may be obtained but that this depends critically on the way that correspondences are established between members of the training set. This chapter reviews previous approaches to establishing correspondence between a set of shapes and discusses their limitations. Although application-specific solutions have been proposed (e.g. [13, 50]), we confine our attention to generic approaches. Some methods deal only with correspondences between pairs of shapes rather than the whole training set, but these can be used in a binary tree to establish a full set of correspondences [48].

4.1 Manual Landmarking

Cootes *et al.* [21] define correspondence with manual landmarks placed on points of anatomical significance as in the hand example presented in figure 3.2 (§3.1.3). Although manual landmarking often leads to acceptable results, it is a subjective, error-prone, and time-consuming process that cannot be guaranteed to produce good models. For many applications specialist knowledge is required to place the landmarks in the correct place which might be argued to be an advantage, since the specialist knowledge is captured by the model but, in practice, further complicates the process of model building. In 3D, these problems are exacerbated

and visualisation difficulties make manual landmarking an impractical approach for selecting correspondences.

4.2 Bootstrapping

'Bootstrapping' is a semi-automatic technique that uses an existing model in a data-driven search to find instances of the class of object in unseen images. The resulting segmentation is manually refined and added to the model. Although this approach can save time (when compared to manual landmarking), the resulting correspondence are arbitrary. During the early stages, the model will be under-trained and be unable to represent new examples. Although the constraints on the model can be relaxed [21, 22], this leads to a model that can represent invalid instances of the class of object.

Baumberg and Hogg [5] use a bootstrapping approach to build statistical shape models. They construct an initial model by equally-spacing B-spline control points around a set of training boundaries. The model is then used in a data-driven image search to optimise an image-based objective function. During the search, points are allowed to move normal to the curve so as to better match the image evidence but there is no mechanism that will lead to their redistribution around the boundary – the key to modifying correspondences.

4.3 Correspondence by Parameterisation

The simplest approach to defining correspondence is to select a starting point on each example and equally space a number of points on each boundary. As shown above, this will generally result in a poor model. A similar scheme is advocated by Baumberg and Hogg [4] who equally space spline control points around 2D contours.

Kelemen *et al.* describe a method of building shape models from a set of closed 3D surfaces by defining correspondence through the parameterisation of each shape [55, 56, 98]. We describe the method in some detail since it is used for comparison purposes later in the thesis. An initial parameterisation of each shape is found using the method of Brechbühler *et al.* [8, 9], which poses surface parameterisation as an optimisation problem by finding the mapping from the surface to a sphere that minimises area distortion (this is discussed in further detail in chapter 11). Using the parameterisation, each shape is represented by its expansion into a sum of spherical harmonics. The shapes are aligned so that the axes of their first spherical harmonics (which are ellipsoidal) coincide and principal

component analysis is performed on the coefficients of the expansion. Since the expansion to spherical harmonics is a linear process involving integration over the surface, the net effect is the continuous equivalent of equally spacing points over the surface (according to its parameterisation). This, as was shown earlier, tends to build poor models.

4.4 Distance-based Correspondences

A relatively straightforward automated approach to selecting correspondences is to find points that will be close to each other when the shapes have been suitably aligned.

Besl and MacKay [6] describe the iterative closest point (ICP) algorithm to register a pair of shapes and define point correspondences between them. An iterative method is used to find the correspondences and similarity transformations that minimise the squared distance between pairs of points. An initial correspondence is established by matching each point on one shape to the closest points on the other. Given these correspondences the similarity transformation is found that minimises the sum of squared distances between the shapes. The correspondences are then re-established and the process is repeated until convergence. The basic ICP method has been extended by several authors. For example, Rangarajan *et al.* [79] describe the 'softassign Procrustes matching' algorithm which deals with outliers. All these algorithms have no notion of connectivity, invalid correspondences can ensue leading to illegal reconstructions of the original shapes.

Duta *et al.* [33] use a similar approach to ICP but explicitly constrain the correspondence between shapes to be 'valid'. A clustering algorithm is used to discard shapes that are thought to be outliers. Although this leads to more aesthetic models and implicitly deals with erroneous segmentations, the excluded shape instances could be valid and should be included in the model.

Leventon *et al.* [63] and Golland *et al.* [43] use "distance maps" as shape descriptors. A distance transform is applied to a binary image of the segmentation so that the values in the resulting image give the distance of each pixel/voxel to the nearest boundary/surface point. A model is constructed by performing a statistical analysis on a rigidly aligned set of distance maps. When distance maps are combined, however, they can represent invalid instances of the modelled object leading to a non-specific model.

The general difficulty with distance-based approaches is that the relative position of equivalent points may vary considerably over the training set, invalidating proximity as a satisfactory basis for establishing correspondence.

4.5 Physically-based Correspondences

Scott and Longuett-Higgins [88] perform a singular value decomposition on a (Gaussian-weighted) proximity matrix to correspond two sets of features on different examples. Shapiro and Brady [90] extend this work by also considering the proximity of features on the *same* shape. An affinity matrix is constructed for both shapes. The eigenvectors of the affinity matrix form an orthogonal basis and capture the 'modes of vibration' of the shape. They describe a technique called 'modal matching' to define correspondences using the orthogonal bases.

Sclaroff and Pentland [87] define correspondences between pairs of shapes based on their 'physical' modes of variation. An orthogonal basis is constructed for each training shape using their earlier work on physical modelling [76] (see §2.4) and the modal matching method of Shapiro and Brady [90] is used to define correspondences from the pair of bases.

Bookstein [7] describes a method of optimising the positions of corresponding points by allowing them to slide along a 2D shape boundary. The points slide to minimise the bending energy required to place one pointset on top of the other.

Lorenz *et al.* [65] use a small number of manual landmark points to align a template to a training set of surfaces. An elastic relaxation is performed to fit the template to each training shape. The correspondence defined through the template is used to perform a statistical analysis.

All these approaches tend to be reasonably stable but the resulting correspondences are essentially arbitrary.

4.6 Shape-based Correspondence

A common approach to establishing pair-wise correspondence is to correspond points of similar local shape. Curvature-based correspondences are often employed for non-rigid contour tracking (e.g. [17]). Wang *et al.* use a curvature-based criteria to construct 3D shape models [106]. Although this ties in with human intuition, equivalent points may not in practice lie on regions of similar curvature.

Hill *et al.* [49] describe a method of defining correspondences between pairs of closed contours. The critical point detection decimation algorithm [111] is used to obtain a sparse representation of the source shape. The decimation algorithm produces a set of points that lie on regions of high curvature. An initial correspondence is established between the sparse source shape and the dense target

shape using arc-length parameterisation. An arbitrary objective function – based on the representation error of the two shapes – is employed to improve this correspondence.

Brett and Taylor [12, 11] extend the work of Hill *et al.* [49] to build shape models from sets of surfaces. The decimation algorithm of Schroder *et al.* [86] is used to obtain a sparse set of points that lie on regions of high curvature on the source shape. A variant of the ICP algorithm aligns the sparse source shape with the dense target shape and creates a sparse correspondence between them. Both surface are tessellated (to generate the same graph) using geodesic paths to establish a dense correspondence. The method, however, allows invalid correspondences between the examples. The authors overcome this in [10] by flattening the surfaces before establishing correspondence. The resulting correspondences are still completely arbitrary.

Tagare [100] guarantees valid correspondences between pairs of 2D curves by constraining each pairwise mapping (that defines the correspondence) to be bimorphic. Correspondences are found using a curvature-based objective function.

4.7 Image-based Correspondence

So far, all methods that have been reviewed find correspondences on shape boundaries (i.e. curves or surfaces). An alternative approach is to define correspondence in the space that embeds the shapes. The approach is closely related to the deformable atlases described in section 2.6. The basic idea is to deform the space of one object onto the other in order to optimise some similarity criterion between the images. A shape model can then be built from the resulting deformation field. The deformation of a specific structure can be obtained by segmenting it in one of the images and characterising the movement of the corresponding points. Any non-rigid registration algorithm can potentially be used to define a spatial correspondence, providing the topology is preserved (a comprehensive review of image registration methods can be found in [104], [44] and [66]).

Fleute and Lavallee [37] use the registration method of Szeliski and Lavallee [99] to build statistical shape models of the femur. Shapes are registered by deforming a hierarchical spline representation of the surrounding volume so as to minimise the squared distance between examples.

Frangi *et al.* [38] represent a training set of pre-segmented surfaces as binary (volume) images. Each example is iteratively registered to a reference 'atlas' by optimising a normalised measure of mutual information [97]. The registration

is performed in two stages. First, a rigid (affine) registration is performed using the method of Studholme *et al.* [96]. This is optimised using the non-rigid registration of Rueckert *et al.* [84] by manipulating a B-spline representation of the volume.

Rueckert *et al.* [83] extend this method to model unsegmented magnetic resonance images. It is not clear, however, that there is enough 'information' in a set of unsegmented images to construct *accurate* shape models.

Although non-rigid image registration methods produce plausible results, the resulting correspondences are essentially arbitrary. There are infinitely many non-rigid deformations that could match the intensities in two images and those chosen are, in effect, a side-effect of the optimisation process employed.

4.8 Finessing the Correspondence Problem

An alternative approach worthy of discussion is to bypass the correspondence problem rather than solve it. The basic idea is that all possible configurations of correspondences lie on a non-linear manifold in a high-dimensional space. A practical approach would be to consider correspondences obtained by equally spacing points, leaving the origin as a free parameter. This means that each shape is represented by a manifold (1D for 2D shapes and 2D for 3D shapes). The set of training manifolds can then be modelled using a non-linear representation (e.g. principal curves/surfaces [53] or non-linear principal component analysis [60]) .

There are, however, serious drawbacks to this approach. First is the computational complexity of constructing the non-linear model. More important is the increased number of parameters needed to estimate the non-linear model, increasing the risk, for a given size of training set, of overfitting and thus poor generalisation. In practice, linear models are of wide application, providing appropriate correspondences are chosen, and that is the approach that we choose to adopt.

4.9 Correspondence as Optimisation

A more principled approach than those considered so far is to treat correspondence as an optimisation problem, choosing correspondences to optimise an explicit objective function. This allows models with well-defined properties to be created.

Meier and Fisher [72] describe an optimisation approach to finding correspondences on pairs of shapes. They follow the work of Kelemen *et al.* [55] by representing each shape using spherical harmonics. The coefficients of the spherical harmonics of the source shape are manipulated in order to optimise a measure of similarity to corresponding points on the target shape. The similarity measure consists of an arbitrary weighting of Euclidian distance, normal directions and curvature. Although plausible results are reported, the method can generate invalid correspondences between shapes and although spherical harmonics provide a continuous, hierarchical representation, they can only define the shapes to a given accuracy.

Hill and Taylor [48] use a curvature matching dynamic programming algorithm to obtain an initial correspondence and optimise it to minimise an objective function. The objective function is the trace of the model covariance matrix plus a correction term that penalises points for moving off the original shape boundary. Correspondences are optimised by manipulating the parameters of the current model. There are two major drawbacks to this approach. First, as points can move off the boundary, illegal reconstructions of the shapes can ensue. Second, it is not entirely clear that manipulating the parameters of the current model always allows movement towards the 'optimal' solution.

Kotcheff and Taylor [59] use the determinant of the model covariance matrix as an objective function. The determinant effectively measures the volume that the training set occupies in shape space so minimising this should lead to more compact models. As we will show later (chapter 6), this objective function has degenerate minima and thus requires an arbitrary parameter to keep it well defined. This arbitrary parameter affects the convergence properties of the method. Correspondences are manipulated by explicitly re-parameterising each shape in the training set using a piecewise-linear function (this is described in further detail in chapter 7). Explicit constraints are enforced on the re-parameterisation functions to ensure that the resulting correspondences are valid. The representation of re-parameterisation cannot, however, be extended to deal with surfaces so as to build 3D statistical shape models. A genetic algorithm (GA) is used to optimise the objective function by manipulating the re-parameterisation function of each training example. Although some encouraging results are reported for small sets of simple shapes, the method takes many hours to converge and cannot cope with complex objects or large datasets.

4.10 Conclusions

We have reviewed previous work on defining correspondences between shapes. Although some of these produce plausible results, they all suffer from one or more of the following problems:

- the resulting correspondences are arbitrary;
- invalid reconstructions of the original shapes are possible;
- correspondences are considered for pairs of shapes rather than the entire training set;
- assumptions are made (e.g. a consistent direction of the principal axis) about the class of object;
- the method cannot be extended to deal with surfaces in 3D.

The method of Kotcheff and Taylor [59] comes closest to avoiding these pitfalls, but is far from an ideal solution. The objective function has an intuitive appeal, but has no rigorous justification. The genetic algorithm search is slow to converge and sometimes fails; in any case it scales poorly and is only practical for small training sets. Finally the method does not extend to surfaces in 3D, because it depends on imposing an ordering constraint around the boundary. The following chapters describe a method based on the idea of group-wise optimisation that deals with the limitations of the method of Kotcheff and Taylor.

Chapter 5

A Principled Approach to Learning Shape

"Nature uses as little as possible of anything."
– Johannes Kepler.

This chapter provides an overview of a new approach to learning shape, that integrates the problem of establishing the correct correspondences into the learning process. A general framework is established – details and alternative implementations are given in subsequent chapters. For ease of exposition, the approach is explained fully in 2D before the extension to 3D is described, although all the methods considered extend to 3D. The chapter also establishes objective criteria that can be used to compare models built in different ways and introduces training sets and benchmark models used in subsequent evaluations.

5.1 Summary of Goals

Our aim is to develop a method of solving the correspondence problem by treating it as an integrated part of the shape learning process. We seek an approach with the following properties:

- uses a principled criterion for defining correspondence;
- can guarantee that the resulting correspondences are valid;
- considers the entire training set when defining correspondence;

- can scale easily to cope with large training sets;
- extends to surfaces in 3D.

Although model-building is a one-off, off-line process, the method should also converge in reasonable time (preferably in minutes, not days).

5.2 Overview of Approach

The approach adopted is to select the set of correspondences that build the 'best' model, treating model building as an optimisation task. This requires a framework involving a method of **manipulating correspondences**, an **objective function** to assess the 'quality' of the model built from a given set of correspondences, and a method of **optimising** the objective function with respect to the set of correspondences. Each component is discussed briefly below and a roadmap is provided to the detailed treatment in subsequent chapters.

5.2.1 Manipulating Correspondence

The proposed approach requires an efficient method for manipulating correspondences, that guarantees only valid correspondences will be generated.

One possible approach would be to place n points on each of the training shapes and manipulate them directly. It is, however, difficult to ensure the legality of the resulting correspondences using this approach. For 2D shapes, legal correspondences are those that preserve the ordering of points around the boundary. If this is not respected, the resulting model will generate non-realisable shapes. The situation becomes more complicated in 3D.

We choose instead to follow Kotcheff and Taylor [59] by treating the problem of corresponding continuous curves/surfaces as one of re-parameterisation. A different re-parameterisation function $\phi_i(\mathbf{u})$ is defined for *each* shape $S_i(\mathbf{u})$ allowing points to be moved around on the boundary/surface. The idea is illustrated in figure 5.1 where a set of points are sampled from a path-length parameterisation and a re-parameterisation of a shape. For correspondences to be legal, $\phi_i(\mathbf{u})$ must be a diffeomorphic mapping – that is, ϕ_i must not cause folds or tears. In 2D, we can use the following notation:

$$S_i(\mathbf{u}) \rightarrow S_i(\mathbf{u}'), \;\; \mathbf{u}' = \phi_i(\mathbf{u}), \;\; \phi_i : [0,1] \rightarrow [0,1], \tag{5.1}$$

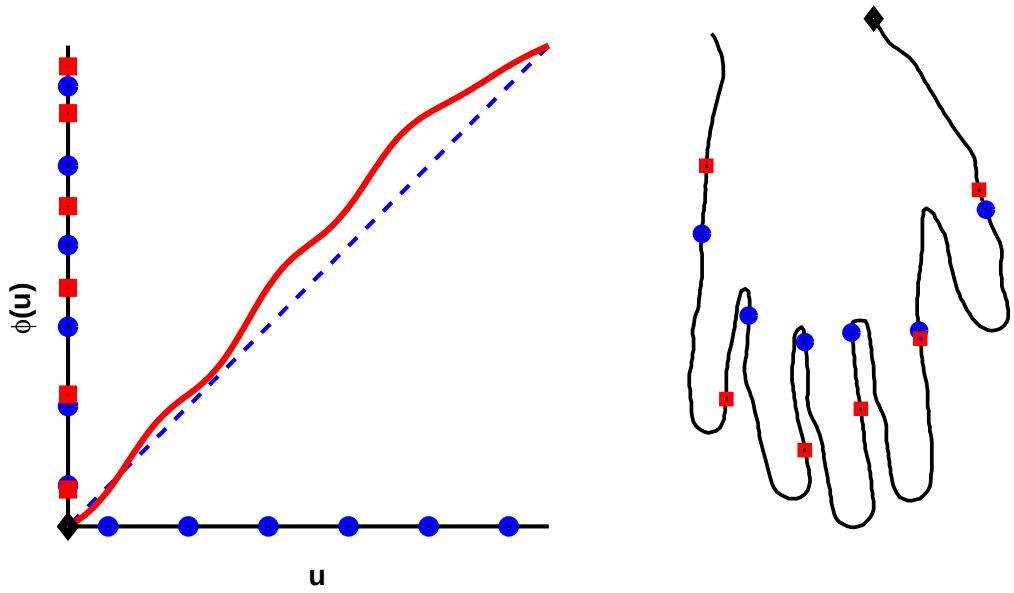


Figure 5.1. *Left*: a re-parameterisation function. The blue (dashed) line represents the original (path-length) parameterisation ($\mathbf{u}$) and the (solid) red line represents the re-parameterisation function ($\phi(\mathbf{u})$). Right: a shape outline. The blue circles are sampled according to the original parameterisation and the red squares are sampled according to the re-parameterisation function. The diamond represents the origin of the shape/parameterisation.

where ϕ is a diffeomorphism of the line (for open curves $S_i(0) \neq S_i(1)$) or the circle (for closed curves $S_i(0) = S_i(1)$) onto itself.

Two different representations of ϕ have been investigated: chapter 7 describes a recursive, piecewise-linear representation which is related to but extends the work of Kotcheff and Taylor [59]; chapter 8 describes an improved, smooth (differentiable) representation of re-parameterisation constructed from a set of kernel functions. Both representations have straightforward extensions to 3D, which are described in chapter 11.

5.2.2 Objective Function

A key assumption is that we can measure the 'quality' of a model with a suitable objective function. The approach that has been taken is based on the idea that a good model should allow a concise description of the members of the training set. As in any branch of science, this is the essential role of a model – to account for a possibly large number of observations as manifestations of some underlying

pattern which is itself described as simply as possible. This idea is formalised by using concepts from information theory to derive a minimum description length (MDL) objective function. The detailed derivation is given in chapter 6.

5.2.3 Optimisation

Several strategies are described for optimising the MDL objective function by manipulating the re-parameterisations of the training shapes. The first attempt uses a genetic algorithm to optimise the piecewise-linear representation of re-parameterisation. This is extended to allow the use of a local optimisation algorithm, leading to faster and more robust convergence. Both methods are described in chapter 7. Chapter 8 describes how the smooth representation of re-parameterisation can be optimised using a multi-resolution local optimisation approach. This last approach is chosen as the basis for a more detailed investigation of the speed, accuracy and robustness of convergence, the results of which are presented in chapter 9.

5.3 Evaluating the Methods

During the development of the methods outlined above it was important to use a consistent and objective basis for comparing different approaches. This section describes the evaluation methodology and provides 'benchmark' results for two training sets used in the comparisons.

5.3.1 Evaluation Criteria

In chapter 2 we identified the properties we would ideally require of a model: generalisation ability, compactness and specificity. The following paragraphs describe how each of these may be measured objectively. All the measures described allow meaningful comparisons between different models constructed *using the same training set*, but not across different training sets. In practice, the computational effort required to build a model is also important and is normally quoted either in terms of run-time on a certain machine (where different methods of optimising the same objective function are compared) or in terms of the number of objective function evaluations performed.

Generalisation Ability

The generalisation ability of a model measures its capability to represent unseen instances of the class of object modelled. This is a fundamental property as it allows a model to learn the characteristics of a class of object from a limited training set. If a model is overfitted to the training set, it will be unable to generalise to unseen examples.

The generalisation ability of each model is measured from the training set using leave-one-out reconstruction. A model is built using all but one member of the training set and then fitted to the excluded example. The accuracy to which the model can describe the unseen example is measured and the process is repeated excluding each example in turn. The approximation error is averaged over the complete set of trials.

Ideally, the set of correspondences used to build the models and to calculate the approximation error between the model and each excluded example should be obtained independently. Unfortunately this would make some of the experiments impractical. In practice, the correspondences obtained by considering all of the training shapes have been used. This tends to overestimate the absolute error, but allows an unbiased comparison of different models. The generalisation ability is measured as a function of the number of shape parameters, M, used in the reconstructions. More formally:

- **for** $M = 1 \dots n_s - 2$
  - **for** $i = 1 \dots n_s$,
    - build the model $(\bar{\mathbf{x}}_i, \mathbf{P}_i = \{\mathbf{p}^m : m = 1 \dots M\})$ from the training set, with $\mathbf{x}_i$ removed;
    - estimate the model parameters for shape i: $\mathbf{b}_i = \mathbf{P}_i^T(\mathbf{x}_i - \bar{\mathbf{x}}_i)$;
    - reconstruct the unseen shape using M shape parameters: $\mathbf{x}'_i(M) = \bar{\mathbf{x}}_i + \sum_{m=1}^{M} \mathbf{p}_i^m b_i^m$;
    - calculate the sum of squares approximation error: $\epsilon_i^2(M) = |\mathbf{x}_i - \mathbf{x}'_i(M)|^2$
  - calculate the mean squared error: $G(M) = \frac{1}{n_s}\sum_{i=1}^{n_s} \epsilon_i^2(M)$

If for two methods A and B, $G_A(M) \leq G_B(M)$ for all M and $G_A(M) < G_B(M)$ for some M, we can conclude that the generalisation ability of method A is better than that of method B.

In order to reason about the significance of differences we need to estimate the

likely error in $G(M)$. The standard error, of $G(M)$ derived from the sampling distribution for a mean [94] is defined as:

$$\sigma_{G(M)} = \frac{\sigma}{\sqrt{n_s - 1}}, \tag{5.2}$$

where σ is the sample standard deviation of $G(M)$.

Specificity

A specific model should only generate instances of the object class that are similar to those in the training set. It is useful to assess this qualitatively by generating a population of instances using the model and comparing them to the members of the training set. We also define a quantitative measure of specificity (again as a function of M) using:

$$S(M) = \frac{1}{N}\sum_{j=1}^{N} |\mathbf{x}_j(M) - \mathbf{x}'_j|^2, \tag{5.3}$$

where $\mathbf{x}_j$ are shape examples generated by the model (by choosing values of b in (3.7) in the range over the training set) and $\mathbf{x}'_j$ is the nearest member of the training set to $\mathbf{x}_j$. As with the measure of generalisation, we can say that model A is more specific if $S_A(M) \leq S_B(M)\text{for all } M$ and $S_A(M) < S_B(M)$ for some M. The standard error of $S(M)$ is given by:

$$\sigma_{S(M)} = \frac{\sigma}{\sqrt{N - 1}}, \tag{5.4}$$

where σ is the sample standard deviation of $S(M)$ and N is the number of samples. A value of $N = 10000$ was used to obtain the results reported in this thesis.

Compactness

A compact model is one that has as little variance as possible and requires as few parameters as possible to define an instance. This suggests that the important information is captured in a plot of cumulative variance:

$$C(M) = \sum_{m=1}^{M} \lambda^m, \tag{5.5}$$

where λ^m is the m^{th} eigenvalue (that measures the variance of the data in the m^{th} direction) and $C(M)$ is the cumulative variance of the M^{th} mode. Similar comments apply to comparing $C_A(M)$ and $C_B(M)$ for two models A and B as for generalisation and specificity.

In order to reason about the significance of differences we need to estimate the likely error in $C(M)$. The standard deviation of the sampling distribution of the variance of the m^{th} mode is given by [94]:

$$\sigma_{\lambda^m} = \sqrt{\frac{2}{n_s}}\lambda^m, \tag{5.6}$$

where λ^m is the m^{th} largest eigenvalue of the covariance matrix. Using error propagation, the standard error of $C(M)$ can be obtained from:

$$\sigma_{C(M)} = \sum_{m=1}^{M} \sqrt{\frac{2}{n_s}}\lambda^m. \tag{5.7}$$

Units of Performance Measures

$G(M)$ and $S(M)$ are derived from distances on the training shapes but the units are arbitrary because shape alignment involves scaling each shape vector to unit norm, By keeping track of these scaling factors the results can be rescaled (using the average scale factor) to units of squared pixels/voxels in the original image frame where the training data were obtained. This allows comparison with the annotation error, which is assumed to be of the order of one pixel/voxel.

5.3.2 Benchmark Models

The performance of different model building methods will be compared using a training set of 10 hand outlines. The results are typical of those obtained using other training sets and the fact that patterns of variation are easily grasped makes the data ideal for illustrating the steps in development. A second, synthetic, training set, for which the ideal correspondences are known, is also used – particularly to illustrate the behaviour of different objective functions. Results for other training sets obtained using the fully developed method are presented in chapters 10 (2D) and 12 (3D).

Hand Model

The training set for the Hand model is shown in figure 5.2. Each hand is represented by a polygon with 400 sides of equal length. Using this data, two benchmark models were built: one with equally spaced points on the boundary – the simplest scheme; and one with manually positioned landmarks – the current gold standard. The equally spaced model was constructed by choosing one corresponding point on each shape (as shown in figure 5.2) and corresponding the remaining vertices of the polygons. The shapes were aligned using a similarity transformation to minimise the sum of squared distances between corresponding points [24]. Figure 5.3 shows the aligned boundaries and a subset of the correspondences.

The manual landmark model was generated by placing landmarks on easily distinguished points (such as the tips of the fingers) as shown in figure 5.4. A total of 400 points were sampled by equally spacing points along the boundaries between the landmarks. The shapes were aligned using the same method as for the arc-length models [24] – the aligned shapes are shown in figure 5.4.

The first two modes of variation for the equally spaced and manual models are shown in figures 5.5 and 5.6 respectively. The generalisation, specificity and compactness are plotted in figure 5.7. The error bars indicate one standard error on either side of the measured value. If the error bars do not overlap we can conclude that the difference is statistically significant. If the error bars do overlap, the difference is probably not significant in a statistical sense. The results show that – for this particular training set – manual landmarking results in a better model than that obtained by equally spacing points on the boundary, though the results for compactness are barely significant.

Bump Model

The training set for the Bump model is shown in figure 5.8. The examples are generated from a synthetic object that exhibits a single mode of shape variation where the 'bump' moves along the top of the box. Given an appropriate set of correspondences, it should be possible to describe the variation with a single parameter. An ideal model is obtained if the landmarks shown in figure 5.9 are corresponded and intermediate correspondences are established by equally spacing points along the boundary giving 300 points in total. The model has one non-zero[1] eigenvalue of 13665 – the corresponding mode of variation correctly captures the known behaviour, as shown in figure 5.10.

An equally spaced points model was created by setting the origin to the top left

[1] To the limits of numerical accuracy.

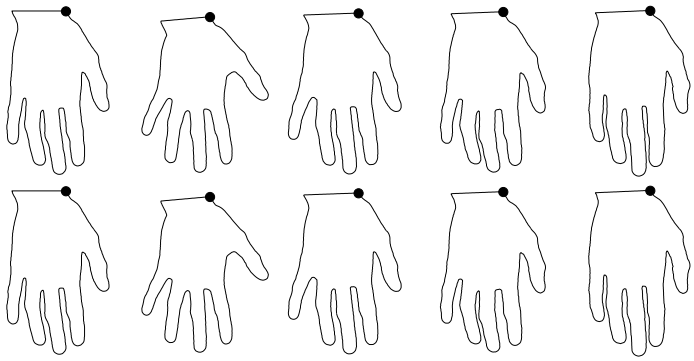


Figure 5.2. The training set of 10 hand outlines used for the evaluations. The circles represent the manual landmarks that define the starting point of each parameterisation.

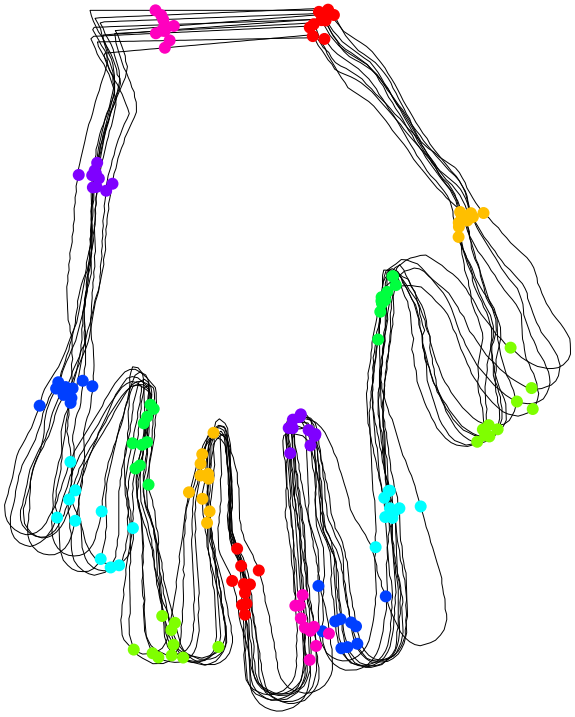


Figure 5.3. A subset of the ocr-respondences and the alignment used to build the equally spaced Hand model.

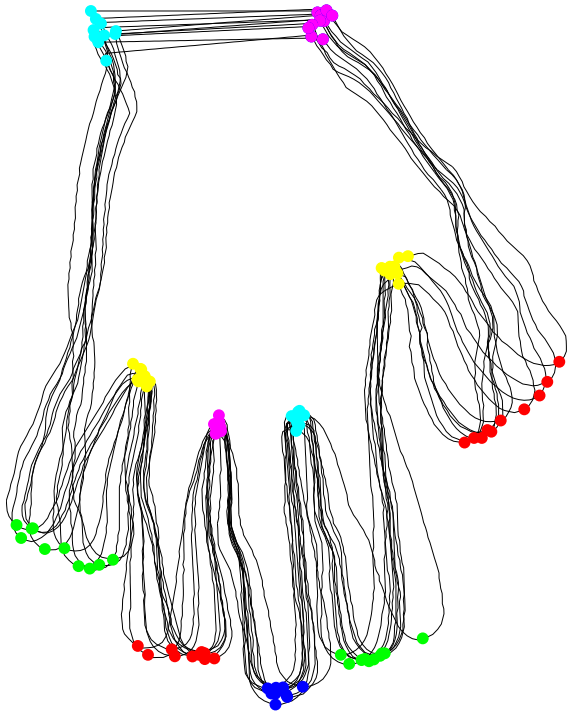


Figure 5.4. The manual landmarks and alignment used to build the manual Hand model.

hand corner of each example and equally spacing 300 points around the boundary. The first three eigenvalues of the resulting models are (5663,1347,155) – the corresponding modes of variation are shown in figure 5.11. The figure shows clearly that the model can generate invalid instances of the object. Graphs of generalisation ability, specificity and compactness are shown in figure 5.12. The plots show that, as expected, the manual model has significantly better generalisation and specificity properties than the equally spaced model for all number of

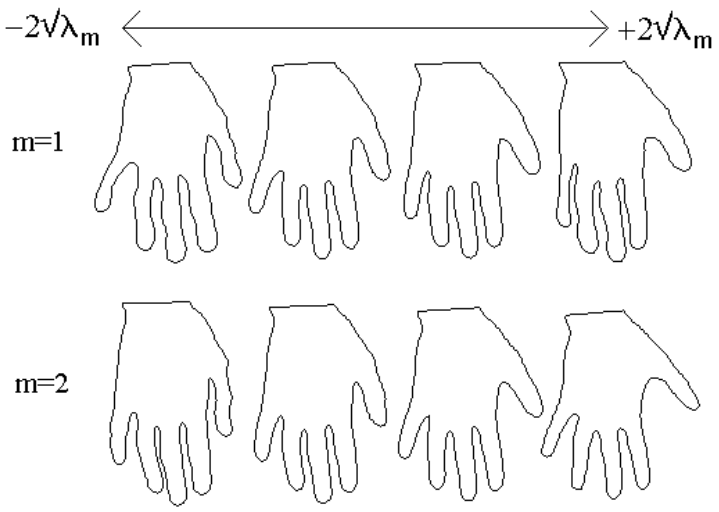


Figure 5.5. The first two modes of variation of the model built by the equally spaced model of the Hand training set. The figure show the effect of independently varying the first two modes of variation by ± 2[standard deviations found over the training set].

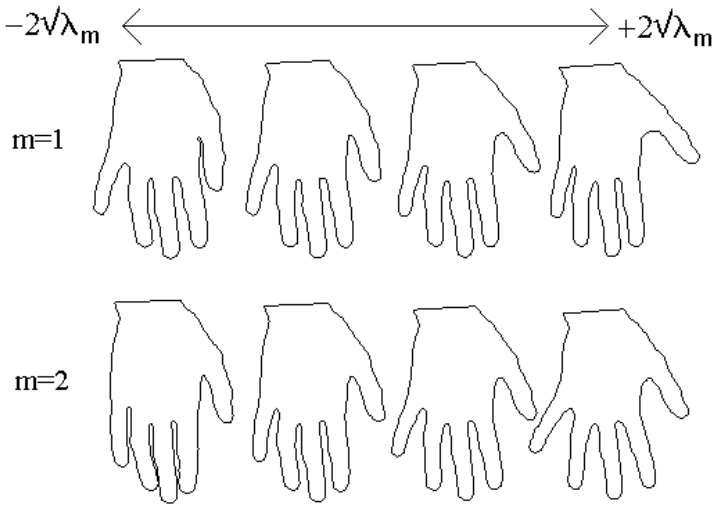
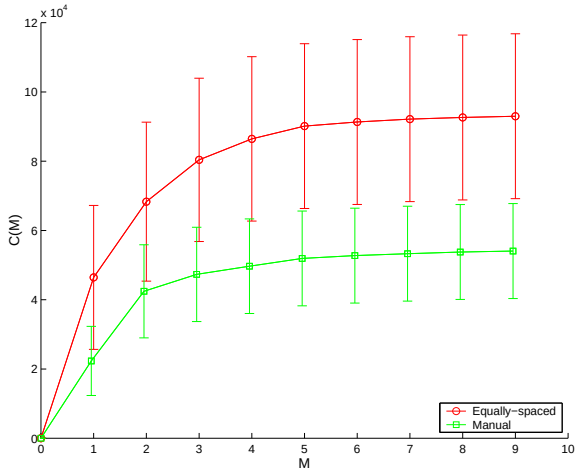
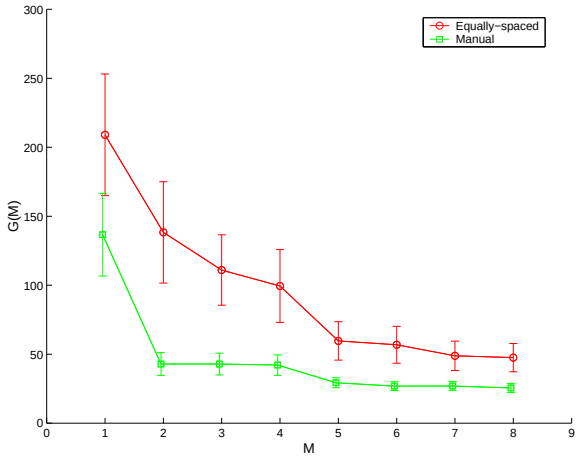


Figure 5.6. The first two modes of variation of the manual landmark model for the Hand training set. The figure show the effect of independently varying the first two modes of variation by ± 2[standard deviations found over the training set]

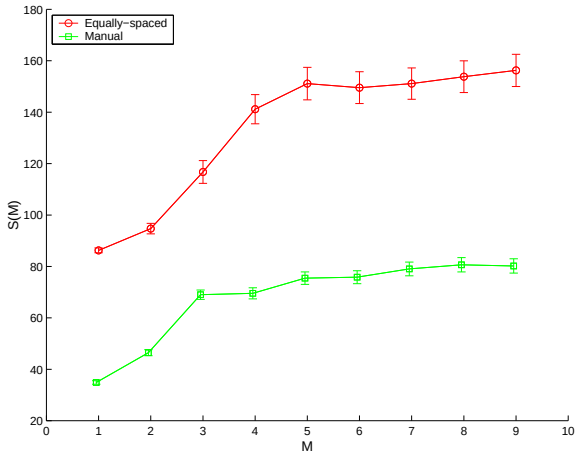
modes. It is interesting to note, however, that the equally spaced model is more compact than the ideal model. This is discussed in more detail in §6.1.1.



Compactness.



Generalisation Ability.



Specificity.

Figure 5.7. The compactness, generalisation and specificity for the equally spaced and ideal model of the Hand data. M is the number of modes used.

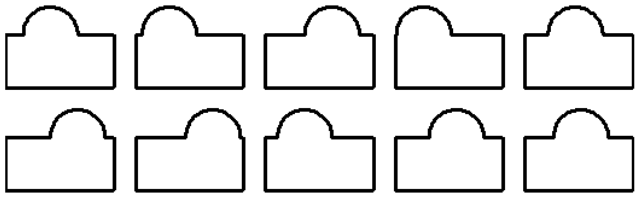


Figure 5.8. The synthetic training set for the Bump model.

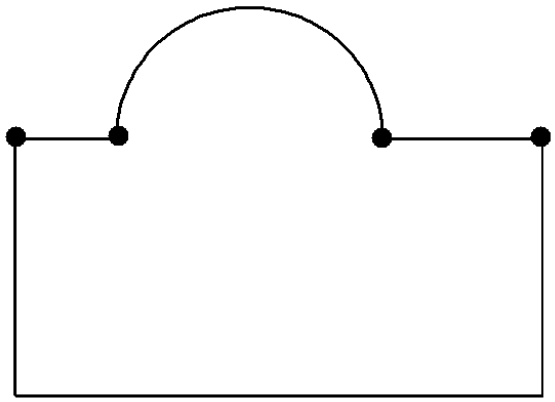


Figure 5.9. The correspondences required to construct the ideal Bump model.

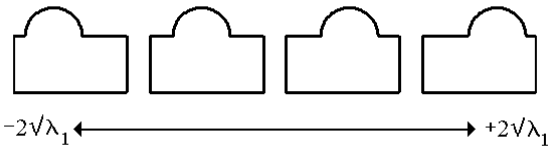


Figure 5.10. The first and only mode of variation of the ideal model built using manual landmarks.

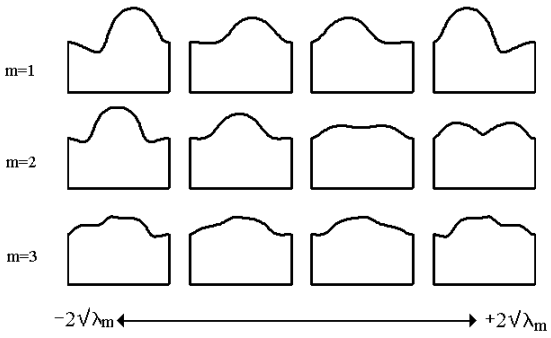
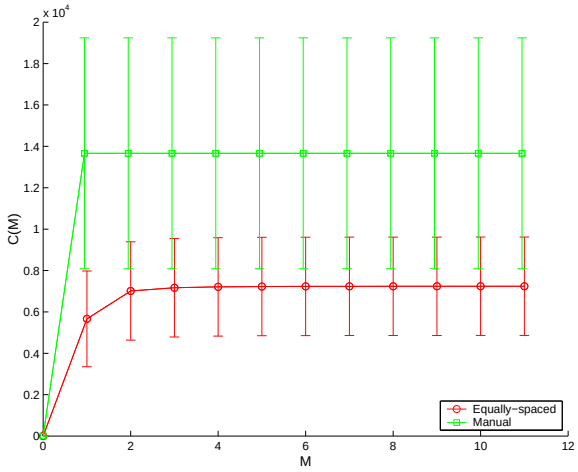
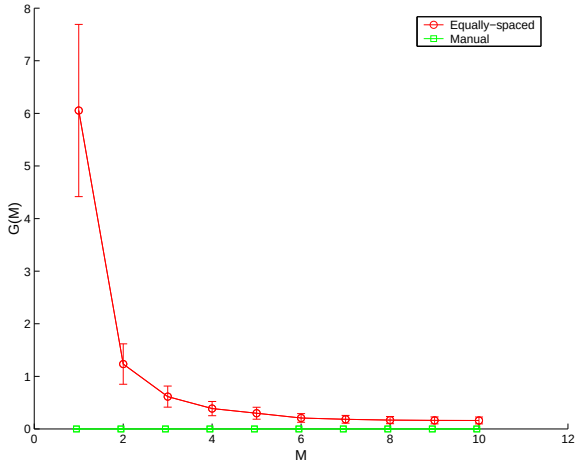


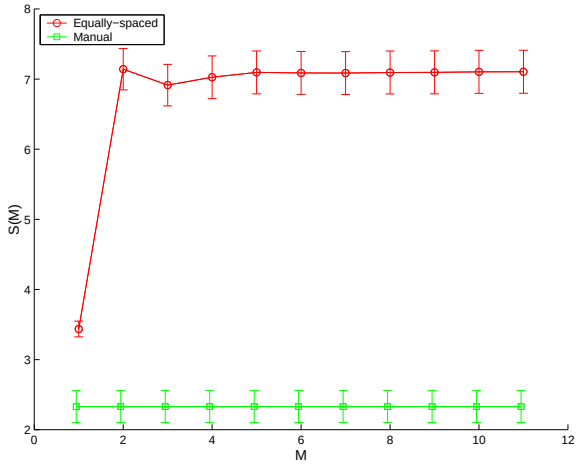
Figure 5.11. The first three modes of variation of the equally-spaced Bump model.



Compactness.



Generalisation Ability.



Specificity.

Figure 5.12. The compactness, generalisation and specificity for the equally spaced and ideal model of the Bump data.

Chapter 6

The Objective Function

"Pluralitas non est ponenda sine neccesitate"
– William of Ockham

A key component of the proposed approach is an objective function to assess the 'quality' of a model constructed from a given set of correspondences. This chapter begins by considering objective functions that have been proposed previously and illustrates their behaviour using synthetic data. This is followed by a detailed derivation of a new objective function, based on ideas from information theory, that measures model utility in a principled way.

6.1 Previous Work

6.1.1 The Trace of the Model Covariance

Several authors (e.g [6, 48, 79, 5]) have proposed optimising correspondences by minimising the trace of the covariance matrix, $\mathbf{D}$, of the training data:

$$F_T = Tr(\mathbf{D}) = \sum_m \lambda^m, \tag{6.1}$$

where $\{\lambda^m\}$ are the eigenvalues of $\mathbf{D}$.

F_T measures the sum of squared point-to-point distances between each training shape and the mean. Note that this is the same as the compactness measure

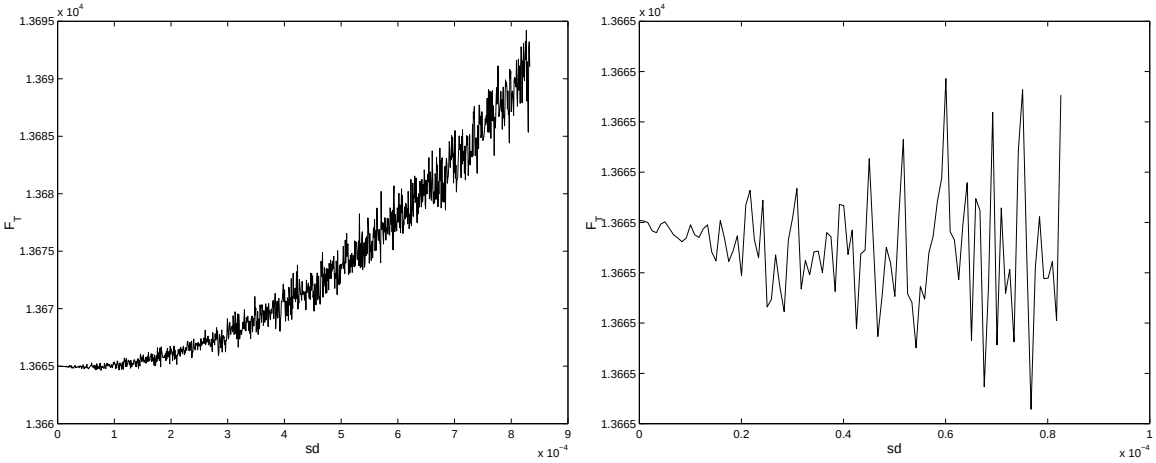


Figure 6.1. The behaviour of F_T when the correspondences of the ideal Bump model are perturbed by Gaussian random noise of increasing standard deviation. The plot shows the value of F_T averaged over $1,000$ realisations of noise. The graph on the right is a blow-up of the region close to the origin.

defined in (5.5). By minimising F_T, points are moved as close as possible to the mean, directly minimising the total variance of the model. The behaviour of F_T close to the optimum can be investigated using the Bump training set (see §5.3.2) by perturbing the correct correspondences using Gaussian random noise of increasing standard deviation, as shown in figure 6.1. The graph show that F_T has many local minima and the smallest value is not at zero displacement (i.e. the ideal model).

The value of F_T for the equally spaced points and ideal models are 7240 and 13665 respectively. Thus the equally spaced points model is preferred even though we have already seen (see figure 5.12) that it is less specific.

6.1.2 The Determinant of the Model Covariance

Kotcheff and Taylor [59] propose using the determinant of the covariance matrix, **D**, as an objective function:

$$F_D = \log(det(\mathbf{D})) = \log\left(\prod_m \lambda^m\right) = \sum_m \log(\lambda^m), \tag{6.2}$$

where $\{\lambda^m\}$ are the eigenvalues of **D**; the $\log$ is taken to stop roundoff errors dominating the calculation. F_D effectively measures the 'volume' that the training

set occupies in shape space which tends to favour compact models. Inspection of (6.2) shows that the objective function has a degenerate minimum when any eigenvalue approaches zero. To overcome this, Kotcheff and Taylor add a small regularisation constant ϵ:

$$F_D = \log(det(\mathbf{D} + \mathbf{I}\epsilon)) = \sum_m \log(\lambda^m + \epsilon), \tag{6.3}$$

where $\mathbf{I}$ is the identity matrix. They argue that an appropriate value of ϵ can be chosen with reference to the noise on the training shapes. Figures 6.2 and 6.3 show the behaviour of F_D as the correspondences of the ideal Bump model are displaced with Gaussian noise of increasing standard deviation, for two values of ϵ – one calculated as suggested by Kotcheff and Taylor and one several orders of magnitude larger. Both plots show a minimum value for F_D at zero displacement, though the behaviour away from the minimum is quite different. The value of F_D for the equally spaced points model and ideal model are 37.22 and 9.71 respectively (for ϵ = 0.01).

In summary, F_D seems to display appropriate behaviour, but there is no rigorous justification for its choice. It explicitly favours compact models (in a certain sense) but there is no particular reason to suppose that it will favour models with good specificity and generalisation properties.

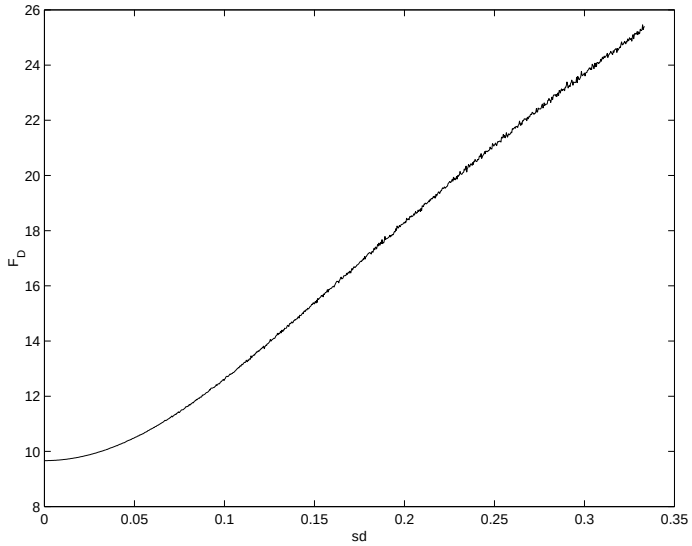


Figure 6.2. The behaviour of F_D ($\epsilon = 0.01$) when the correspondences of the ideal Bump model are perturbed by Gaussian random noise of increasing standard deviation. The plot shows the value of F_D averaged over $1,000$ realisations of noise.

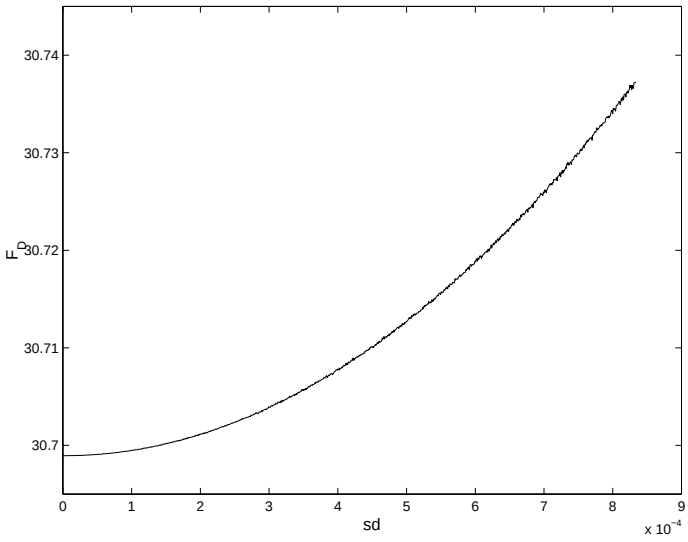


Figure 6.3. The behaviour of F_D ($\epsilon = 10$) when the correspondences of the ideal Bump model are perturbed by Gaussian random noise of increasing standard deviation. The plot shows the value of F_D averaged over $1,000$ realisations of noise.

6.2 An Information Theoretic Objective Function

We seek a principled basis for choosing an objective function that will directly favour models with good generalisation ability, specificity and compactness. The ability of a model to generalise *whilst being specific* depends on its ability to interpolate and, to some extent, extrapolate the training set. In order to achieve these properties, we apply the principle of Occam's razor, which can be paraphrased as: "the simplest description of the training set will interpolate/extrapolate best".

The notion of the 'simplest description' can be formalised using ideas from information theory – in particular, by applying the minimum description length (MDL) principle [81, 82]. The basic idea is to minimise the length of a message required to transmit a full description of the training set, using the model to encode the data. As the message is encoded, the receiver must know the encoding model in order to fully reconstruct the original data, making it necessary to measure the description length of the encoding model as well as the encoded data.

6.2.1 The Encoding Model

Our training data are a set of n_p-dimensional shape vectors $\{\mathbf{x}_i : i = 1 \dots n_s\}$. These are coded with a multivariate Gaussian model. Principal component anal-

ysis is used to find a co-ordinate system aligned with the principal directions of the data, giving the linear model defined earlier (3.7):

$$\mathbf{x}_i = \bar{\mathbf{x}} + \sum_{m=1}^{n_s-1} \mathbf{p}^m b_i^m, \tag{6.4}$$

where $\{\mathbf{p}^m : m = 1 \dots n_s - 1\}$ are eigenvectors that define the principal axes of the data. The eigenvectors are normalised to have unit length. Each eigenvector has a corresponding eigenvalue λ^m that gives the variance of the data in the m^{th} direction. The eigenvalues (and corresponding eigenvectors) are sorted by descending value.

The set of eigenvectors, $\{\mathbf{p}^m\}$, define an $(n_s - 1)$-dimensional "shape space" in which the m^{th} coordinate of the i^{th} shape, $\mathbf{x}_i$, is given by:

$$y_i^m = (\mathbf{x}_i - \bar{\mathbf{x}})^T \mathbf{p}^m. \tag{6.5}$$

The description length of the mean shape $\bar{\mathbf{x}}$ and the eigenvectors $\{\mathbf{p}^m\}$ are assumed to be constant for a given training set[1] thus only the description length of the shape space coordinates $\{y_i^m : i = 1 \dots n_s, m = 1 \dots n_s - 1\}$ that describe the training shapes need to be calculated.

As the eigenvectors are mutually orthornormal, the total description length can be decomposed to a sum over the $(n_s - 1)$ dimensions of shape space:

$$\mathcal{L}_{\text{Total}} = \sum_{m=1}^{n_s-1} \mathcal{L}^m. \tag{6.6}$$

Hence we only require an expression for the description length of the 1D datasets, $Y^m = \{y_i^m : i = 1 \dots n_s\}$. Since the data, Y^m, in each principle direction has a mean of zero we use a set of one-parameter Gaussian models to encode the data:

$$\rho(y^m; \sigma^m) = \frac{1}{\sigma^m \sqrt{2\pi}} \exp\left(-\frac{(y^m)^2}{2(\sigma^m)^2}\right). \tag{6.7}$$

6.2.2 Calculating Description Length[1]

[1]For continuous shapes ($n_p \to \infty$), both the mean shape and the eigenvectors have an infinite description length.

[1]The author was responsible for developing the MDL framework. The original derivation of

Given a value, $\hat{\alpha}$, encoded using a probabilistic model, $P(\hat{\alpha})$, its description length can be calculated using Shannon's codeword length [89]:

$$\mathcal{L} = -\log P(\hat{\alpha}), \tag{6.8}$$

all logarithms are calculated to base 2, giving the message lengths in units of bits. To utilise this expression, the values must be quantised and the range of the data must be known[2].

Quantisation

Describing a real number to an arbitrary accuracy requires an infinite amount of information, leading to an incalculable description length. In practice, each data value, y_i^m, must be quantised to some accuracy Δ:

$$y_i^m \rightarrow \hat{y}_i^m,\ \hat{y}_i^m = n\Delta,\ n \in \mathbb{Z}, \tag{6.9}$$

where the notation $\hat{y}$ is used to represent the quantised value of y. The value of Δ should be related to the expected noise on the training shapes (typically one pixel/voxel in the original images from which they were annotated).

Data Range

The range of the data can be calculated in the original space so that:

$$-\frac{r}{2} \leq x_{i\alpha} \leq \frac{r}{2} \text{ for all } \alpha = 1, \ldots n_p,\ i = 1, \ldots n_s, \tag{6.10}$$

and transformed to give the range of the data, R, in shape space:

$$R = r\sqrt{n_p}, \text{ so that } |y_i^m| \leq R \text{ for all } i, m. \tag{6.11}$$

this form of objective function was due to Dr. C J Twining.

[2]The range and accuracy of the data can be agreed prior to transmission hence they can be excluded from the calculation of the total description length.

6.2.3 The Description Length of a 1D Gaussian Model

We have established that the total description length of the training data can be calculated independently for each principle direction, m. To utilise this result, we require an expression that gives the description length of a 1D dataset $Y = \{y_i : i = 1 \dots n_s\}$ using a Gaussian model (6.7). Using a two part coding scheme [82], the description length can be decomposed to:

$$\mathcal{L} = \mathcal{L}_{\text{Parameters}} + \mathcal{L}_{\text{Data}}. \tag{6.12}$$

Coding the Parameters

$\mathcal{L}_{\text{Parameters}}$ is the description length for sending the σ parameter, calculated from the quantised values of Y:

$$\sigma = \sqrt{\frac{1}{n_s}\sum_{i=1}^{n_s} \hat{y}_i^2}. \tag{6.13}$$

σ must be quantised to some accuracy δ:

$$\hat{\sigma} = n\delta, \;\; n \;\epsilon\; \mathbb{N}, \tag{6.14}$$

where $\hat{\sigma}$ is chosen to be the closest possible value to the unquantised value of σ (6.13). Although a simple modelling choice could be made by coding σ to an accuracy $\delta = \Delta$ (the same as the data), a more precise estimate of the description length can be obtained by optimising the value of δ analytically. It will be shown later that the value of $\hat{\sigma}$ is bounded such that:

$$\sigma_{min} \leq \hat{\sigma} \leq \sigma_{max}. \tag{6.15}$$

In the absence of any prior knowledge, $\hat{\sigma}$ is assumed to follow a uniform distribution over $[\sigma_{min}, \sigma_{max}]$ hence the description length of $\hat{\sigma}$ becomes:

$$\begin{aligned} \mathcal{L}_{\hat{\sigma}} &= -\log\left(P(\hat{\sigma})\right), \\ &= \log\left(\frac{\sigma_{max} - \sigma_{min}}{\delta}\right). \end{aligned} \tag{6.16}$$

In order to decode the message, the receiver must know the accuracy, δ. Rissanen [82] gives an exact calculation for the description length of such a value but the expression uses a computationally expensive recursive function. Alternatively, if δ is of the form:

$$\delta = 2^{\pm k}, \tag{6.17}$$

the description length can be calculated from:

$$\mathcal{L}_\delta = 1 + |\log \delta|. \tag{6.18}$$

From (6.16) and (6.18), the description length of the parameters is given by:

$$\begin{aligned} \mathcal{L}_{\text{Parameters}} &= \mathcal{L}_{\hat{\sigma}} + \mathcal{L}_\delta \\ &= \log\left(\frac{\sigma_{max} - \sigma_{min}}{\delta}\right) + 1 + |\log \delta|. \end{aligned} \tag{6.19}$$

Coding the Data

Using the Gaussian model in (6.7), the probability, $P(\hat{y})$, of a value lying in the range $\hat{y} \pm \frac{\Delta}{2}$ is:

$$P(\hat{y}) = \int_{\hat{y}-\frac{\Delta}{2}}^{\hat{y}+\frac{\Delta}{2}} dk\, \rho(k; \hat{\sigma}), \tag{6.20}$$

this can be approximated to first order by:

$$P(\hat{y}) \approx \frac{\Delta}{\hat{\sigma}\sqrt{2\pi}} \exp\left(-\frac{\hat{y}^2}{2\hat{\sigma}^2}\right). \tag{6.21}$$

For all values $\hat{\sigma} \geq 2\Delta$, the approximation has a mean fractional error of $< 1\% \pm 0.8\%$ hence $\sigma_{min} = 2\Delta$ is chosen as a lower bound. Without prior knowledge, $\hat{\sigma}$ is assumed to have the same range as the data, hence $\sigma_{max} = \frac{R}{2}$ is chosen as an upper bound.

If σ falls below σ_{min}, the approximation in (6.21) does not hold hence an alternative scheme must be used to code the data, Y. The case where all data, Y, has the

same quantised value (i.e. the range of $Y \leq \Delta$) must also be addressed, giving three different coding schemes:

- **Case 1**: If $\sigma > \sigma_{min}$: explicitly code the data.
- **Case 2**: If $\sigma \leq \sigma_{min}$: estimate σ with σ_{min}.
- **Case 3**: If the range of $Y \leq \Delta$.

Each case is now considered in more detail.

Case 1: $\sigma > \sigma_{min}$

Using (6.21), the code length for the data is:

$$\begin{aligned} \mathcal{L}_{\text{data}} &= -\sum_{i=1}^{n_s} \log P(\hat{y}_i) \\ &= -n_s \log \Delta + \frac{n_s}{2} \log(2\pi\hat{\sigma}^2) + \frac{1}{2\hat{\sigma}^2} \sum_{i=1}^{n_s} \hat{y}_i^2. \end{aligned} \tag{6.22}$$

We must now decide on the accuracy, δ, to which to describe σ. The real value of σ can vary from the quantised value, $\hat{\sigma}$, by $\pm\frac{\delta}{2}$. Assuming a uniform distribution over this range, the expectation value of $\hat{\sigma}$ can be obtained by averaging over the interval:

$$\left\langle \frac{1}{\hat{\sigma}^2} \right\rangle = \frac{1}{\delta} \int_{\hat{\sigma}-\delta/2}^{\hat{\sigma}+\delta/2} dy \frac{1}{y^2} = -4 \left(-4\,\sigma^2 + \delta^2\right)^{-1} \approx \frac{1}{\sigma^2} \left(1 + \frac{\delta^2}{4\sigma^2}\right), \tag{6.23}$$

$$\left\langle \log(\hat{\sigma}^2) \right\rangle = \frac{1}{\delta} \int_{\hat{\sigma}-\delta/2}^{\hat{\sigma}+\delta/2} dy \log(y^2) \approx \log \sigma^2 - \frac{\delta^2}{12\sigma^2}. \tag{6.24}$$

Substituting (6.23), (6.24) into (6.22) and using (6.13):

$$\mathcal{L}_{\text{data}} = -n_s \log \Delta + \frac{n_s}{2} \log(2\pi\sigma^2) + \frac{n_s}{2} + \frac{n_s\delta^2}{12\sigma^2}. \tag{6.25}$$

Substituting (6.25) and (6.19) into (6.12):

$$\begin{aligned}\mathcal{L} = \quad & \log\left(\frac{\sigma_{max}-\sigma_{min}}{\delta}\right) + 1 + |\log\delta| \\ & -n_s \log\Delta + \frac{n_s}{2}\log(2\pi\sigma^2) + \frac{n_s}{2} + \frac{n_s\delta^2}{12\sigma^2}.\end{aligned} \tag{6.26}$$

The optimal accuracy of σ can be found by differentiating $\mathcal{L}_{\text{total}}$ with respect to δ and setting the derivative to zero, giving:

$$\delta^*(\sigma, n_s) = \min\left(1, \sigma\sqrt{\frac{12}{n_s}}\right), \tag{6.27}$$

where δ^* is the optimum value of δ. Substituting δ in (6.26) with δ^* yields the description length of Y using the first coding scheme ($\sigma > \sigma_{min}$):

$$\begin{aligned}\mathcal{L}^{(1)} = \quad & \log\left(\frac{\sigma_{max}-\sigma_{min}}{\delta^*}\right) + 1 + |\log\delta^*| \\ & -n_s \log\Delta + \frac{n_s}{2}\log(2\pi\sigma^2) + \frac{n_s}{2} + \frac{n_s\delta^{*2}}{12\sigma^2}.\end{aligned} \tag{6.28}$$

Case 2: $\sigma \leq \sigma_{min}$

If $\sigma \leq \sigma_{min}$, the value of δ cannot be optimised and the substitution $\sum \hat{y}^2 \rightarrow n_s\sigma^2$ cannot be made.

Although the approximation in (6.21) is inaccurate, it can still be used as it is always overestimates the description length. The value of δ is chosen using $\delta^*(\sigma_{min}, n_s)$ in (6.27) to find the optimum value at σ_{min}. Given this, the total description length, $\mathcal{L}^{(2)}$, for the second case ($\sigma \leq \sigma_{min}$) can be obtained from (6.22) and (6.16):

$$\begin{aligned}\mathcal{L}^{(2)} = \quad & \log\left(\frac{\sigma_{max}-\sigma_{min}}{\delta^*}\right) + 1 + |\log\delta^*| \\ & -n_s \log\Delta + \frac{n_s}{2}\log(2\pi\sigma_{min}^2) + \frac{1}{2\sigma_{min}^2}\sum_{i=1}^{n_s}\hat{y}_i^2.\end{aligned} \tag{6.29}$$

Case 3: The Range of $Y \leq \Delta$

The final case is when the range of $Y \leq \Delta$ where only the position of the data (i.e. the mean) needs to be sent. As the mean is always zero, it costs nothing to describe hence:

$$\mathcal{L}^{(3)} = 0. \tag{6.30}$$

6.2.4 The Total Description Length

From (6.6), (6.28), (6.29) and (6.30) we have the following expression for the total description length for the training set:

$$\mathcal{L}_{\text{total}}(\Delta) = \sum_{m=1}^{n_s-1} \mathcal{L}^m, \tag{6.31}$$

where:

$$\mathcal{L}^m = \begin{cases} \log\left(\frac{\sigma_{max}-\sigma_{min}}{\delta^*}\right) + 1 + |\log\delta^*| - n_s \log\Delta \\ +\frac{n_s}{2}\log(2\pi(\sigma^m)^2) + \frac{n_s}{2} + \frac{n_s\delta^{*2}}{12(\sigma^m)^2} & \text{if } \sigma^m > \sigma_{min} \\ \log\left(\frac{\sigma_{max}-\sigma_{min}}{\delta^*}\right) + 1 + |\log\delta^*| - n_s \log\Delta \\ +\frac{n_s}{2}\log(2\pi\sigma_{min}^2) + \frac{1}{2\sigma_{min}^2}\sum_{i=1}^{n_s}(\hat{y}_i^m)^2 & \text{if } \sigma^m \leq \sigma_{min} \\ 0 & \text{if the range of } Y^m \leq \Delta \end{cases} \tag{6.32}$$

Although this level of complexity is necessary to deal with the general case, further insight can be gained if we consider an appropriate limiting case.

If the number of shapes is sufficiently large, then from (6.27):

$$\delta^* \to \sigma^m\sqrt{\frac{12}{n_s}}. \tag{6.33}$$

In the limit $\Delta \to 0$, the quantised values approach the original values:

$$\hat{y}^m \to y^m \quad \text{and} \quad (\hat{\sigma}^m)^2 \to \frac{1}{n_s}\sum_{i=1}^{n_s}(y_i^m)^2. \tag{6.34}$$

Using these results:

$$\begin{aligned}\mathcal{L}^{(1)}(\sigma^m, n_s, R, \Delta) = \quad & \log(\sigma_{max} - \sigma_{min}) + 1 - 2\log\left(\sqrt{\frac{12}{n_s}}\right) \\ & -n_s \log\Delta + \frac{n_s}{2}\log(2\pi) + \frac{n_s}{2} \\ & -2\log(\sigma^m) + n_s\log(\sigma^m),\end{aligned} \tag{6.35}$$

and:

$$\begin{aligned}\mathcal{L}^{(2)}(\sigma^m, n_s, R, \Delta) = \quad & \log(\sigma_{max} - \sigma_{min}) + 1 - 2\log\left(\sqrt{\frac{12}{n_s}}\right) \\ & -n_s \log\Delta + \frac{n_s}{2}\log(2\pi) - 2\log(\sigma_{min}) \\ & +\frac{n_s}{2}\log(\sigma_{min}^2) + \frac{n_s}{2}\left(\frac{\sigma^m}{\sigma_{min}}\right)^2.\end{aligned} \tag{6.36}$$

Substituting (6.35) and (6.36) into (6.31) gives:

$$\begin{aligned}\mathcal{L}_{\text{total}}(\Delta) = \quad & \textstyle\sum_{p=1}^{n_g} \quad f(n_s, R, \Delta) + (n_s - 2)\log(\sigma^p) + \frac{n_s}{2} \\ & + \textstyle\sum_{q=n_g+1}^{n_g+1+n_{min}} \quad f(n_s, R, \Delta) + (n_s - 2)\log(\sigma_{min}) + \frac{n_s}{2}\left(\frac{\sigma^q}{\sigma_{min}}\right)^2,\end{aligned} \tag{6.37}$$

where n_g is the number of directions where the first case ($\sigma^m > \sigma_{min}$) holds, n_{min} is the number of directions where the second case ($\sigma^m \leq \sigma_{min}$) holds and $f(n_s, R, \Delta)$ is a function that is constant for a given training set.

In this limit, the first term is used to explicitly encode the data, ensuring that the model fits well to the data. The second term approximates the data using a simpler model. Thus the MDL objective function uses the two cases to trade off the model complexity against its ability to fit to the data.

Inspection of (6.37) reveals that the $\log(\sigma)$ terms dominate the expression. This is similar to the determinant-based objective function used by Kotcheff and Taylor [59] ($\sum_m 2\log(\sigma^m + \epsilon)$) but the more complete treatment shows that other terms are also important.

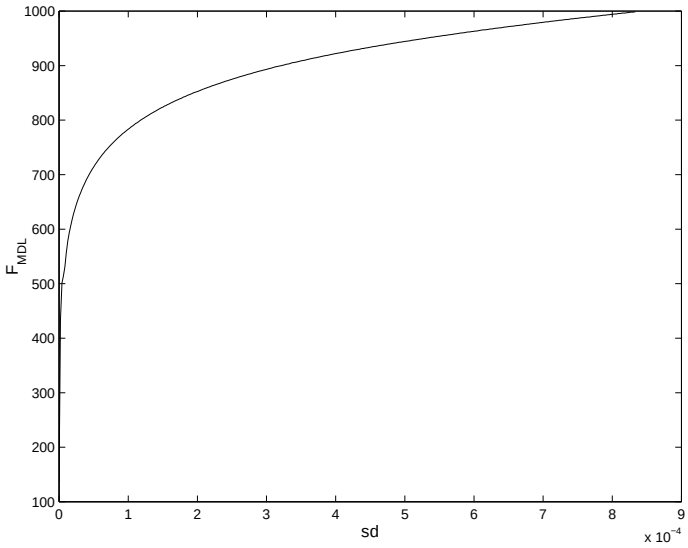


Figure 6.4. The behaviour F_{MDL} ($\Delta = 0.1$ pixels) when the correspondences of the ideal Bump model are perturbed by Gaussian random noise of increasing standard deviation. The plot shows the value of F_{MDL} averaged over $1,000$ realisations of noise.

6.2.5 The MDL Objective Function

From (6.31), we can define an objective function that measures the description length of the training set using a given encoding model.

$$F_{MDL} = \mathcal{L}_{total}(\Delta). \tag{6.38}$$

This expresses the notion that the model which most efficiently (simply) encodes the training set is likely to interpolate/extrapolate the training data most reliably. Figure 6.4 shows the behaviour of F_{MDL} as the correspondences on the ideal Bump model are displaced with Gaussian noise of increasing standard deviation. A value of $\Delta = 0.1$ pixels was assumed.

Although Davies *et al.* have previously shown that using F_{MDL} as an objective function produces good shape models [31, 30, 29], switching between the cases in (6.32) causes significant discontinuities. The point at which switching occurs depends on the choice of Δ, which should be comparable with the noise of the training data. Rather than choosing a single value of Δ, it is more realistic to assume a distribution of values. Averaging F_{MDL} over the distribution of Δ results in an objective function with more continuous behaviour close to convergence. In the experiments reported below a uniform distribution for Δ over the range Δ_{min} to Δ_{max} is assumed.

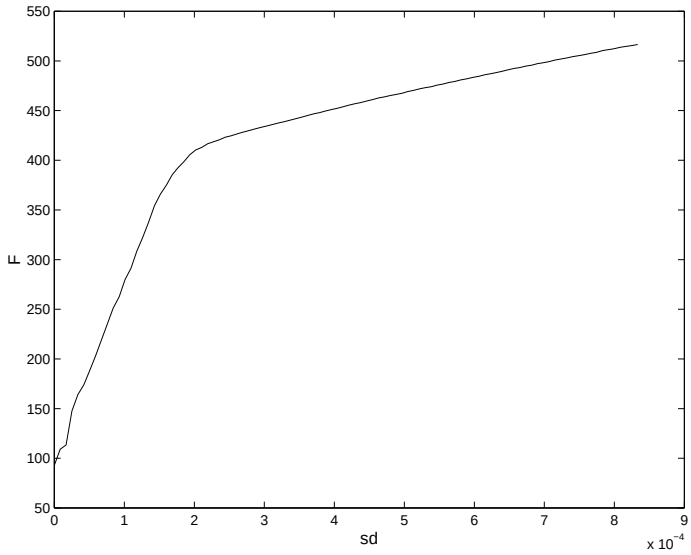


Figure 6.5. The behaviour of F in (6.39) when the correspondences of the ideal Bump model are perturbed by Gaussian random noise of increasing standard deviation. The plot shows the value of F averaged over $1,000$ realisations of noise.

$$F = \int_{\Delta_{min}}^{\Delta_{max}} d\Delta \mathcal{L}_{\text{total}}(\Delta). \tag{6.39}$$

The integral can be solved by numerical integration (e.g. by using Simpson's rule [77]). We have used $\Delta_{min} = 0.01$ and $\Delta_{max} = 2$ pixels for the results reported in this thesis. The behaviour of F is shown in figure 6.5 by perturbing the ideal correspondences of the Bump model using Gaussian random noise of increasing standard deviation. The plot shows that F has a clear minimum with a reasonably large capture range. The numerical integration is computationally expensive to perform but the determinant-based objective function (6.3) can be used to obtain a good initial approximation to the optimum. The full objective function can then be used to refine the correspondences.

6.3 Summary

We have derived an objective function that favours simple models that are likely to interpolate/extrapolate well from the training set. For synthetic data, the objective function behaves appropriately in the region of the ideal solution. It has one parameter, Δ, which has a clear interpretation in terms of the noise (or noise distribution) on the training data.

In the dual limit ($n_s \rightarrow \infty$, $\Delta \rightarrow 0$), the objective function is similar to the determinant-based function proposed by Kotcheff and Taylor [59]. We have found in practice that the two objective functions produce very similar results given a suitable value of ϵ in (6.3), but the MDL function is well-defined under all circumstances. As the determinant-based function is computationally simpler (it does not require numerical integration or data quantisation), we use it to obtain an initial estimate of the MDL solution.

Chapter 7

Building Models Using Piecewise Linear Re-parameterisation

"Order and simplification are the first steps toward mastery of a subject - the actual enemy is the unknown."
– Thomas Mann.

This chapter shows how the MDL objective function can be combined with a parameterised representation of correspondence to build optimal shape models. Correspondences are manipulated using a piecewise linear (PL) representation of re-parameterisation that straightforwardly extends to 3D. Two methods are investigated for optimising the objective function with respect to the re-parameterisations of the training examples. The resulting models are evaluated using the methods described in chapter 5.

7.1 Representing Re-parameterisation

Chapter 5 introduced the idea of manipulating correspondence in 2D by re-parameterising each example using a monotonic function, ϕ_i. This section reviews the piecewise-linear representation of re-parameterisation used previously by Kotcheff and Taylor [59] before describing an improved representation with a straightforward extension to 3D.

7.1.1 A PL Representation of Re-parameterisation

Kotcheff and Taylor [59] define a piecewise-linear representation of ϕ for 2D shapes using a set of nodes $\{p_i\}$. Linear interpolation is used to evaluate ϕ in between nodes to give a continuous representation (see figure 7.1):

$$\phi(u) = \phi(u_i) + (\phi(u_{i+1}) - \phi(u_i)) \frac{u - u_i}{u_{i+1} - u_i}, \quad u_i \leq u \leq u_{i+1}, \tag{7.1}$$

where u_i and $\phi(u_i)$ are the path-length parameterisation and re-parameterisation of node p_i respectively.

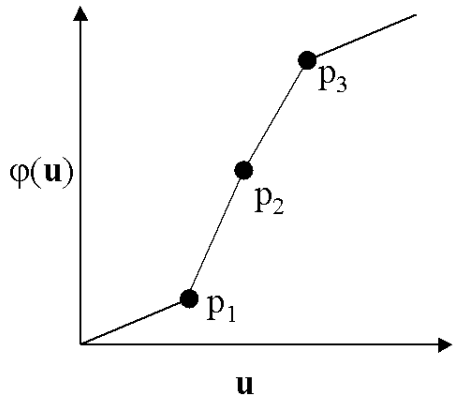


Figure 7.1. A piecewise-linear representation of $\phi(u)$ using three nodes (p_1, p_2, p_3). Each node is allowed to move vertically.

To ensure that $\phi(u)$ is diffeomorphic [1], both u_i and $\phi(u_i)$ must be ordered by increasing value:

$$\begin{aligned} 0 \leq u_i \leq u_{i+1} \ldots \leq u_n \leq 1, \\ 0 \leq \phi(u_i) \leq \phi(u_{i+1}) \ldots \leq \phi(u_n) \leq 1. \end{aligned} \tag{7.2}$$

Kotcheff and Taylor impose this constraint directly by explicitly ordering their nodes, but this does not extend in any obvious way to 3D.

[1] $\phi(u)$ is not differentiable at the control points hence it is not diffeomorphic in a strict sense

7.1.2 A Recursive Definition of Re-parameterisation

For the representation of ϕ to be extendable to 3D, we need to impose an implicit rather than explicit ordering on the nodes $\{p_i\}$. This can be achieved by defining a recursive representation of the re-parameterisation function by inserting nodes between those already present. In 2D, the position of each new node is coded as its fractional distance τ_{ij} between its two parent nodes p_i and p_j. Thus, by constraining the positions of new nodes to lie in the range [0,1] (where the node has a value of 0 if it is positioned on its left parent, 1 on its right parent and 0.5 in the centre) we can enforce an *implicit ordering*. This is illustrated in figure 7.2.

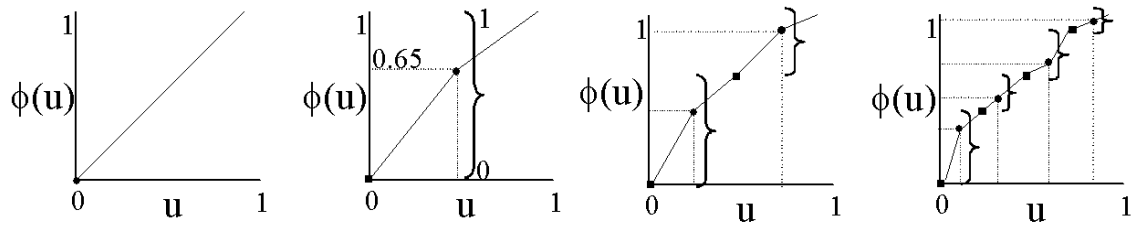


Figure 7.2. The recursive representation of re-parameterisation. The circles represent child nodes and squares represent parent nodes. The brackets show the range that each node is allowed to move. The node has a value of 0 at the bottom of the bracket, 1 at the top and 0.5 in the middle. The parameter values (fractional distances) for this example are: $\phi(0.65(0.65(0.8,0.4),0.7(0.2,0.5))$.

As the representation of re-parameterisation does not use an explicit ordering, it can be straightforwardly extended to represent surface re-parameterisation, as described in chapter 11.

7.2 Optimising the Correspondences

Two methods of optimising F with respect to the re-parameterisation function of the training set have been investigated – one global, the other local. The performance of the two methods is compared using the hand training set. The re-parameterisation functions were defined using three levels of recursion giving 7 (1+2+3) free parameters for each example:

$$\phi(\tau_{11}(\tau_{21}(\tau_{31},\tau_{32}),\tau_{22}(\tau_{33},\tau_{34}))). \tag{7.3}$$

In order for the optimisation to be well defined, a 'reference shape' must be se-

lected whose re-parameterisation is fixed for the entire optimisation. For now, the first example is used but it will be shown later how the selection can be automated.

7.2.1 Global Optimisation

If k levels of recursion are used to represent each re-parameterisation function, the resulting optimisation space is $(2^k - 1)(n_s - 1)$ - dimensional. This space is likely to contain many local minima leading us to prefer a stochastic, global optimisation method such as genetic algorithm search (GA) [42] or simulated annealing [57]. For the results reported below, a GA was used (see appendix A for a full description) with each chromosome encoding the $(2^k-1)(n_s-1)$ parameters. The following settings were used for the GA search[2]:

- Crossover operator: single point.
- Crossover rate: 100%.
- Mutation operator: single point.
- Mutation probability: 0.01%.
- Selection: roulette wheel, sigma scaling, non-elitist.

For the Hand data set, the algorithm converged after 243 iterations of the genetic algorithm which required 108896 evaluations of the genetic algorithm.

The behaviour of the resulting model is shown in figure 7.3 by varying the first two modes of variation by $\pm 2\sqrt{\lambda^m}$. The modes of variation reflect the variation we would expect of the hand outlines observed in the training set. Quantitative results are shown in figure 7.4. These show that the GA model has better generalisation ability, specificity and compactness properties than the equally spaced model but there is no significant improvement on the manual model.

These results are encouraging but this method of optimisation takes many hours to converge and scales very poorly with the number of training examples.

7.2.2 Local Search

Although stochastic algorithms can find the (approximately) global optimum in a high-dimensional search space, they are very inefficient and require many func-

[2]The implementation of the genetic algorithm was based on the freeware MATLAB implementation described in [51]

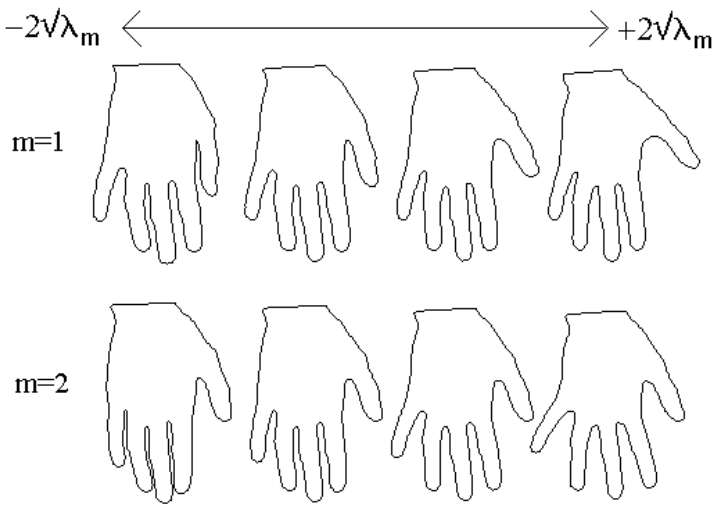
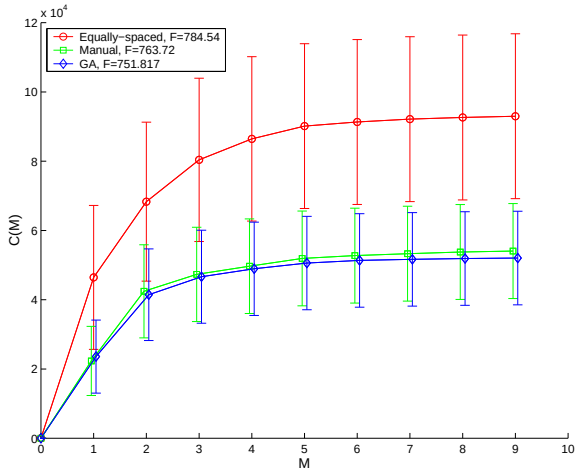


Figure 7.3. The first two modes of variation of the model built by genetic algorithm optimisation of 10 Hand data.

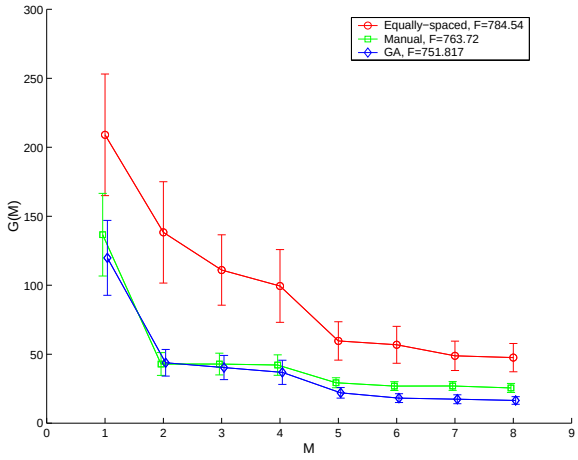
tion evaluations. More efficient algorithms exist that search for an optimum solution in a local region around a starting estimate. This approach has also been investigated.

The local optimisation algorithm used here (and for the remainder of the thesis) is the Nelder-Mead downhill simplex method (using the default parameters employed in [77]) which has some ability to avoid local minima. Although the concept is simple, the algorithm is complex and will not be described here. The reader is referred to [77] for an introduction and to [61] for a more detailed analysis. The value of the current parameterisation (initially the identity) is used as a starting estimate for each iteration ensuring that correspondences can only be improved.

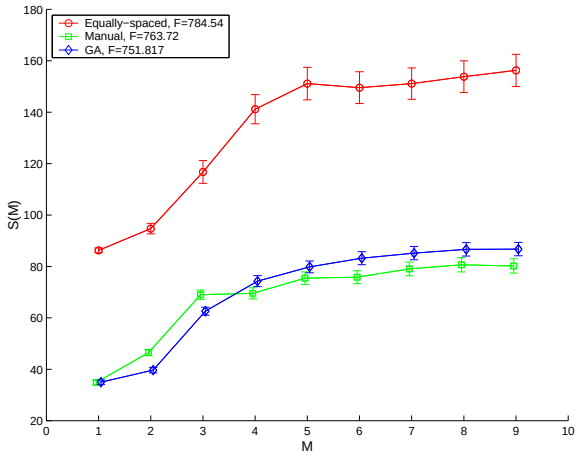
The number of parameters used in the previous section was found to be too large for the local optimiser to converge reliably. To overcome this, the number of parameters optimised concurrently must be reduced. This can be achieved by using a greedy algorithm to optimise the parameters at each level of recursion before moving on to the next, giving $(n_s - 1)2^{k-1}$ parameters to optimise on the k^{th} iteration. The algorithm is as follows:



Compactness.



Generalisation Ability.



Specificity.

Figure 7.4. The compactness, generalisation and specificity for the equally-spaced, manual and GA models of the Hand data. M is the number of modes used.

- Begin with a single free parameter, τ_{11}, for each example and set the remainder to 0.5 (i.e. the original parameterisation):

  $\phi(\tau_{11}(0.5(0.5, 0.5), 0.5(0.5, 0.5)))$.

  Use the optimisation algorithm to find the values of $\{\tau_{11}\}$ that minimise the objective function. The optimal values of $\{\tau_{11}\}$ are fixed for the remaining iterations.

- At each subsequent iteration, k, the parameters at the k^{th} level of the hierarchy are optimised. e.g., for $k = 2$:

  $\phi(\tau_{11}(\tau_{21}(0.5, 0.5), \tau_{22}(0.5, 0.5)))$

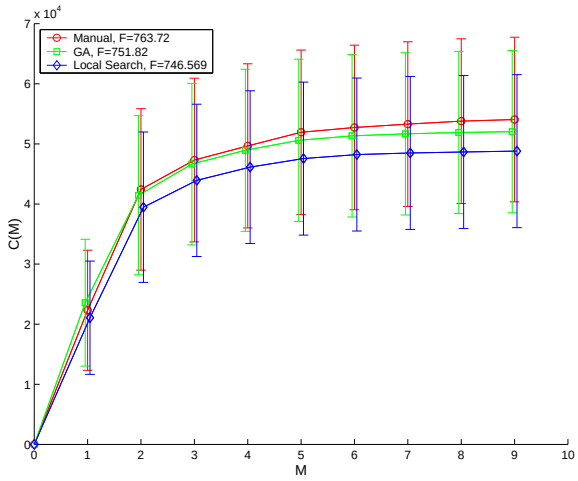
For the hand data, this algorithm takes 6172 objective function evaluations to converge, which is substantially less than the number required by the genetic algorithm (108896).

If the first two modes are viewed by eye, there is no obvious difference to those produced by the GA but the quantitative results plotted in figure 7.5 show that the method produces models with superior specificity than both the GA and manual models and significantly better generalisation than the manual model. There is no statistically significant difference in the compactness measure.

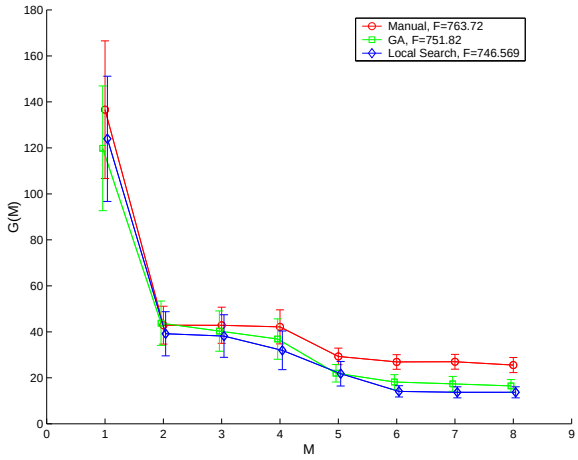
Although this optimisation method is substantially quicker and produces better models than the GA search, the scheme is not entirely satisfactory: its greedy nature means that once nodes are optimised, the correspondences they induce are fixed and cannot be altered. This may lead to sub-optimal solutions.

7.3 Summary

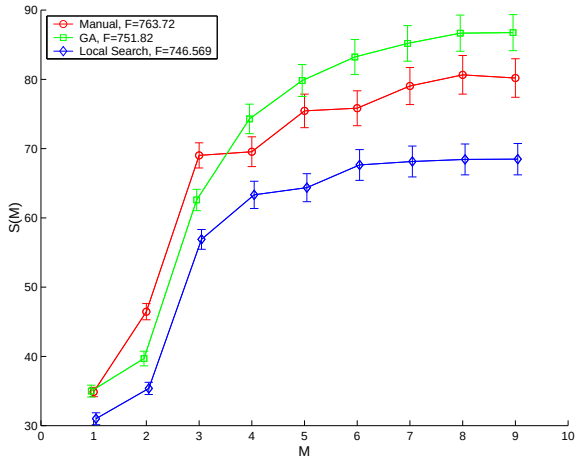
This chapter has described a piecewise linear representation of re-parameterisation and has shown how it can be manipulated to minimise the MDL objective function. Models for the Hand data that are better than the equally spaced model and as good as the manual models were produced automatically using two different optimisation methods. These experiments have established the feasibility of the approach outlined in chapter 5. There are, however, limitations: the piece-wise linear re-parameterisation scheme introduces discontinuities in the derivative of the re-parameterisation function which is likely to inhibit reliable convergence; the GA optimisation approach scales poorly to large training sets (and finds sub-optimal solutions). The local optimisation scheme is more efficient, but adopts a greedy strategy that is also likely to lead to sub-optimal solutions. These problems are addressed in subsequent chapters.



Compactness.



Generalisation Ability.



Specificity.

Figure 7.5. The compactness, generalisation and specificity for the local search, manual and GA models of the Hand data. M is the number of modes used.

Chapter 8

Building Models using an Improved Representation

"Success is the ability to go from failure to failure without losing your enthusiasm."
– *Winston Churchill.*

The previous chapter demonstrated the feasibility of the proposed approach to automatic model-building, but highlighted some weaknesses of the initial implementation. This chapter introduces a compact multi-resolution representation of re-parameterisation that is differentiable. This is coupled with an efficient and robust scheme for optimising the MDL objective function with respect to the re-parameterisations. The results are compared to those obtained earlier.

8.1 Representing Re-parameterisation

The piecewise-linear representation of re-parameterisation described in the previous chapter has the disadvantage that it is not differentiable and thus less than ideal for use in an optimisation framework. This section describes a new, differentiable representation of re-parameterisation. The main insight is that the monotonic reparameterisation function required for each 2D shape can be represented by the cumulative distribution function of some normalised, positive-definite density function $\rho(x)$:

$$\phi(u) = \int_0^u \rho(x) ds. \tag{8.1}$$

The following sections present a formulation of $\phi(u)$ for both open and closed curves. It is shown in chapter 11 that the method extends naturally to surfaces in 3D.

8.1.1 Open Curves

We choose to represent $\rho(x)$ as a sum of Cauchy distribution functions because there is an analytical form of the integral required to calculate the cumulative distribution function. The Cauchy kernel [67] is a unimodal, symmetric function of the form:

$$f(x) = \frac{1}{\pi}\frac{w}{w^2 + (x-a)^2}, \quad -\infty < x < \infty, \;\; w \geq 0, \tag{8.2}$$

where w is the width of the Cauchy and a is the position of the centre. The integral of the function is:

$$g(u) = \int_0^u dx f(x) = \frac{1}{\pi}\arctan\left(\frac{a-u}{w}\right). \tag{8.3}$$

To represent $\rho(x)$ for open curves $(S(0) \neq S(1))$, we take a sum of Cauchy kernels and a constant term:

$$\rho(x) = 1 + \sum_k \frac{A_k}{\pi}\frac{w_k}{w_k^2 + (x-a_k)^2}, \;\; 0 \leq x \leq 1, \tag{8.4}$$

where A_k is the magnitude of the k^{th} kernel. Any arbitrary distribution can be approximated to a given accuracy by a sum of Cauchies with different positions and widths. We can obtain $\phi(u)$ by taking its integral:

$$\begin{aligned} \phi(u) &= \int_0^u dx\rho(x), \quad 0 \leq x \leq 1 \\ &= u - \sum_k \frac{A_k}{\pi}\arctan\left(\frac{a_k-u}{w_k}\right). \end{aligned} \tag{8.5}$$

To normalise $\phi(u)$ to lie in the range [0,1], two normalisation constants, c and b, must be introduced:

$$\phi(u) = c\left[u + b - \sum_k \frac{A_k}{\pi}\arctan\left(\frac{a_k - u}{w_k}\right)\right], \tag{8.6}$$

where

$$\begin{aligned} b &= \sum_k \frac{A_k}{\pi}\arctan\left(\frac{a_k}{w_k}\right), \\ c^{-1} &= \left[1 + b - \sum_k \frac{A_k}{\pi}\arctan\left(\frac{a_k - 1}{w_k}\right)\right] + b. \end{aligned} \tag{8.7}$$

The constant term ensures that $\phi(u) = u$ when all A_ks are set to zero.

8.1.2 Closed Curves

For closed curves ($S(0) \equiv S(1)$), $\phi(u)$ must be a diffeomorphism of the unit circle. Although the formulation in the previous section could be used by treating closed curves as open curves that are 'joined' at ($u = 0, 1$), $\phi(u)$ would not, in general, be differentiable or smooth at this point. This can be addressed by 'wrapping' the Cauchy function around a circle [36]:

$$\begin{aligned} & f(\theta) = \frac{1}{2\pi}\frac{1-\omega^2}{1+\omega^2-2\omega\cos\theta}, \quad 0 \le \theta \le 2\pi \\ \text{where:} \quad & 0 \le \omega \le 1 \;\rightarrow\; \omega = e^{-w}, \;\; w \ge 0, \end{aligned} \tag{8.8}$$

and w is the width of the kernel. The kernel has a closed-form indefinite integral [67], which gives the cumulative distribution for a Cauchy centred at $\theta = 0$:

$$g(\theta, w, 0) = \frac{1}{2\pi}\arccos\left(\frac{(1+\omega^2)\cos\theta - 2\omega}{1+\omega^2-2\omega\cos\theta}\right), \;\; 0 \le \theta \le \pi, \;\; 0 \le g \le 1. \tag{8.9}$$

This can be re-written to be valid for ($0 \le \theta \le 2\pi$)[1]:

$$g(\theta, w, 0) = \frac{1}{2\pi}\arctan\left(\frac{(1-\omega^2)\sin\theta}{(1+\omega^2)\cos\theta - 2\omega}\right), \;\; 0 \le g \le 1, \tag{8.10}$$

where the inverse tangent is defined from 0 to 2π. From (8.10), the cumulative distribution for a Cauchy function centred at a_k is[1]:

$$g(\theta, \omega_k, a_k) = g(a_k, \omega_k, 0) + g(\theta - a_k, \omega_k, 0) - \frac{1}{2}[1 - \text{sign}(\theta - a_k)], \tag{8.11}$$

where:

$$\text{sign}(x) = \begin{cases} -1 & \text{if } x < 1, \\ 1 & \text{otherwise.} \end{cases} \tag{8.12}$$

To represent $\rho(\theta)$, we use a set of Cauchy kernels and a constant term:

$$\rho(\theta) = 1 + \sum_k \frac{A_k}{2\pi} \frac{1 - \omega^2}{1 + \omega^2 - 2\omega\cos\theta}, \quad 0 \leq \theta \leq 2\pi, \tag{8.13}$$

where A_k is the magnitude of the k^{th} kernel. We can obtain $\phi(u)$ by integrating $\rho(\theta)$:

$$\begin{aligned} \phi(u) &= \int_0^u dx \rho(\theta) \\ &= u + \sum_k A_k \, g(2\pi u, \omega_k, a_k), \quad 0 \leq \theta \leq 1. \end{aligned} \tag{8.14}$$

A normalisation constant, c is included to normalise $\phi(u)$ to lie in the range [0,1]:

$$\phi(u) = c \left[u + \sum_k A_k g(2\pi u, \omega_k, a_k) \right] \quad \text{where} \quad c^{-1} = 1 + \sum_k A_k. \tag{8.15}$$

Note that, for closed curves, the position of the origin, $\phi(0)$, is a free parameter.

8.1.3 Examples of Re-parameterisation

Figure 8.1.3 shows a monotonic function formed from a set of Cauchy kernels and demonstrates how such a set of functions can be used to re-parameterise a shape. The re-parameterisation can be manipulated by varying the coefficients of the Cauchy functions (i.e. A, w, a). Figure 8.2 shows the effect of varying the magnitude of a single kernel.

[1] Personal communication, Carole Twining, University of Manchester.

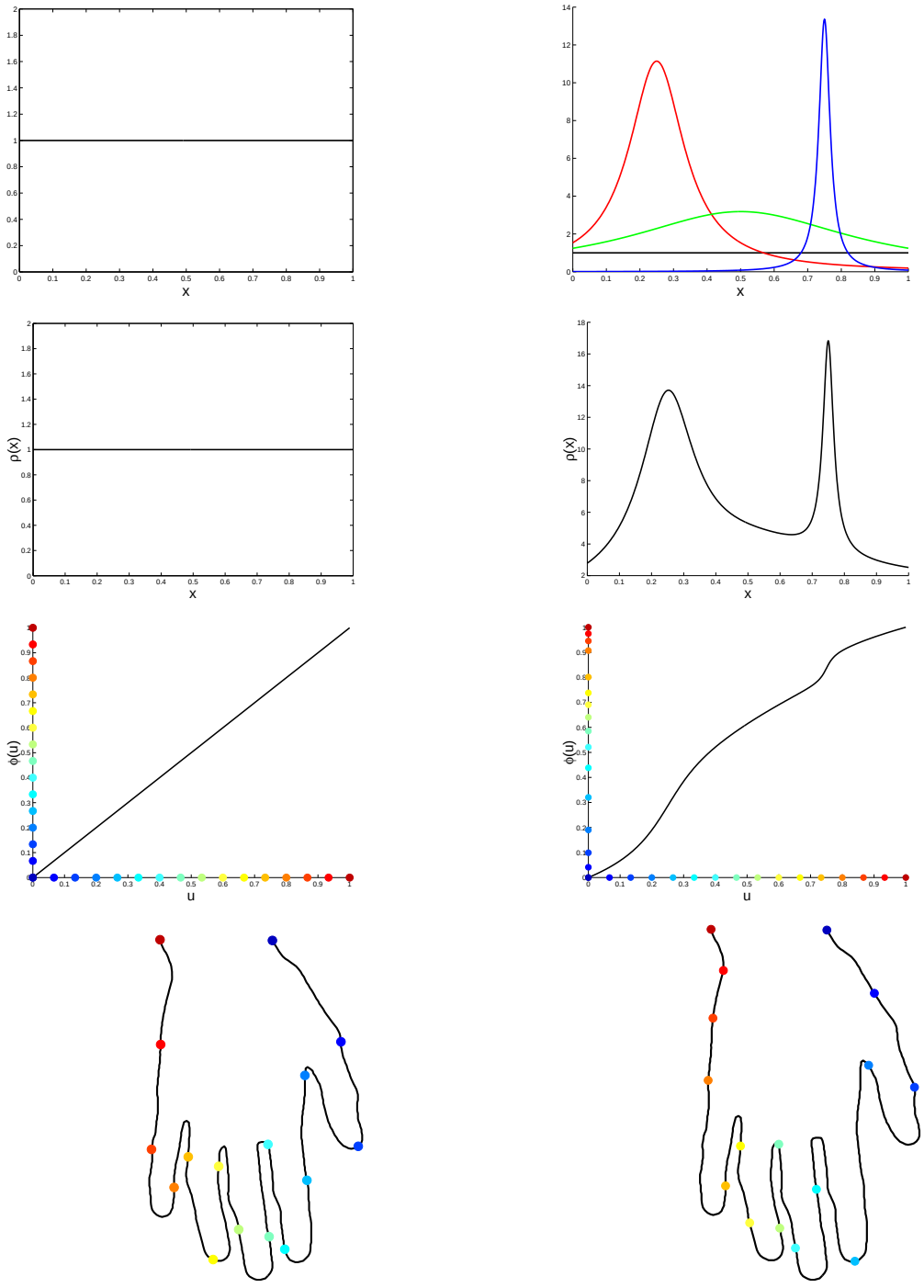


Figure 8.1. An example re-parameterisation. *Left Column*: The original parameterisation. *Right Column*: A re-parameterisation using Cauchy kernels with parameters $A = \{3.5, 4, 0.84\}$, $w = \{0.1, 0.4, 0.02\}$, $a = \{0.25, 0.5, 0.75\}$. *Top row*: The kernels; *second row*: $\rho(x)$, the sum of the kernels, *third row*: the parameterisation, $\phi(u)$; *bottom row*: the result of uniformly sampling the parameterisation on a hand outline.

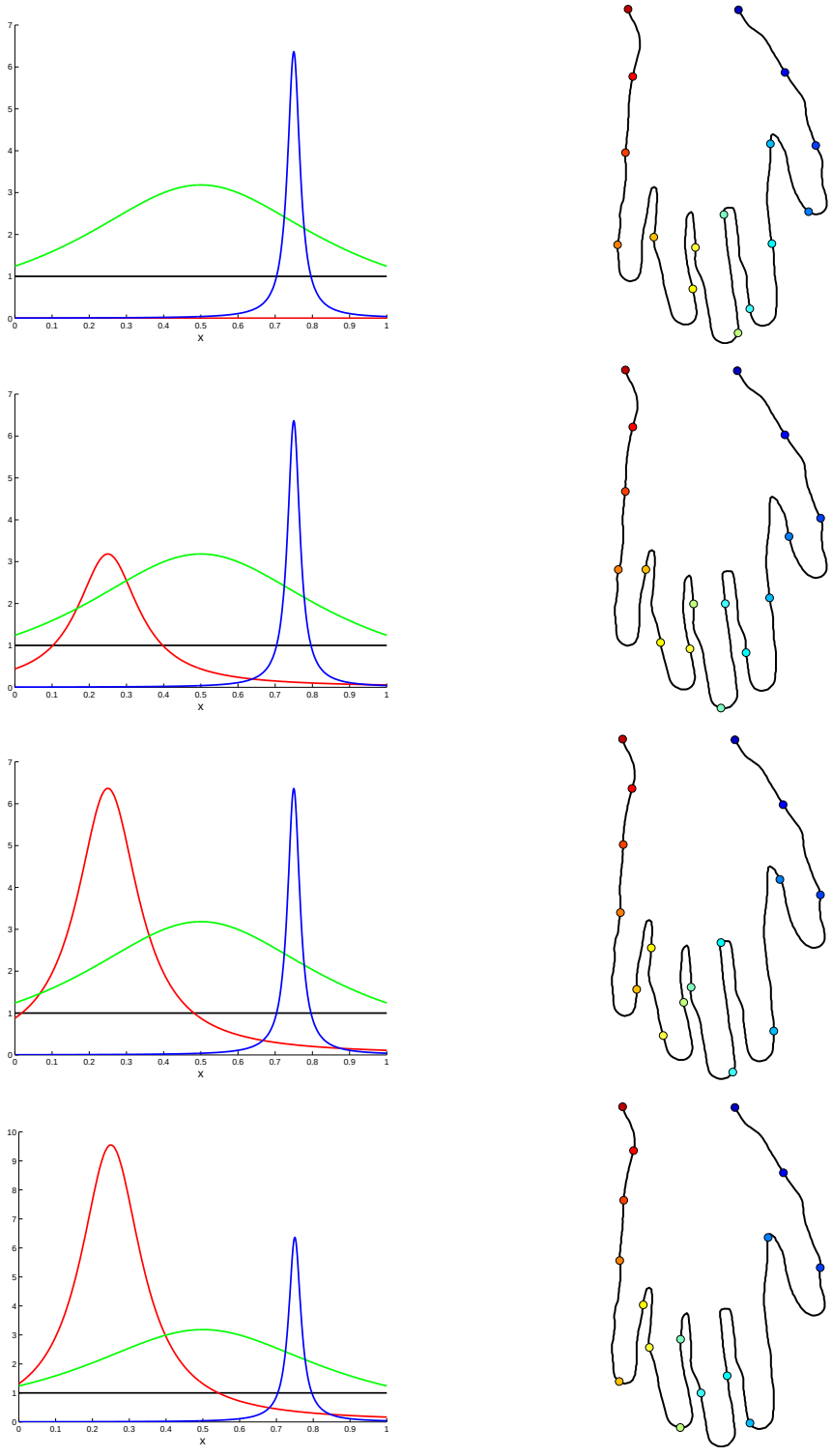


Figure 8.2. An illustration to show the affect of changing the height of a single Cauchy kernel. As the magnitude of the red kernel is increased, points are moved around the boundary.

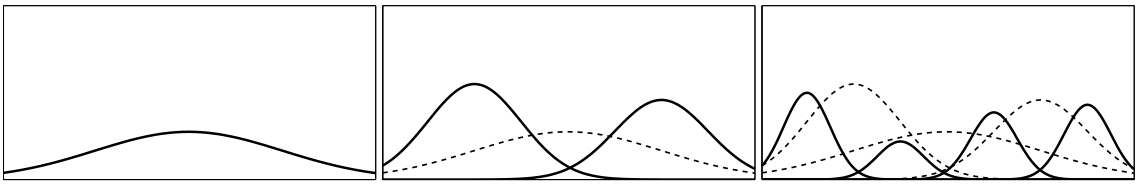


Figure 8.3. How a multi-resolution basis is formed: at each level of recursion, new kernels (solid lines) are introduced in-between the current ones (dashed lines). The widths of the new kernels are halved on each level of recursion.

8.1.4 Advantages of the Representation

The representation using Cauchy kernels allows the construction of arbitrary continuous, differentiable re-parameterisation functions with parameters $\{A_k, w_k, a_k\}$. Because kernels of variable width are used, the method naturally supports a coarse-to-fine representation of re-parameterisation, which proves useful during optimisation. In the limit of zero width the kernels become Dirac delta functions which, if there are infinitely many of them, can clearly represent any given re-parameterisation function. In practice, any desired re-parameterisation can be represented to a given accuracy by choosing a finite number of kernels with appropriate widths and positions.

Other functions could have been used to represent $\rho(\theta)$ (e.g. Von Misses, Cardioid, Gaussian [67]). The Cauchy kernel was chosen because it has a relatively simple integral. Davies *et al.* describe a similar representation of re-parameterisation based on a set of Gaussian kernels [29, 28]. To calculate the integral of a Gaussian, however, requires the evaluation of the error function, which is computationally expensive. The Cauchy-based formulation presented here can be evaluated in significantly less time.

8.2 A Multi-resolution Approach to Optimisation

The ability to build a coarse-to-fine representation of reparameterisation using kernels of different widths allows a multi-resolution approach to optimising the objective function. This is similar in spirit to the recursive splitting method with local optimisation described in the previous chapter, but without the disadvantage of fixing some correspondences prematurely. The basic idea is to begin with a coarse representation of re-parameterisation using a single wide kernel which is successively refined by recursively introducing narrower ones in-between, as illustrated in figure 8.3.

The optimisation algorithm is as follows:

- For each re-parameterisation function ϕ_i, begin with a single kernel of width w_1, centred at $a_1 = \frac{1}{2}$. The height, A_{1i}, of the kernel on each example is initialised to zero (equivalent to the original parameterisation). A local optimisation algorithm (Nelder-Mead simplex algorithm is used again) is used to find the set of magnitudes $\{A_{1i}\}$ that minimise F.
- At each level of recursion, k, add an additional 2^{k-1} kernels of width $(\frac{w_1}{2})^k$. The new kernels are positioned halfway between the kernels added at the previous level of recursion. Local optimisation is used to find the best height for the set of new kernels.

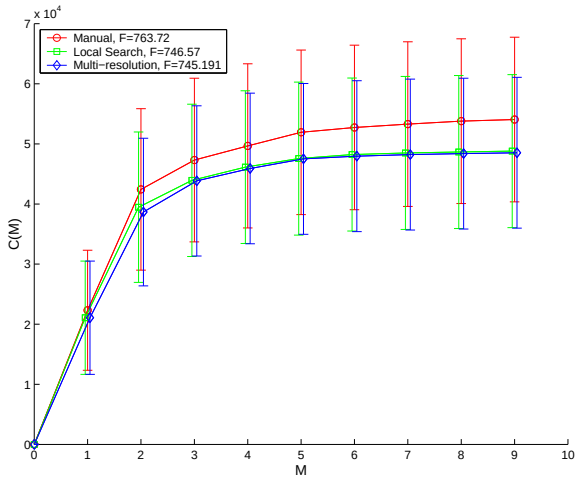
The convergence of the algorithm is quite robust to the selected value of w_1; a value of $w_1 = 0.25$ was used to obtain the results reported below.

8.3 Results

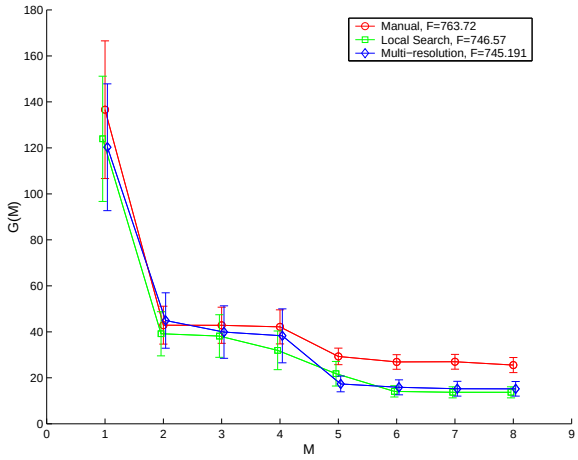
The method described above was applied to the Hand training set. The algorithm was run for three levels of recursion, giving the same number of free parameters as used for the local piecewise linear scheme described in the previous chapter. The results are shown in figure 8.4. Qualitatively, the modes of variation found are indistinguishable from those found previously. Although the optimisation finds a smaller value of F than the previous method, the specificity is worse. The algorithm requires 9145 evaluations of the objective function to converge, which is $\sim 50\%$ more than required for the piecewise linear scheme.

8.4 Conclusions

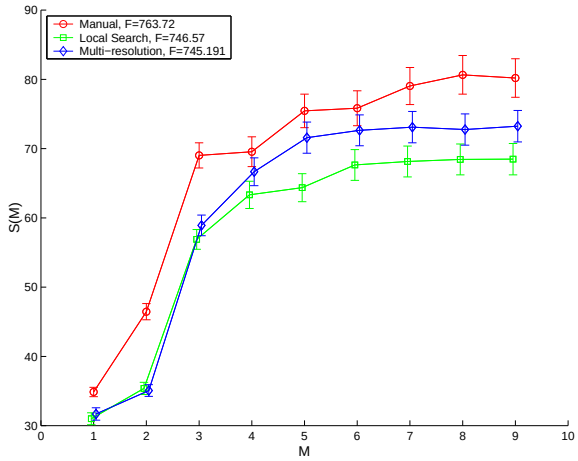
The methods introduced in this chapter provide a practical framework for automatic model building. The multi-resolution scheme for representing re-parameterisation using Cauchy kernels is well behaved and provides a basis for robustly optimising the MDL objective function. Although the results are worse than the previous method, we show how the speed and robustness of convergence can be improved in the following chapter.



Compactness.



Generalisation Ability.



Specificity.

Figure 8.4. The compactness, generalisation and specificity for the manual, local search (piecewise linear) and multi-resolution models of the hand data. M is the number of modes used.

Chapter 9

Refining the Optimisation Strategy

"An undefined problem has an infinite number of solutions. "
– *Robert A. Humphrey* .

The previous chapter showed how a kernel-based representation of re-parameterisation could be combined with a multi-resolution optimisation method to construct shape models automatically. The method produces good models, but it is not clear that it finds the global optimum or that the search is as efficient as possible. This chapter investigates several methods of improving the optimisation strategy. The final algorithm – described in section 9.5 – is capable of routinely constructing shape models directly from sets of training shapes. It is invariant to the starting point of the initial parameterisation of each shape and to the initial alignment of the training examples. All the experiments are performed using the Hand training set (§5.3.2), except when indicated otherwise.

9.1 A More Flexible Multi-scale Approach

In the multi-resolution optimisation scheme described in the previous chapter only a limited number of kernel widths and positions are used. Nevertheless, the number of parameters to be optimised simultaneously grows exponentially with the number of recursion levels and linearly with the number of shapes. Thus application of the method is limited to small training sets of relatively simple objects. Another disadvantage is that the scheme requires an optimisation schedule (number of recursion levels, iterations for each level, etc.). The optimisation may also fail to find the best solution, since the parameters of the wider kernels are fixed at an early stage in the process.

These problems can be overcome by selecting values for $\{a_k\}$ and $\{w_k\}$ stochastically and only optimising a single kernel for each example simultaneously giving $n_s - 1$ parameters to optimise at each iteration. The values for a are selected from a uniform distribution over the range $[0, 1]$. The widths of the kernels, w, are chosen from the positive half of a Gaussian distribution with zero mean and standard deviation σ_G. The kernel magnitudes, A, are once again used as the parameters of the optimisation.

The algorithm is as follows:

- **repeat**
  - randomly assign values $\{a_i\}$ and $\{w_i\}$ for all examples i except the reference example;
  - use the Simplex algorithm to find the values of $\{A_i\}$ that minimise F.
- **until** convergence.

The convergence of the algorithm is relatively insensitive to the value of σ_G. A value of $\sigma_G = \frac{1}{32}$ is used in the experiments reported below.

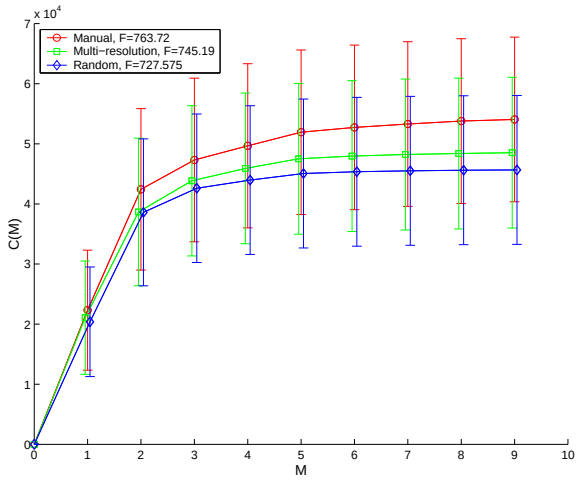
As an initial experiment, the algorithm was run until the objective function had been evaluated 9145 times. This is the same number of evaluations used in the multi-resolution scheme. The value of F was 740.12, an improvement when compared to 745.19 for the multi-resolution scheme.

The algorithm was then run to convergence. This took 106441 iterations. The quantitative results in figure 9.1 show that the method produces a model with significantly better generalisation ability and specificity than the manual and multi-resolution schemes.

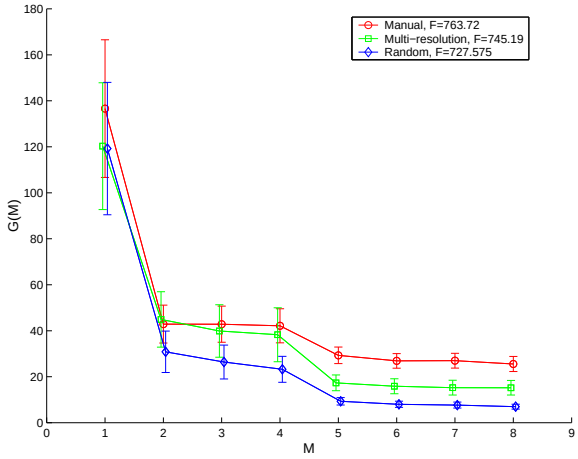
9.2 Manipulating the Origin

For closed curves, the position of the origin is a free parameter. In all previous experiments, this was fixed using a manually positioned landmark. This section investigates allowing the position of the origin to vary.

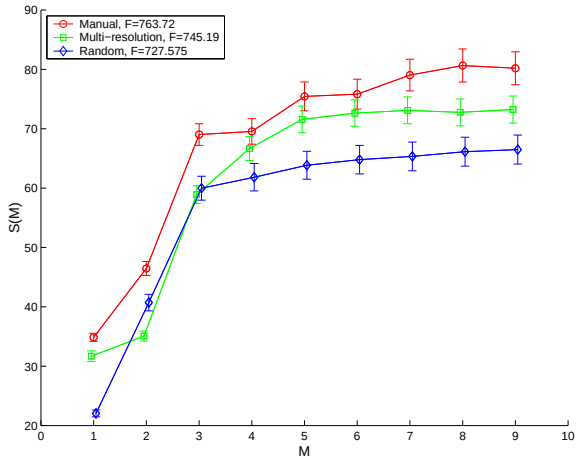
If the origin positions are included directly as parameters in the optimisation, they have a global effect that can disrupt any existing locally optimised correspondences. Instead, before each iteration, the position of the origin on each example is randomly placed along the length of the curve, with a uniform distribution, and fixed *for that iteration*, allowing every point on the boundary to move at some stage.



Compactness.



Generalisation Ability.



Specificity.

Figure 9.1. A quantitative comparison of the multi-resolution, manual and random scheme. The plots show the compactness, generalisation and specificity of the models as a function of M, the number of modes used.

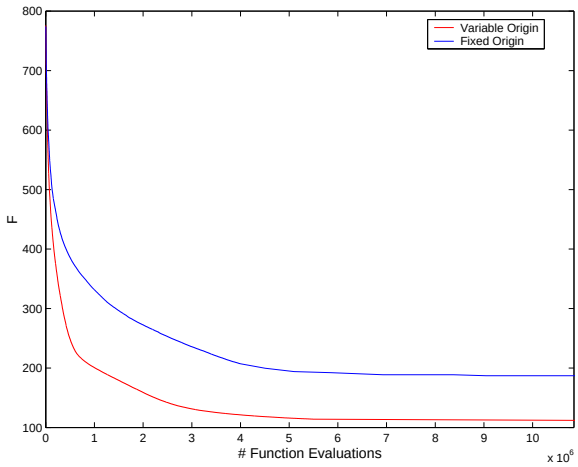


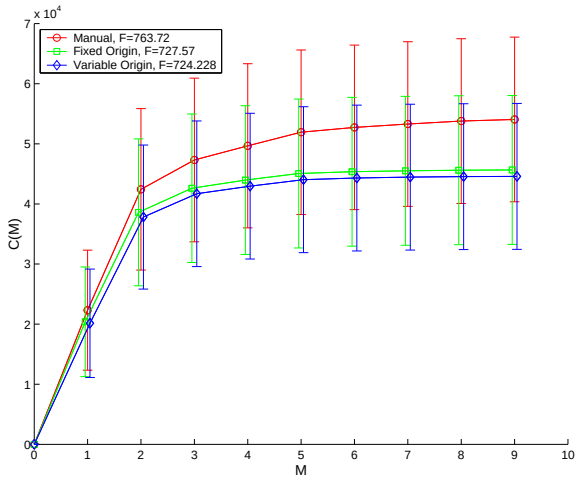
Figure 9.2. The convergence properties of two alternative optimisation schemes for the Bump data: fixing the origin and allowing the origin to vary.

The modified algorithm is:

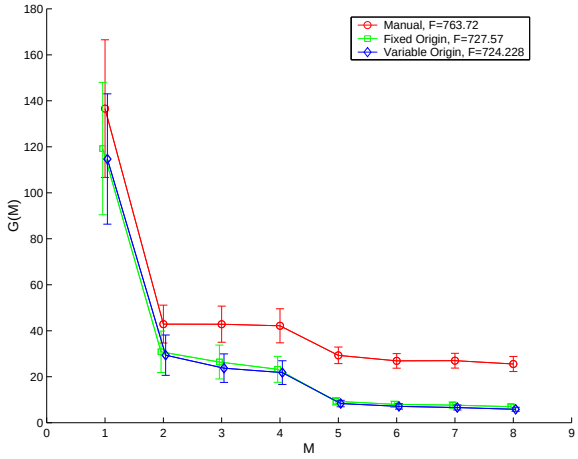
- **repeat**
  - randomly assign values $\{a_i\}$, $\{w_i\}$ and $\{\phi_i(0)\}$ for all examples i except the reference example;
  - use the Simplex algorithm to find the values of $\{A_i\}$ that minimise F.
- **until** convergence

The algorithm takes 332806 iterations to converge for the Hand data – almost three times as many as required when the origins are fixed. The quantitative results in figure 9.3 show that, overall, the resulting model is an improvement on the fixed origin scheme.

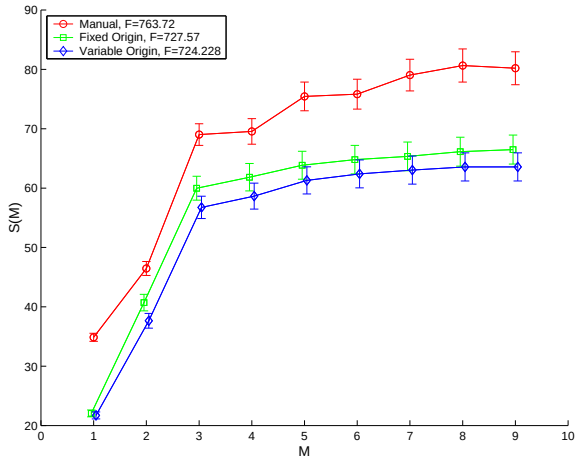
Although the improvement is relatively small for the Hand data, the global optimum for the Bump data described in section 5.3.2 can only be found if the origin is allowed to vary. This is demonstrated in figure 9.2 where the value of the objective function is plotted against the number of function evaluations. The scheme works because, if the origins are fixed, it is difficult for correspondences close to the origin to change significantly. The new scheme allows all correspondences to be modified freely whilst maintaining the localised behaviour of the optimisation process.



Compactness.



Generalisation Ability.



Specificity.

Figure 9.3. A quantitative comparison of the manual, fixed origin and variable origin schemes. The plots show the compactness, generalisation and specificity of the models as a function of M, the number of modes used.

9.3 Optimising One Example at a Time

All the optimisation methods proposed so far scale poorly with the number of training examples. For larger training sets ($\gtrsim 100$ examples) the number of parameters to be optimised simultaneously prevents the local optimisation algorithm from converging reliably. It is also not well suited to an iterative model-building scheme where examples are segmented and added one by one.

This section investigates the effect of optimising the parameterisation of one example at a time. This is achieved by cycling through the training set, optimising the current re-parameterisation of each example before moving on to the next iteration. Note that we are still considering the *entire* training set (i.e. the model is built using the current parameterisations of *all* examples) but the parameterisation of each example is optimised independently. To remove any bias, the ordering of the training set is permutated at random before each iteration. The algorithm is a direct extension of that described in the previous section and can be summarised as follows:

- **repeat**
  - randomise the ordering of the training set;
  - **for** each example, i (except the reference example):
    - randomly assign values to a_i, w_i and $\phi_i(0)$
    - use the Simplex algorithm to find the value of A_i that minimises F.
- **until** convergence.

As only one free parameter is optimised at a time, we could use a simpler optimisation algorithm (such as golden section search [77]). For the sake of consistency, however, we continue to use the Simplex method.

The quantitative results in figure 9.5 show that, for the Hand data, optimising one example at a time produces similar results to those produced by optimising all examples simultaneously. The algorithm takes 771866 function evaluations which is almost double the number required when optimising all examples concurrently.

The experiment was repeated using a set of 30 Hand examples. The convergence properties are illustrated in figure 9.4 where the value of the objective function is plotted against the number of function evaluations. The plots show that, for large numbers of training examples, optimising one example at a time is at least as efficient as optimising all examples simultaneously. The implementation of the method for very large training sets is much more tractable.

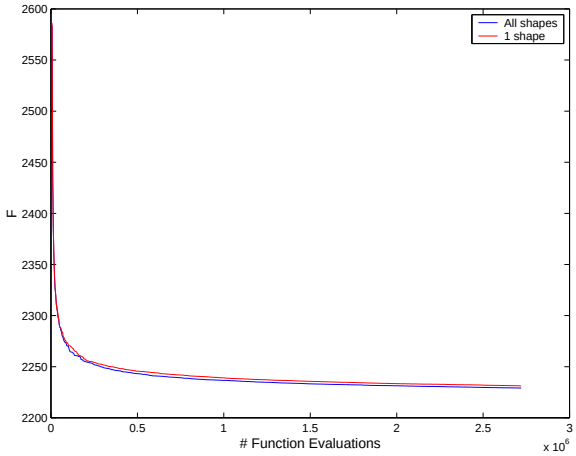


Figure 9.4. The convergence properties of two alternative optimisation schemes for the Hand data: optimising one example at a time and optimising all examples simultaneously.

9.4 Including Pose Parameters

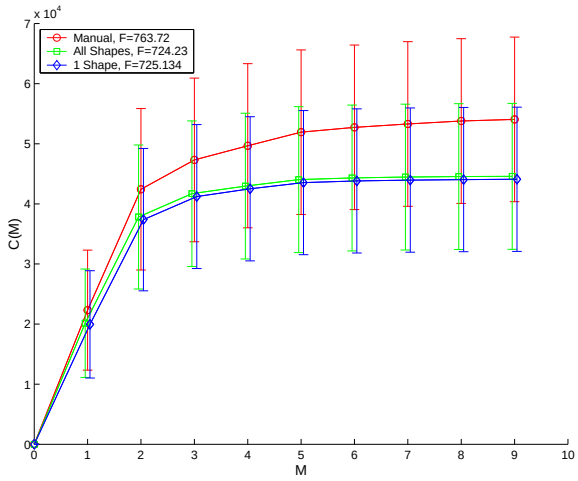
The positions of corresponding points depend on the rigid pose parameters of each example as well as the shape re-parameterisation. This is easily illustrated if we view the full expansion of the inner product defined earlier (3.14):

$$\mathbf{D}'_{ij} = \left((s_i \mathbf{R}_i S_i(\phi_i(\mathbf{u})) - \mathbf{d}_i - \bar{S}(\mathbf{u})) \cdot (s_j \mathbf{R}_i S_j(\phi_j(\mathbf{u})) - \mathbf{d}_j - \bar{S}(\mathbf{u}))\right), \tag{9.1}$$

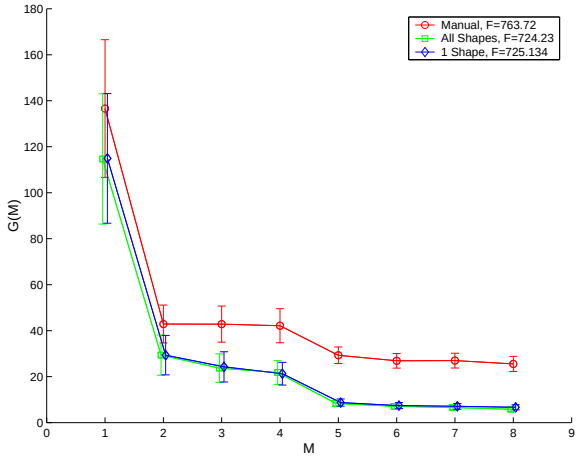
where S_i is a continuous, parameterised representation of the i^{th} shape, $\bar{S} = \frac{1}{n_s}\sum S_i$ is the mean shape, $\mathbf{d}$ is a translation, s is a rigid scaling and $\mathbf{R}$ is a rotation matrix.

All the results presented so far were obtained by performing Procrustes analysis [32] on the training set, which is equivalent to minimising the squared distance between each shape and the mean, then leaving the alignment fixed during optimisation. Since we have already seen that the mean squared error is not a good objective function (§6.1.1), it seems worthwhile to explore the effect of allowing the pose of each training example to vary during optimisation. It is important to note that the optimal pose parameters depend on the parameterisation of the shape and must therefore be included in the iterative process. In practice, translation can be dealt with directly by setting the centre of gravity of each re-parameterised shape to the origin. Scale and rotation must be included in the optimisation.

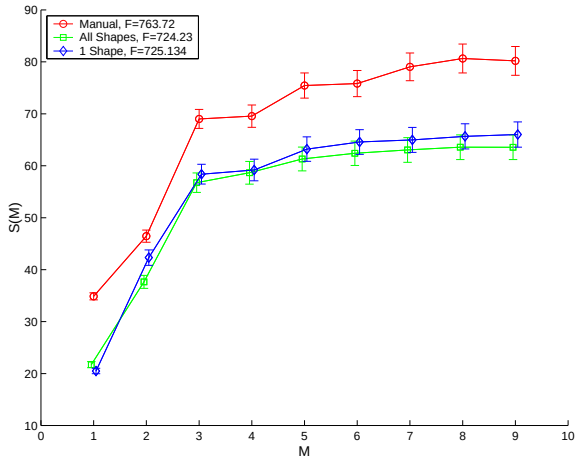
One way of minimising the objective function is to simply reduce the size of the training examples by a scaling, i.e. $F \rightarrow sF, s < 1$. To prevent the optimisation algorithm from doing this, the size of the mean should be constrained to have



Compactness.



Generalisation Ability.



Specificity.

Figure 9.5. A quantitative comparison of the manual model and models built by optimising one shape at a time and optimising all examples simultaneously. The plots show the compactness, generalisation and specificity of the models as a function of M, the number of modes used.

constant size; unfortunately this can only be achieved if all examples are optimised concurrently. Since we wish to optimise one example at a time we must use an alternative approach where the mean is restricted to varying by $\pm 5\%$ of its initial size.

The algorithm is as follows:

- **repeat**
  - Randomise the ordering of the training set
  - **for** each example, i (except the reference example):
    - randomly assign values to a_i, w_i and $\phi_i(0)$
    - use the Simplex algorithm to find the value of A_i that minimises F.
    - transform the re-parameterised pointset to the centre of gravity
    - use the Simplex algorithm to find the values of s_i and $\mathbf{R}_i$ that minimises F.
- **until** convergence

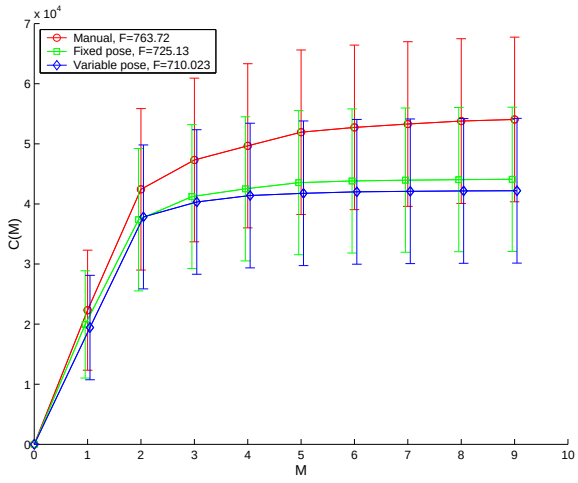
9.4.1 Results and Discussion

The algorithm takes 1000031 function evaluations to converge. Including pose parameters therefore slows down the convergence of the algorithm, but the results in figure 9.6 show that the resulting model has significantly better specificity properties than the model built using the previous scheme. The compactness and generalisation measures are also better, but the difference is not statistically significant.

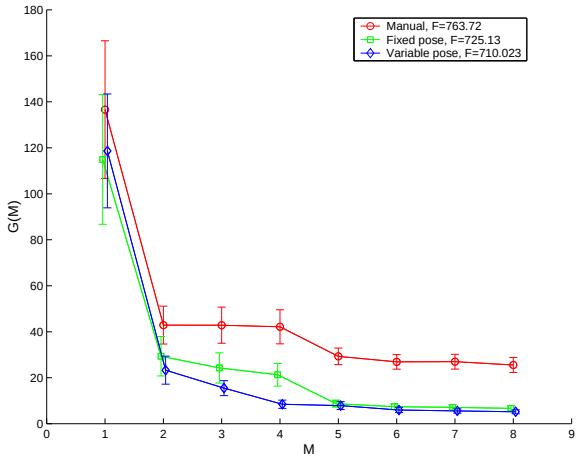
9.5 Summary of Method

This section provides a more complete summary of the algorithm that produces the best results. This is the method that is used to generate the results reported in the following chapter.

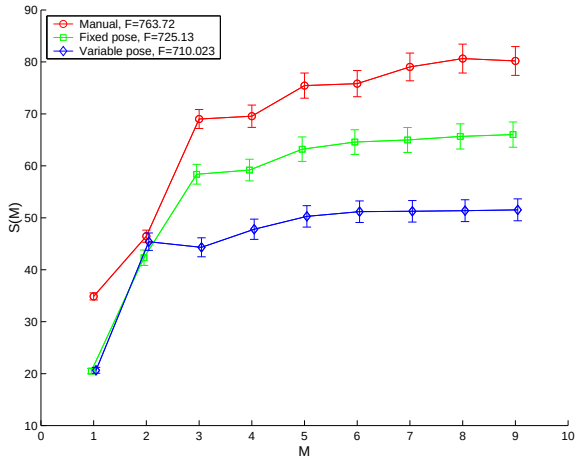
To obtain an initial estimate for the optimal values of $\{\phi_i, s_i, \mathbf{R}_i, \mathbf{d}_i\}$:



Compactness.



Generalisation Ability.



Specificity.

Figure 9.6. A quantitative comparison of the manual, fixed pose and variable pose optimisation schemes. The plots show the compactness, generalisation and specificity of the models as a function of M, the number of modes used.

- set $\phi(u) = u$ for all examples
- **for** each example, i (except the first):
  - **do**
    - select a position for the origin
    - use a Procrustes transformation to align it to the first shape
    - calculate the squared distance between the re-aligned shape and the first shape
  - **until** the squared distance to the first shape is minimised.

The following algorithm is used to optimise the value of F with respect to a set of re-parameterisation functions $\{\phi_i\}$ and pose parameters $\{s_i, \mathbf{R}_i\}$:

- **for** $i = 1 \ldots n_s$
  - $S'_i = s_i \mathbf{R}_i S_i(\phi_i)$
- build the model from $\{S'_i\}$
- evaluate the model using F

Given this, F can be optimised with respect to $\{\phi_i, s_i, \mathbf{R}_i\}$:

- **repeat**
  - randomise the ordering of the training set
  - **for** each example, i (except the reference example):
    - randomly assign values to a_i, w_i and $\phi_i(0)$
    - optimise the value of F wrt A_i using the Simplex algorithm
    - transform the re-parameterised pointset to the centre of gravity
    - optimise the value of F wrt s_i and $\mathbf{R}_i$ using the Simplex algorithm
- **until** convergence

9.6 Conclusions

This chapter has described an algorithm that can construct optimal statistical shape models directly from a set of pre-segmented example outlines. The algorithm needs no manual intervention, has no arbitrary parameters and can scale to cope with large training sets of examples. The following chapter employs the

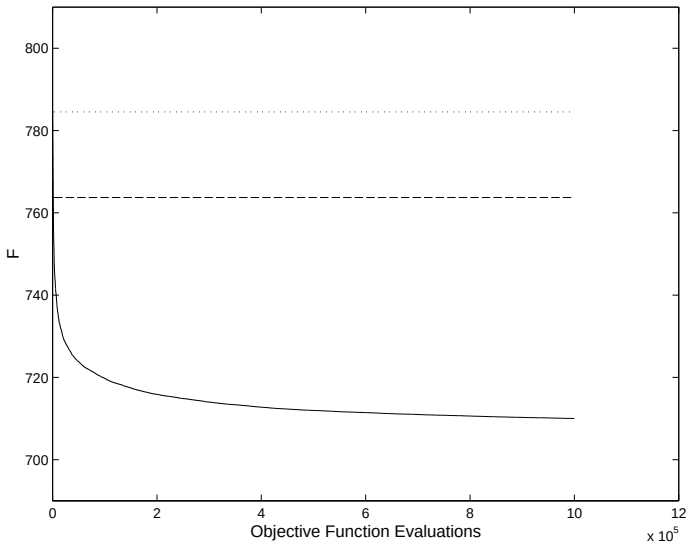


Figure 9.7. How the algorithm described in section 9.5 converges. The dashed line shows the objective function value of the manual model and the dotted line shows the value of the arc-length model.

algorithm described in §9.5 to automatically construct shape models of a number of biomedical objects.

All the optimisation methods reported in this chapter offer a significant improvement over the manually landmarked model – the current gold standard. Finding the optimum value of F, however, requires many function evaluations. The convergence properties of the algorithm are shown in figure 9.6 where the value of the objective function is plotted against the number of function evaluations. Although running the method until convergence takes over 1000000 objective function evaluations, the manual model is improved upon after only 1389 iterations. This suggests that satisfactory models could be obtained by stopping the optimisation prematurely.

We should also note that the termination criterion used is rather *ad hoc*. Due to the stochastic nature of the optimisation, halting the algorithm when the improvement is less than some given threshold is likely to result in premature convergence. In practice, we have chosen to terminate the algorithm manually by viewing a plot such as the one in figure 9.7. Limited emphasis should therefore be placed on the number of function evaluations required to reach convergence as this is likely to be an imprecise estimate.

Chapter 10

2D Results

"I do not fear computers, I fear the lack of them."
– Isaac Asimov.

The preceding chapters have described the development of a method for automatically constructing minimum description length (MDL) shape models. This chapter presents the results of applying the method to several 2D training sets. In each case the MDL model is compared to models obtained by equally spacing points and manual landmarking – the current gold standard. Using the objective criteria described in chapter 5, the MDL models are shown to be significantly better.

10.1 The 2D Experiments

For each training set, three different models were produced, each using the same number of boundary points. An MDL model was constructed using the algorithm described previously in §9.5. Manual models were built by annotating points on salient features (such as regions of high curvature) on each training outline and using path-length parameterisation to equally space points in between. Equally spaced models were constructed using a single manual landmark and equally spacing the rest around the boundaries. The manual and equally spaced models were aligned into a common co-ordinate frame using Procrustes analysis [32].

Quantitative results are reported by evaluating the models using the criteria described in §5.3 (generalisation, specificity and compactness) and by calculating

the MDL objective function value for each model. Qualitative results are presented by showing the first three modes of shape variation by independently varying the parameters by ± 2[standard deviations over the training shapes].

Results are reported for six training sets of pre-segmented 2D objects: 10 synthetic 'Bump' shapes, 33 heart ventricles, 17 hand outlines, 45 hip prostheses, 16 knee cartilages and 21 prostates. The details of the datasets with signposts to the compendium of results are set out below.

Synthetic Bumps

The first test set was the synthetic 'Bump' example introduced in chapter 5; the training set is shown in figure 10.1. The 'ideal' correspondences were labelled by manual landmarks as shown in figure 10.7a. Qualitative and quantitative comparisons of the resulting models are shown in figures 10.8 (page 99) and 10.14 (page 105) respectively.

Heart Ventricles

A set of 33 transcostal, long-axis echocardiograms of the left ventricle of the heart were acquired in a series of routine investigations and manually segmented by a cardiologist. The training set is shown in figure 10.2. The manual landmark model was created by annotating the points shown in figure 10.7b. Qualitative and quantitative comparisons of the resulting models are shown in figures 10.9 (page 100) and 10.15 (page 106) respectively.

Hands

The outlines of 17 hands were segmented from video images. Unlike the training set used in previous chapters, the outlines are represented as open curves requiring the representation of re-parameterisation described in 8.1.1 to be used. The training set is shown in figure 10.3 and the positions of the landmarks of the manual model are shown in figure 10.7c. Qualitative and quantitative comparisons of the resulting models are shown in figures 10.10 (page 101) and 10.16 (page 107) respectively.

Hip Prostheses

A set of 45 outlines of hip prostheses were segmented from clinical plane-film radiographs of subjects who had undergone total hip replacement [80]. The x-ray beam was centred on the symphysis pubis so that the radiograph captured the full pelvis and contralateral hip. The training set is shown in figure 10.4 and the positions of the landmarks used to build the manual model are shown in figure 10.7d. Qualitative and quantitative comparisons of the resulting models are shown in figures 10.11 (page 102) and 10.17 (page 108) respectively.

Knee Cartilage

3D images of the articular cartilage of the lateral femoral condyle were acquired from asymptomatic human subjects using T1-weighted MRI and segmented as described previously [92]. For each 3D image, a single sagittal slice was chosen from the centre of the lateral femoral condyle. As the width of the femur varies from subject to subject, comparable slices were identified by selecting the slices halfway between the first evidence of the lateral aspect of the meniscal horn and the full extent of the posterior cruciate ligament. The 15 outlines shown in figure 10.5 were used for this study. Only two significantly salient features exist on the outline; these were used as landmarks to build the manual model as shown in figure 10.7e. Qualitative and quantitative comparisons of the resulting models are shown in figures 10.12 (page 103) and 10.18 (page 109) respectively.

Prostate

Sixty ultrasound images were collected during routine ultrasound examinations of sixty different patients presenting symptoms of benign prostatic hyperplasia. The images were acquired using a 7.5 MHz transrectal ultrasound probe containing a mechanically spun transducer providing a 350^0 polar image around the probe. Images were acquired in transverse mode. Each image was captured at the mid-gland portion of the prostate – the probe position was chosen to be the image that contained the largest cross-sectional area of the prostate. 21 of the ultrasound images were randomly selected and segmented by a trained urologist who marked a number of points around the outline of the prostate. Spline interpolation was used to create a closed curve. The manual landmark model was created by annotating the outlines as shown in figure 10.7f. Qualitative and quantitative comparisons of the resulting models are shown in figures 10.13 (page 104) and 10.19 (page 110) respectively.

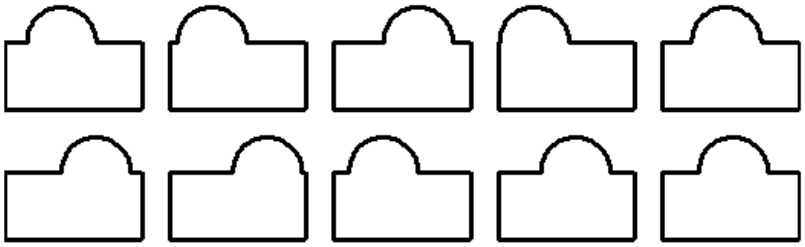


Figure 10.1. The 'Bump' training set.

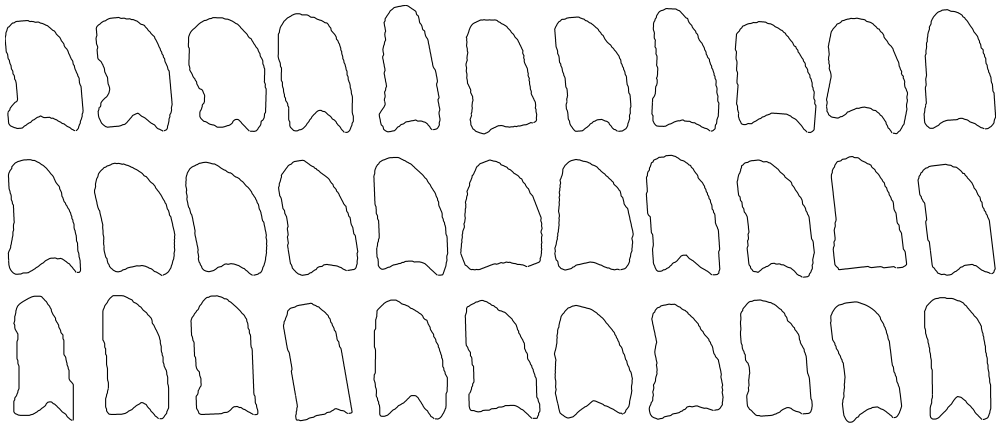


Figure 10.2. The heart ventricle training set.

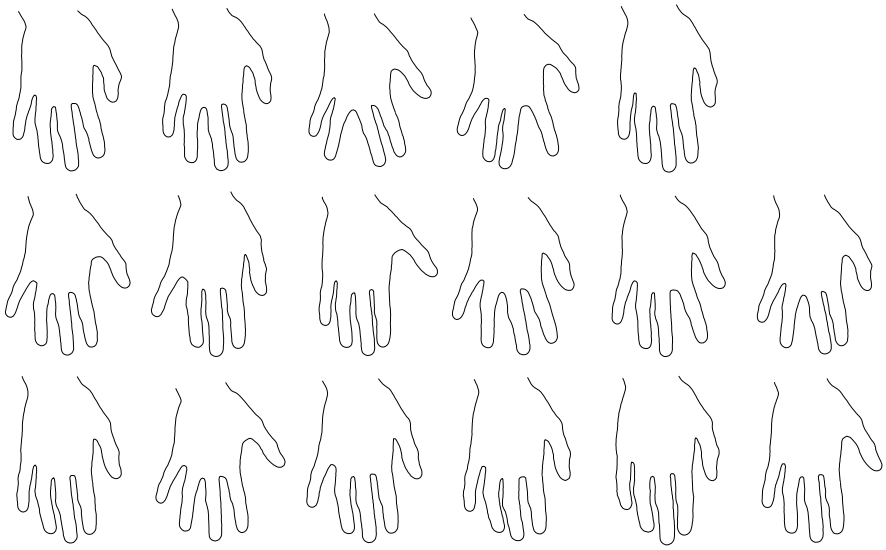


Figure 10.3. The Hand training set.

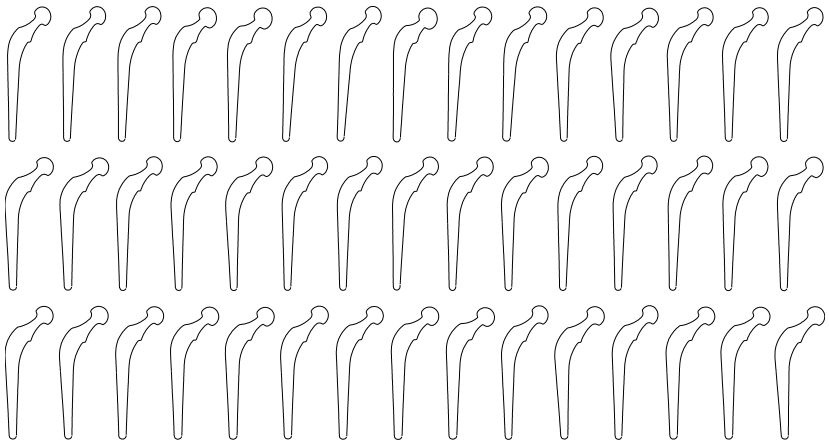


Figure 10.4. The hip prosthesis training set

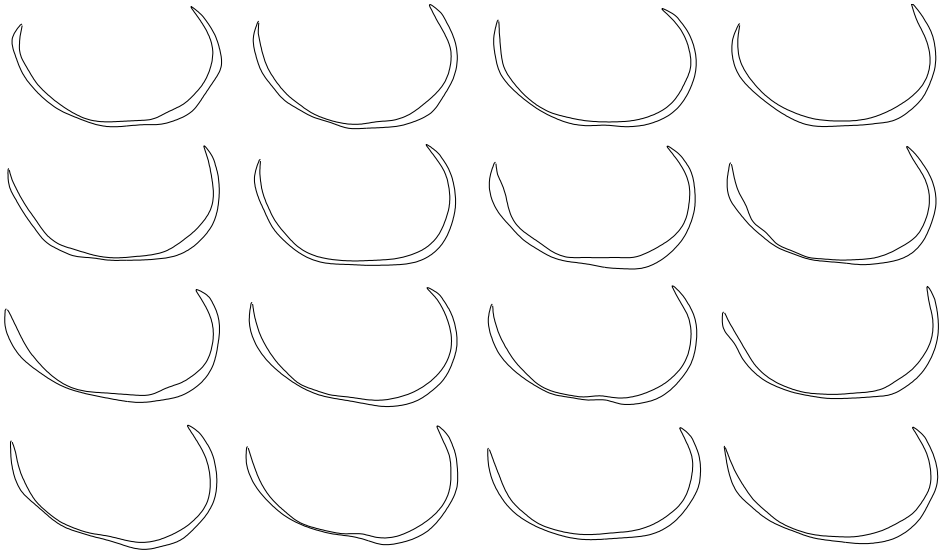


Figure 10.5. The knee cartilage training set

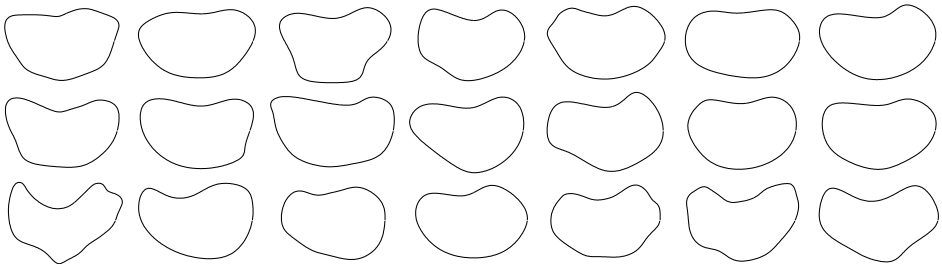
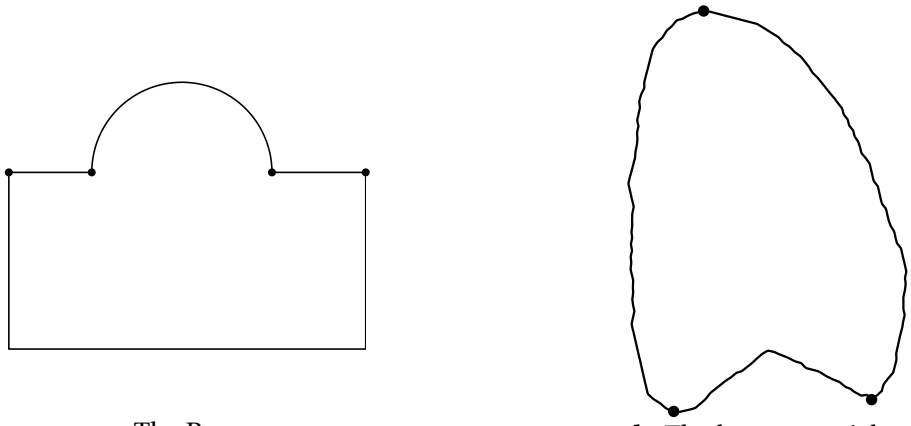
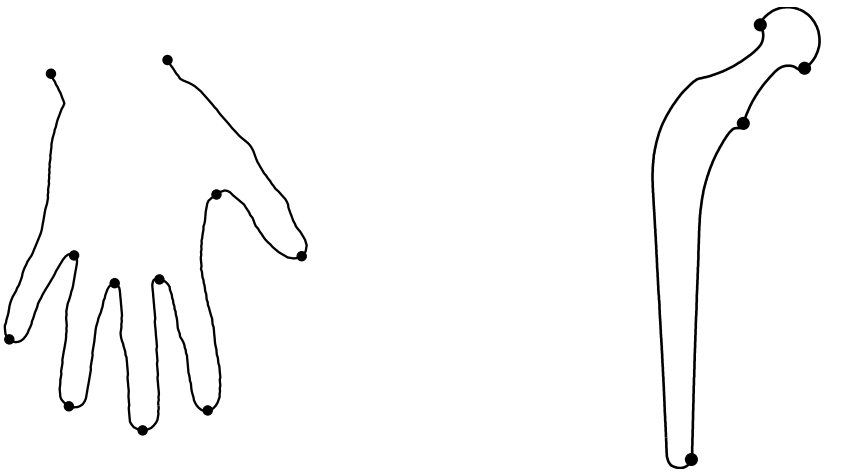


Figure 10.6. The prostate training set



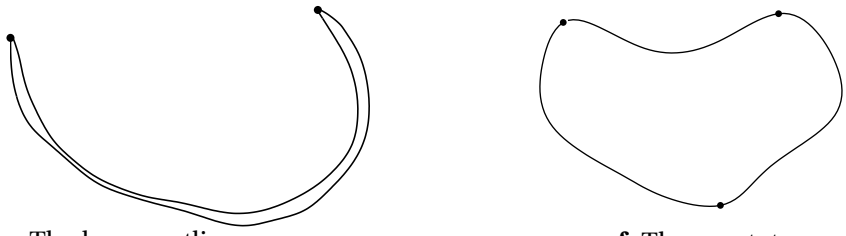
a: The Bump.

b: The heart ventricle.



c: The hand.

d: The hip prosthesis.



e: The knee cartliage.

f: The prostate.

Figure 10.7. The manual landmarks chosen for each dataset.

10.2 Discussion of Results

A very similar pattern of results is found for all the cases studied, though there are some exceptions that merit discussion. It is thus efficient to identify the general trends and note the exceptions.

Qualitative Results. Figures 10.8 to 10.13

Generally, the qualitative results demonstrate that it is possible to generate superficially plausible results using any of the three methods of establishing correspondences. For the Bumps (figure 10.8) and Hands (figure 10.10) it is clear that the equal spaced model is non-specific, whereas this is not obvious for all models. In all cases, the differences between the manual and MDL models are subtle but, as we shall see shortly, important.

Compactness. Figures 10.14a to 10.19a

For all but the Bump data (figure 10.14a), the MDL model is the most compact. The difference is significant for three of the datasets but not for the Hand (figure 10.16a) or Knee Cartilage (figure 10.18a) where the limited size of the training sets leads to large standard errors. As was already noted in §5.3.2 and §6.1.1, the Bump data illustrates the point that, although desirable, compactness alone is not, in general, a reliable measure of model utility. The manual models are more compact than the equally spaced models except for the hip prosthesis (figure 10.17a) and prostate data (figure 10.19a) where errors in the placement of manual landmarks are captured as statistical variation. For the prostates, there are no obvious stable features hence correspondences are difficult to landmark consistently. Although there are many distinct features on the hip prosthesis, the qualitative results show that the total variation is small hence positional errors introduced by manual landmarking swamp the genuine shape variation.

Generalisation ability Figures 10.14b to 10.19b

For all datasets, the MDL model has the best (or equal best) generalisation ability. In every case there is a statistically significant difference between the MDL model and at least one of the other models, but which is the worst model depends entirely on the dataset. For the ventricle, prostate and, arguably the knee cartilage datasets the MDL model is significantly better that both the equally spaced and manual models.

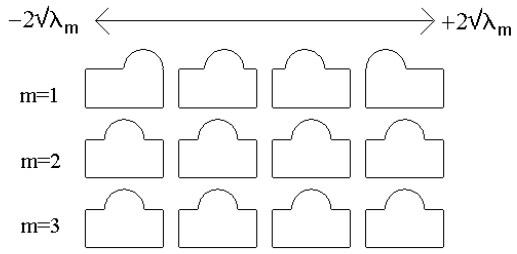
Specificity Figures 10.14c to 10.19c

The specificity results show that, even when it is not obvious from the qualitative results, there are very significant differences between the specificity of the models obtained using different methods. For all datasets, the MDL model has the

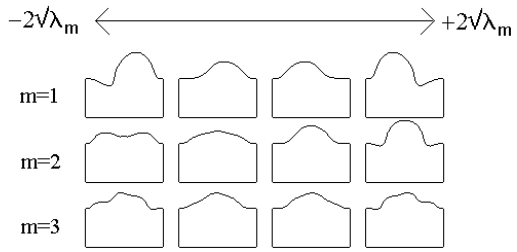
best (or equal best) specificity. There is a statistically significant difference between the MDL model and both of the others for all datasets expect the Hands, Knee Cartilage and Bumps where manual landmarking produces an equally good model. Manual landmarking is effective in these cases because there are obvious and consistent landmarks – such as the tips of the fingers on the hand outlines and the ideal correspondences for the Bumps.

General Observations

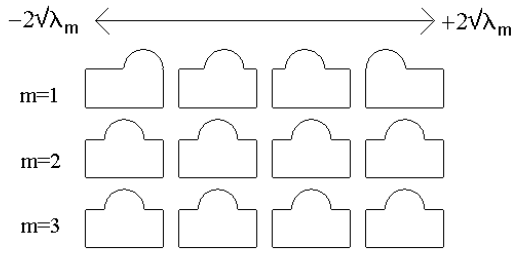
Overall, the MDL models are substantially better than the equally spaced and manual models in terms of specificity, generalisation ability and compactness. This is also reflected in the values of the objective function. Although manual landmarking produces plausible models when there are obvious features that are easy to annotate (such as on the hands and knee cartilage), equally spacing points leads to better models either when there is little variation (hence equally spaced points suffice) or where the shapes have no distinct features. Figure 10.20 shows examples of the correspondences found by the MDL method for the Hand data. This shows that the correspondences are very similar to what would have been obtained by manual landmarking.



a: MDL model.

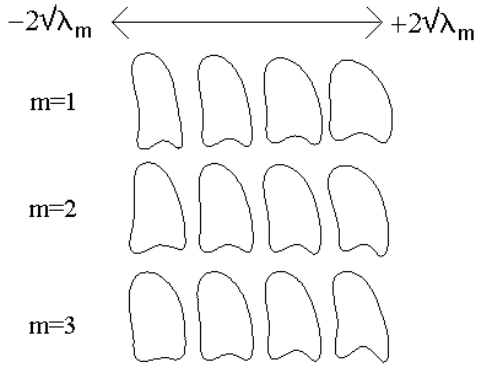


b: Equally spaced model.

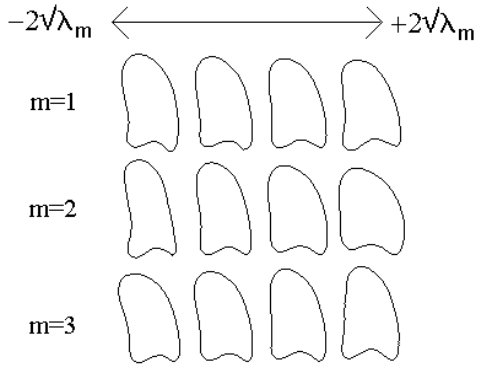


c: Manual model.

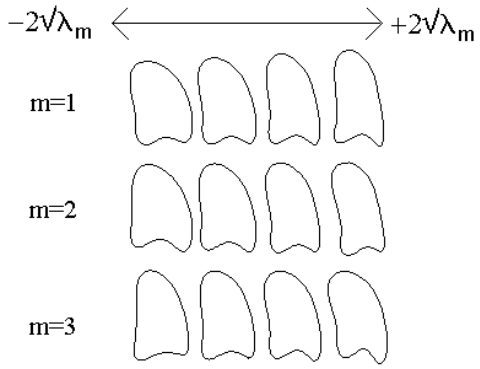
Figure 10.8. A qualitative comparison of the Bump models.



a: MDL model.

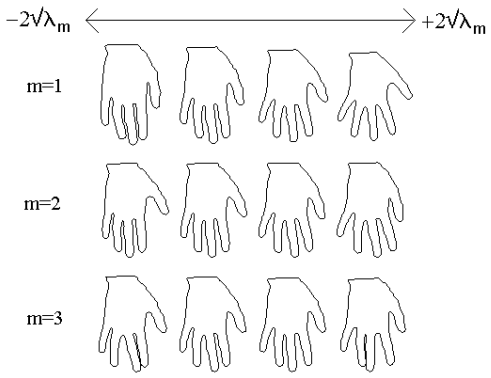


b: Equally spaced model.

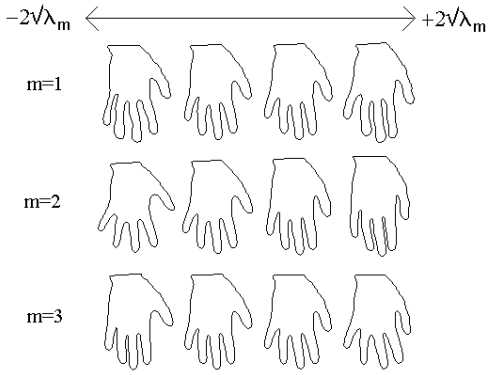


c: Manual model.

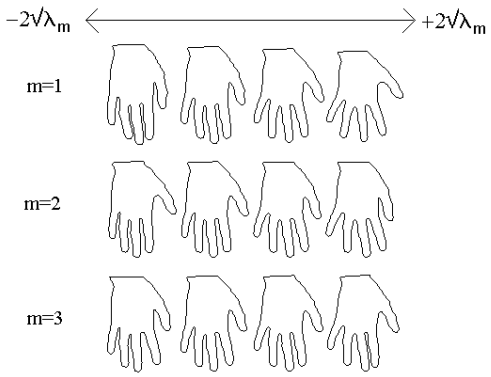
Figure 10.9. A qualitative comparison of the Heart Ventricle models.



a: MDL model.

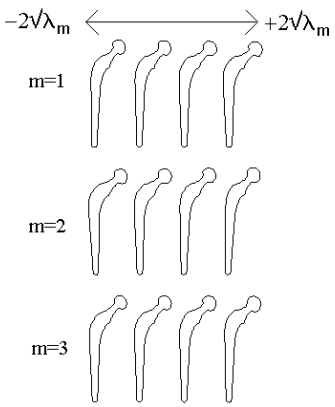


b: Equally spaced model.

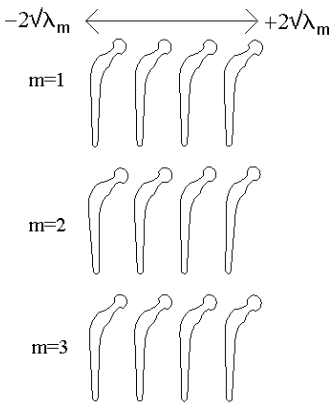


c: Manual model.

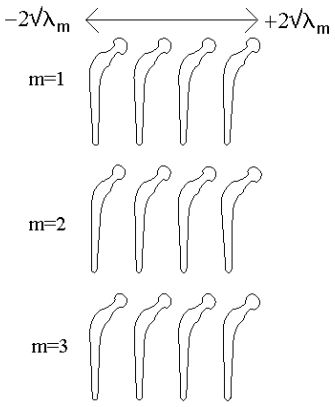
Figure 10.10. A qualitative comparison of the Hand models.



a: MDL model.

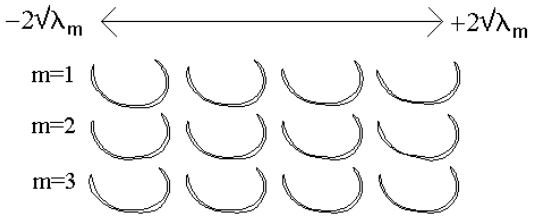


b: Equally spaced model.

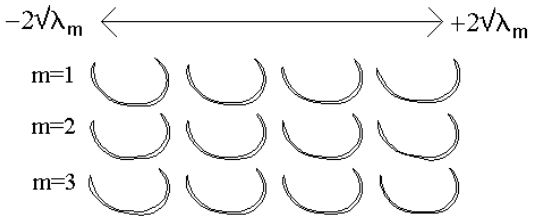


c: Manual model.

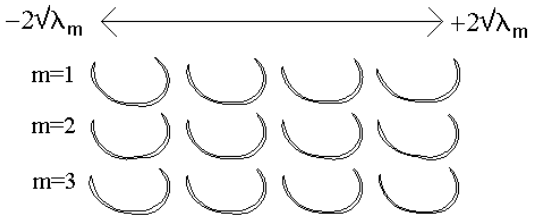
Figure 10.11. A qualitative comparison of the Hip Prosthesis models.



a: MDL model.

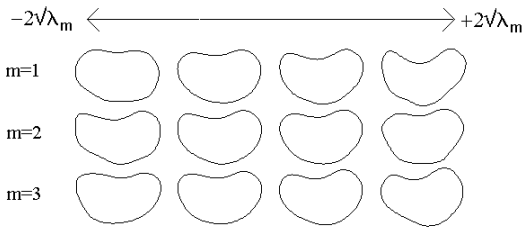


b: Equally spaced model.

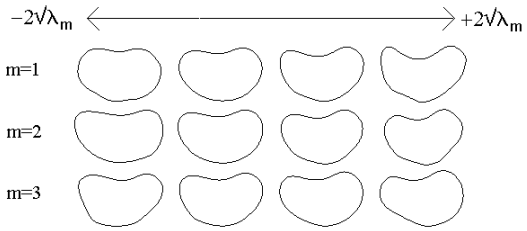


c: Manual model.

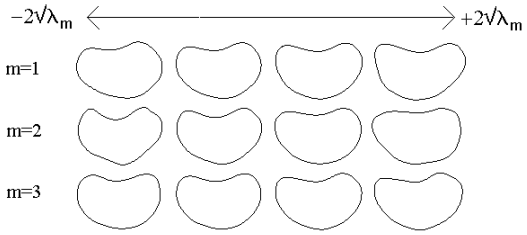
Figure 10.12. A qualitative comparison of the Knee Cartilage models.



a: MDL model.

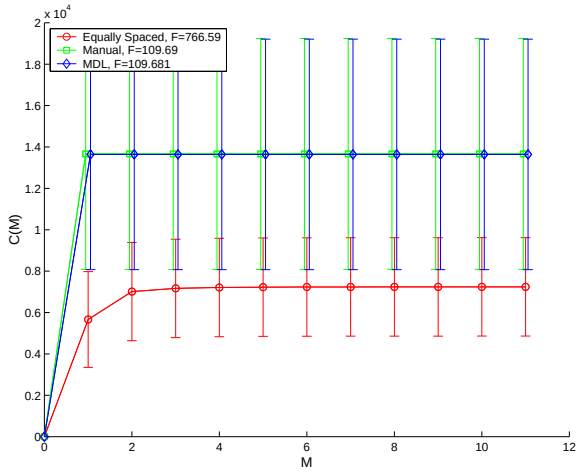


b: Equally spaced model.

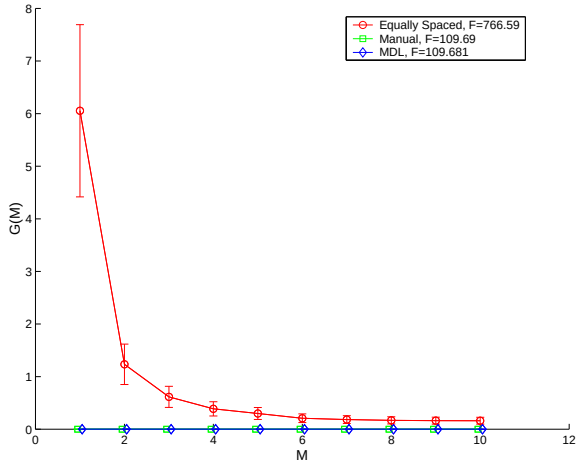


c: Manual model.

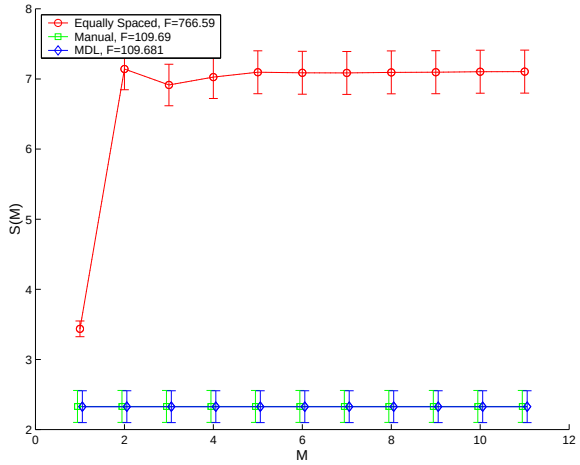
Figure 10.13. A qualitative comparison of the Prostate models.



a: Compactness.

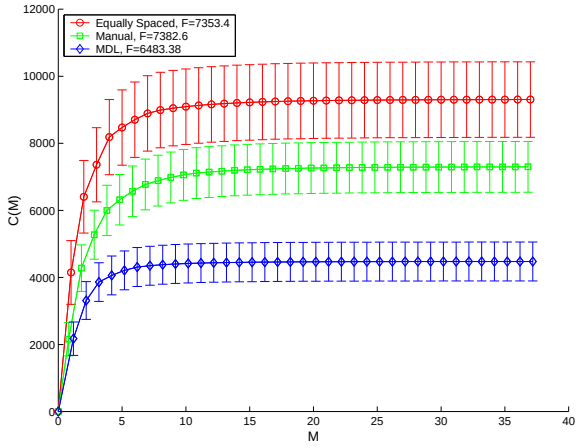


b: Generalisation ability.

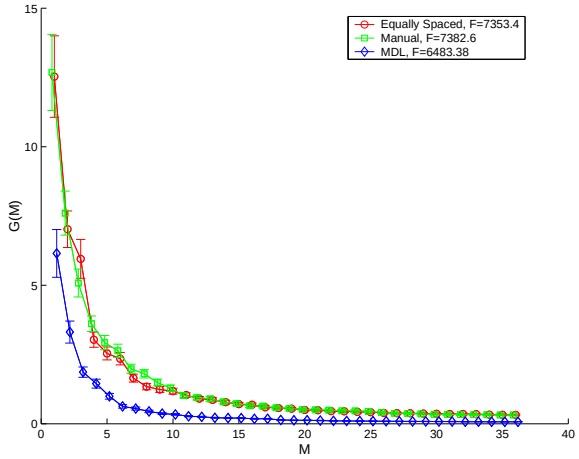


c: Specificity.

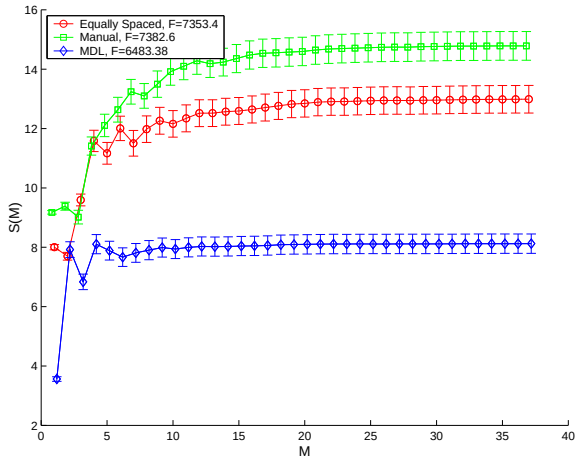
Figure 10.14. A quantitative comparison of the Bump models.



a: Compactness.

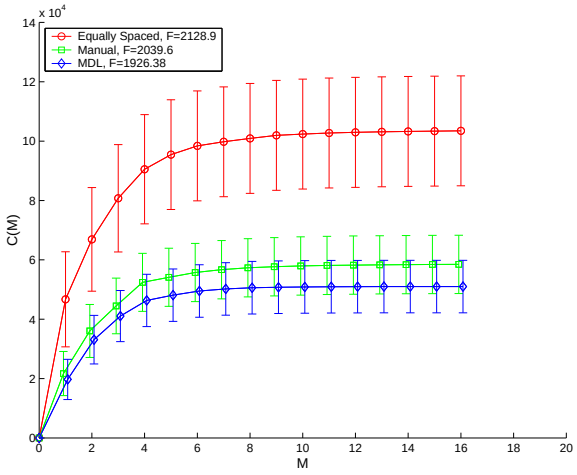


b: Generalisation ability.

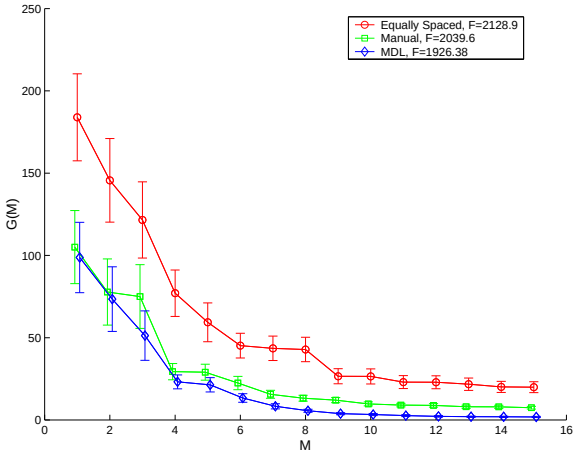


c: Specificity.

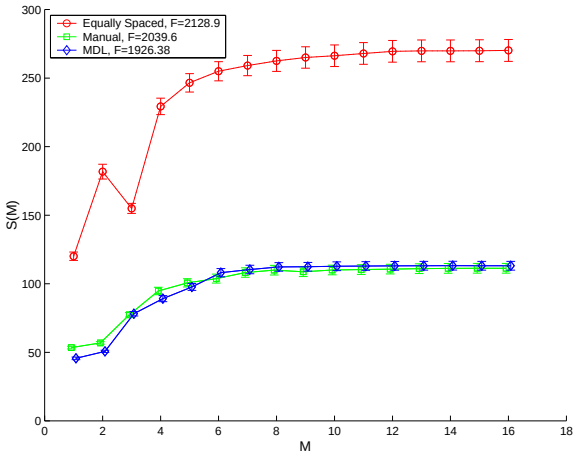
Figure 10.15. A quantitative comparison of the Heart Ventricle models.



a: Compactness.

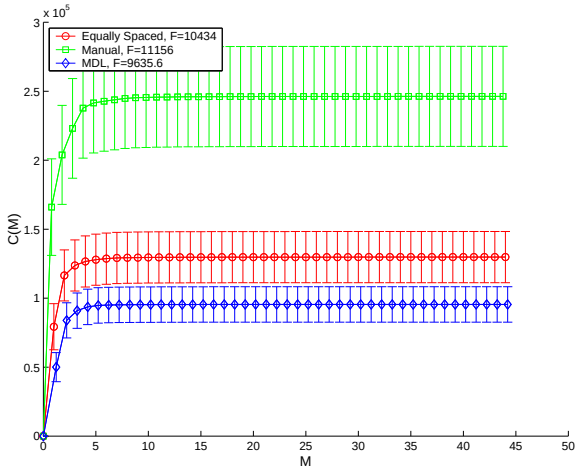


b: Generalisation ability.

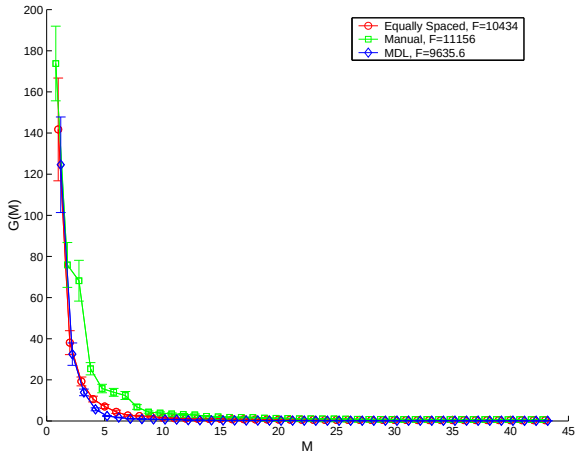


c: Specificity.

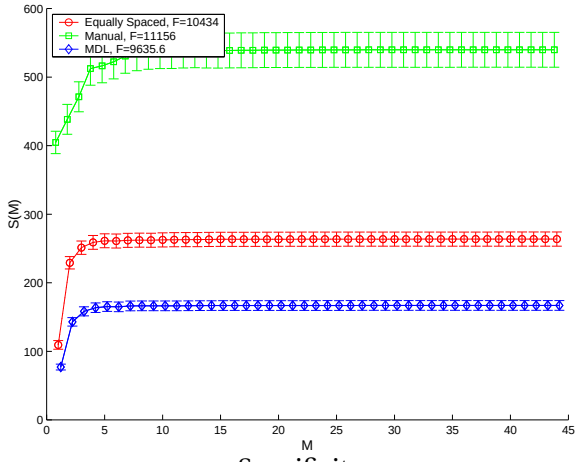
Figure 10.16. A quantitative comparison of the Hand models.



a: Compactness.

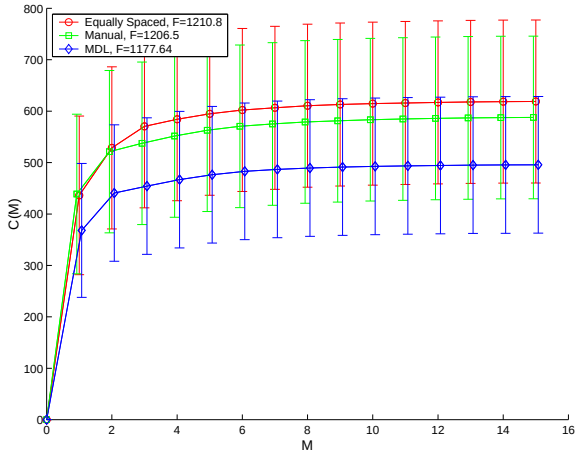


b: Generalisation ability.

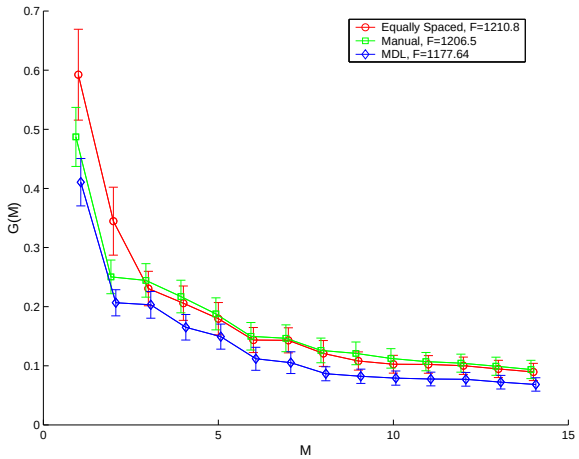


c: Specificity.

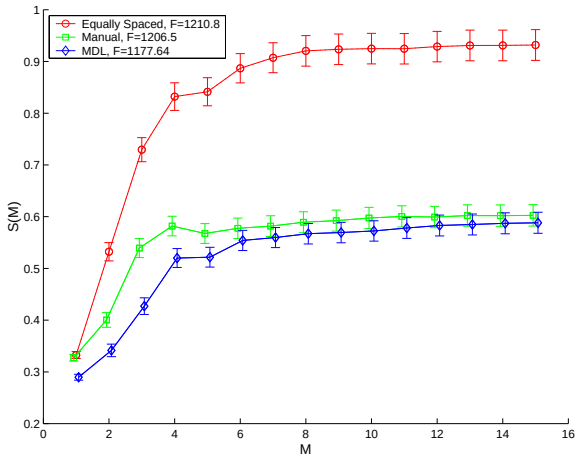
Figure 10.17. A quantitative comparison of the Hip Prosthesis models.



a: Compactness.

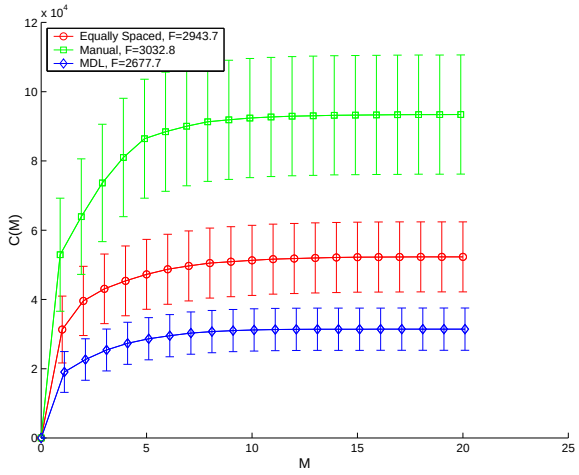


b: Generalisation ability.

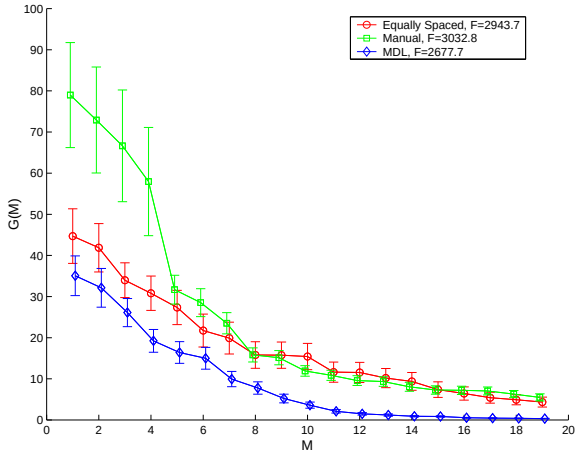


c: Specificity.

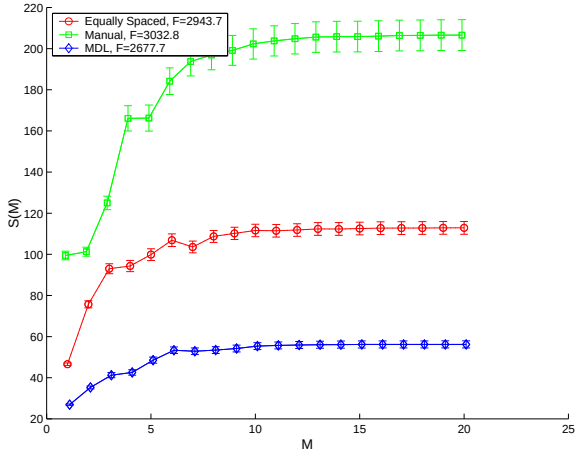
Figure 10.18. A quantitative comparison of the Knee Cartilage models.



a: Compactness.



b: Generalisation ability.



c: Specificity.

Figure 10.19. A quantitative comparison of the Prostate models.

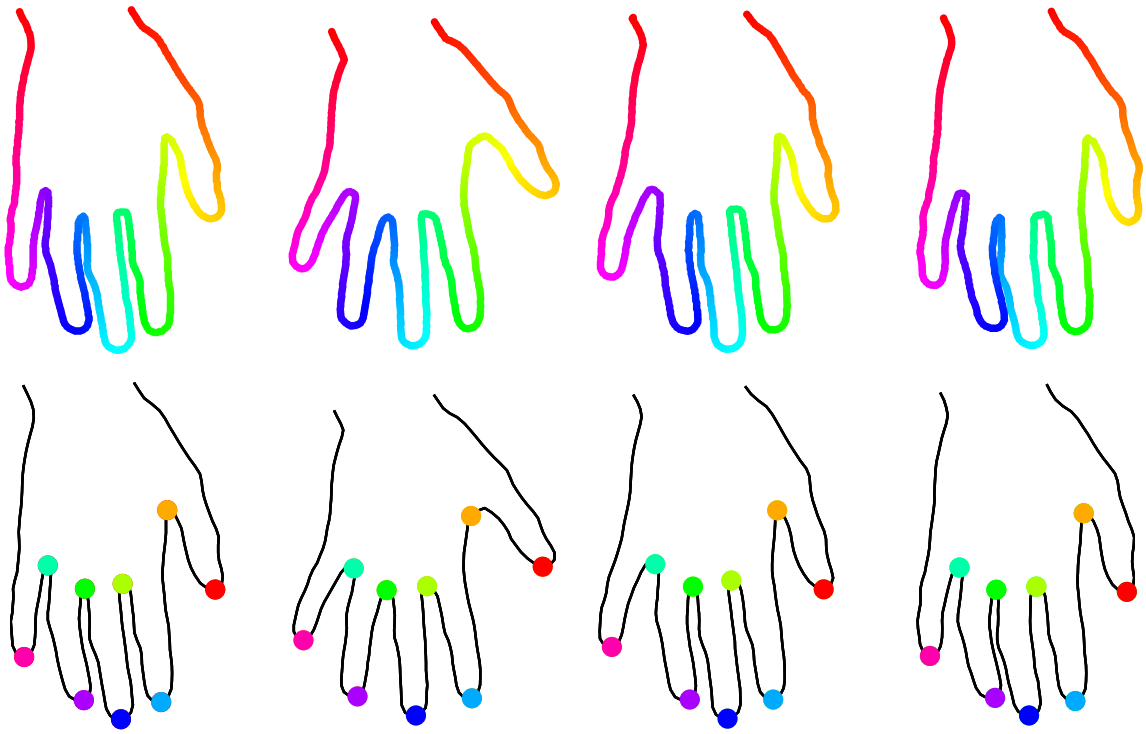


Figure 10.20. The correspondences found by the MDL method for the Hand data. Points of the same colur correspond. *Top*: a continuous correspondence. *Bottom*: the correspondence of salient points.

Chapter 11

Extension to 3D

"Everything should be as simple as it is, but not simpler. "
– Albert Einstein.

This chapter shows how the 2D methods presented in previous chapters can be extended to 3D. The approach is similar, but there is an additional step since the initial parameterisation, which was straightforward in 2D, is non-trivial in 3D. The problem is solved by mapping each training shape onto the surface of a unit sphere. Given an initial parameterisation, we show how both the piecewise linear and Cauchy kernel representations of re-parameterisation can be extended to 3D though, following the experience gained in 2D, more emphasis is placed on the kernel-based approach. Finally, a practical optimisation scheme is described, that is a direct extension of the 2D method described in chapter 9. The whole process is summarised diagrammatically in figure 11.1.

Figure 11.1. A schematic of the 3D model-building scheme. Ω_i is the mapping from the i^{th} training example to a sphere. ϕ_i is the re-parameterisation function of the i^{th} example. The original surfaces (in this case, n_s rat kidneys) are mapped to a sphere using $\{\Omega_i\}$. The spheres are then re-parameterised by $\{\phi_i\}$. A shape model is built from the sampled points after being projected back onto the surfaces using the inverse mappings $\{\Omega_i^{-1}\}$. The model is then evaluated (using the objective function) and the optimisation algorithm adjusts $\{\phi_i\}$ to produce a better model.

Figure 11.2. A typical training example of the anterior horn of a lateral ventricle. The example is represented by a stack of parallel contours, pre-segmented from a 3D MR image.

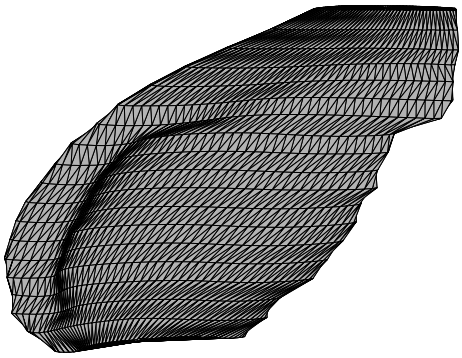


Figure 11.3. A simple triangulation of the brain ventricle data in figure 11.2.

11.1 Parameterising the Training Set

11.1.1 Building the Surfaces

In chapter 12 we present results for several different training sets. In each case the training data was a set of contour stacks manually segmented from parallel slices of 3D magnetic resonance (MR) images. Figure 11.2 shows a typical segmentation obtained from a 3D image of a brain. For all the examples we considered, the topology of each slice was the same (a single closed curve). This allowed the use of a simple triangulation algorithm to interpolate between the slices. If the topology had varied between slices, a general surface construction algorithm such as 'NUAGES' [39] could have been used (see [73] for a survey on surface construction methods).

A triangulated mesh was constructed by connecting neighbouring slices. Correspondence was established between the slices using arc-length parameterisation with the closest point of approach between the slices chosen as the origin. Corresponding points were sampled and connected to produce triangles as shown in figure 11.4. The resulting triangulated mesh was represented by a set of triangles $\{\mathbf{t}_k\}$ that index a set of vertices $\{\mathbf{v}_j\}$; $\mathbf{v}^a(\mathbf{t}_k), \mathbf{v}^b(\mathbf{t}_k), \mathbf{v}^c(\mathbf{t}_k)$ are the vertices of triangle $\mathbf{t}_k$. Figure 11.3 shows an example of running the triangulation method on the brain ventricle data shown in figure 11.2.

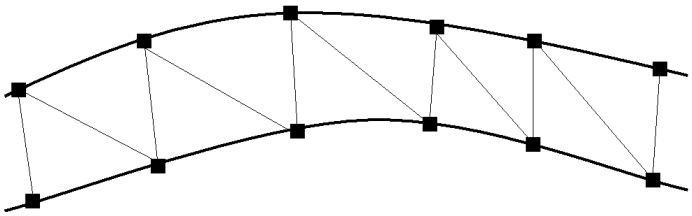


Figure 11.4. A simple triangulation of two parallel slices. The squares represent corresponding points.

11.1.2 Parameterising the Surfaces

We wish to represent each surface, S, in the training set parametrically. The classes of objects studied (and many others of practical interest) had spherical topology, so it is convenient to use a spherical polar parameterisation:

$$S(\mathbf{u}) = \begin{pmatrix} S_x(\mathbf{u}) \\ S_y(\mathbf{u}) \\ S_z(\mathbf{u}) \end{pmatrix}, \tag{11.1}$$

where $\mathbf{u} = (\theta, \psi)$ are spherical polar coordinates. Ideally, the parameterisation should be unique and isometric (distance-preserving); that is, 'moving' a distance d in parameter space (i.e. on the sphere) should produce a proportional displacement on the surface. In 2D, an arc-length parameterisation was used to produce an isometric mapping from the curve onto the line / circle. In 3D, finding a suitable parameterisation is more difficult and the subject of much research (e.g. [1, 52, 8]). Since we intend to find an optimal re-parameterisations for each example, the final result should not depend significantly on this initial parameterisation.

The problem of defining a parameterisation of a closed surface, S_i ($S_i \in \mathbb{R}^3$), can be posed as that of finding a continuous mapping, Ω_i, from the surface to the surface of a unit sphere ($\mathbb{S}^2$). For a triangulated mesh, $(\{\mathbf{v}_j\}, \{\mathbf{t}_k\})$, this can be split into two parts: assigning a parameter value $\mathbf{u}_j \in \mathbb{S}^2$ to each node $\mathbf{v}_j$, and interpolating the values of $\mathbf{u}$ in between the nodes.

Parameterising the Discrete Mesh

To find $\{\mathbf{u}_j\}$, we modified the first stage of the method described by Brechbühler *et al.* [8, 9]. Only a brief overview is given here, a more detailed description – including psuedocode – can be found in [8].

The method finds a diffeomorphic mapping, Ω^0, by solving Laplace's equation on

the surface:

$$\frac{\partial^2\theta}{\partial x^2} + \frac{\partial^2\theta}{\partial y^2} + \frac{\partial^2\theta}{\partial z^2} = 0,$$

$$\frac{\partial^2\psi}{\partial x^2} + \frac{\partial^2\psi}{\partial y^2} + \frac{\partial^2\psi}{\partial z^2} = 0, \tag{11.2}$$

where (x, y, z) are Cartesian coordinates and (θ, ψ) are spherical polar coordinates.

For a discrete mesh $(\{\mathbf{v}_j\}, \{\mathbf{t}_k\})$, (11.2) is satisfied when:

$$\mathbf{u}_j = \sum_{k\in\mathbb{N}} w_k \mathbf{u}_k \quad \sum_{k\in\mathbb{N}} w_k = 1, \tag{11.3}$$

where k indexes the connected neighbours of $\{\mathbf{v}_j\}$ and $\{w_k\}$ are a set of weights which are functions of the distance $|\mathbf{v}_j - \mathbf{v}_k|$. In our implementation these distances are all approximately equal so the weights, w_k, are assumed equal. (11.3) defines a set of linear equations, one for each node, that can be solved for $\{\mathbf{u}_j\}$. To obtain a unique solution boundary conditions are required.

The latitude values (θ) can be fixed by imposing the (Dirichlet) boundary conditions $\theta_{NP} = 0$ and $\theta_{SP} = \pi$, where NP is the node at the north pole and SP is the node at the south pole. We choose NP and SP to be the nodes in $\{\mathbf{v}_j\}$ with maximum and minimum z-coordinate value respectively. The longitude parameter (ψ) is more difficult as it is periodic such that $\psi(t) = \psi(t + 2\pi)$. We accommodate this by defining a 'dateline' where the value of ψ jumps from 2π to 0. Dijkstra's algorithm is used to define the dateline as the shortest path from NP to SP and the condition $\psi_k = 0$ is imposed for the path. Neighbours whose connections cross the 'date line' have 2π added to or subtracted from this value depending on the direction of the crossing.

As stated earlier, we would like the parameterisation to be an isometry (be distance preserving). For any surface with variable Gaussian curvature, finding an exact isometric mapping is impossible hence the resulting parameterisation will be distorted. Due to the nature of our surfaces, the method described above provides a parameterisation with very little distortion. If, however, there is much distortion, the optimisation step of Brechbühler [8, 9] or Quicken [78] may be used to provide a unique, area-preserving mapping.

11.1.3 Defining a Continuous Parameterisation

Assigning a parameter value $\mathbf{u}_j$ to every node $\mathbf{v}_j$ in triangulation $(\{\mathbf{v}_j\},\{\mathbf{t}_k\})$ creates a triangulation $(\{\mathbf{u}_j\},\{\mathbf{t}_k\})$ of identical connectivity on the unit sphere. Given any point $\mathbf{v}\in\mathbb{R}^3$ on surface S, we can compute its parameterisation $\mathbf{u}\in\mathbb{S}^2$ by interpolation. We define $(\alpha_k,\beta_k,\gamma_k)$, the barycentric co-ordinates of $\mathbf{v}$ with respect to triangle $\mathbf{t}_k$:

$$\begin{aligned}\alpha_k &= \frac{\text{Area}(\mathbf{v}\mathbf{v}^b(\mathbf{t}_k)\mathbf{v}^c(\mathbf{t}_k))}{\text{Area}(\mathbf{v}^a(\mathbf{t}_k)\mathbf{v}^b(\mathbf{t}_k)\mathbf{v}^c(\mathbf{t}_k))}\\ \beta_k &= \frac{\text{Area}(\mathbf{v}^a(\mathbf{t}_k)\mathbf{v}\mathbf{v}^c(\mathbf{t}_k))}{\text{Area}(\mathbf{v}^a(\mathbf{t}_k)\mathbf{v}^b(\mathbf{t}_k)\mathbf{v}^c(\mathbf{t}_k))}\\ \gamma_k &= \frac{\text{Area}(\mathbf{v}^a(\mathbf{t}_k)\mathbf{v}^b(\mathbf{t}_k)\mathbf{v})}{\text{Area}(\mathbf{v}^a(\mathbf{t}_k)\mathbf{v}^b(\mathbf{t}_k)\mathbf{v}^c(\mathbf{t}_k))},\end{aligned} \tag{11.4}$$

where $\text{Area}(ABC)$ is the area of triangle ABC.

Triangle $\mathbf{t}_k$ contains $\mathbf{v}$ if $\alpha_k\geq 0,\beta_k\geq 0,\gamma_k\geq 0$ and $\alpha_k+\beta_k+\gamma_k=1$. Having found the triangle that contains $\mathbf{v}$, the barycentric co-ordinates can be use to define the corresponding point $\mathbf{u}\in\mathbb{S}^2$ in the corresponding triangle on the sphere:

$$\mathbf{u}=\alpha_k\mathbf{u}^a(\mathbf{t}_k)+\beta_k\mathbf{u}^b(\mathbf{t}_k)+\gamma_k\mathbf{u}^c(\mathbf{t}_k)=\Omega(\mathbf{v}). \tag{11.5}$$

The inverse transformation Ω^{-1} can be defined analogously:

$$\mathbf{v}=\Omega^{-1}\mathbf{u}. \tag{11.6}$$

An example of a continuous parameterisation is shown in figure 11.5.

11.2 Representing Re-parameterisation

As in 2D, correspondence is manipulated by re-parameterising each training example, S_i, using a function ϕ_i:

$$S_i(\theta,\psi)\rightarrow S_i(\theta',\psi'),\;\;\theta'=\phi_i^{\theta}(\theta,\psi),\;\;\psi'=\phi_i^{\psi}(\theta,\psi). \tag{11.7}$$

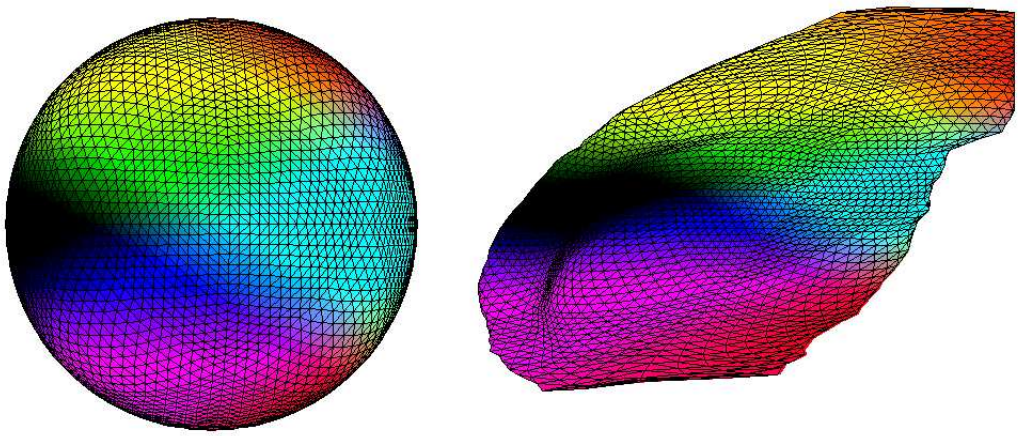


Figure 11.5. An example parameterisation of the anterior horn of a brain ventricle. The colours denote the parameter values on the sphere.

To ensure that the correspondences are valid each ϕ_i must be a diffeomorphism of a sphere. This can be achieved by extending to 3D either the piecewise linear or Cauchy kernel methods we defined for 2D shapes.

11.2.1 Piecewise-linear Re-parameterisation

Re-parameterisation can be achieved by extending the recursive, piecewise-linear representation of ϕ defined earlier in section 8.1.2.

The construction is initialised by selecting four nodes on the sphere; these form the initial mesh of four spherical triangles. At a given level of recursion, each spherical triangle in the mesh is subdivided into three smaller triangles by adding a new child node that is constrained to lie inside the spherical triangle defined by the three parent nodes. If $\mathbf{r}_a$, $\mathbf{r}_b$ and $\mathbf{r}_c$ are the position vectors (w.r.t. the centre of the sphere) of the parent nodes at the vertices of the spherical triangle ABC, a child node can be represented by:

$$\mathbf{r}_d = \alpha\mathbf{r}_a + \beta\mathbf{r}_b + \gamma\mathbf{r}_c, \tag{11.8}$$

see figure 11.6 for details.

The constraint that $\mathbf{r}_d$ lies within the *planar* triangle formed by $\mathbf{r}_a$, $\mathbf{r}_b$ and $\mathbf{r}_c$ can be satisfied by:

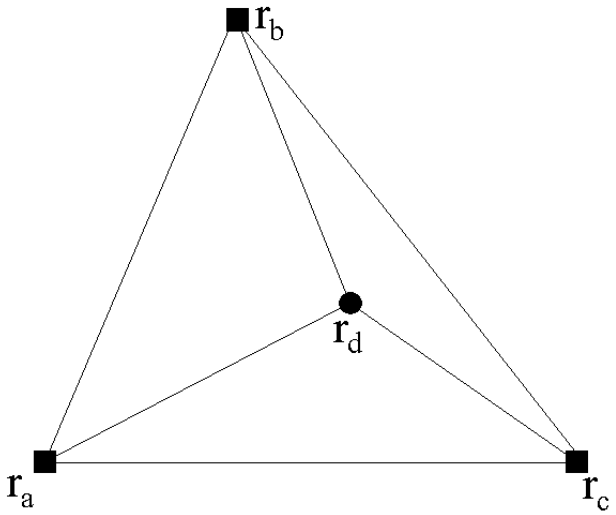


Figure 11.6. How a triangle is subdivide in three. The position of the new node, $\mathbf{r}_d$ is constrained to lie inside the *spherical* triangle ABC formed by its parent nodes.

$$\alpha = 1 - \beta - \gamma, \text{ and } \alpha \geq 0,\ \beta \geq 0,\ \gamma \geq 0, \tag{11.9}$$

which gives two degrees of freedom for each new node. The position of the child node is then projected onto the sphere:

$$\mathbf{r}_d \rightarrow \frac{\mathbf{r}_d}{\|\mathbf{r}_d\|}. \tag{11.10}$$

The mesh is recursively subdivided in this manner until ϕ is defined to some desired accuracy.

11.2.2 Cauchy Kernel Re-parameterisation

The kernel-based method described in §8.1.2 can also be extended to re-parameterising the sphere. The basic idea is to draw a great circle between any point $\mathbf{u}$ and a 'fixed' point $\mathbf{m}$ on the sphere see figure 11.7. A smooth, diffeomorphic re-parameterisation of the sphere can now be achieved by applying the same re-parameterisation[1] to each great circle using the approach described in section 8.1. For now, we will assume that $\mathbf{m}$ corresponds to the point $\theta = 0$. It will be

[1] Personal communication, Carole Twining, University of Manchester.

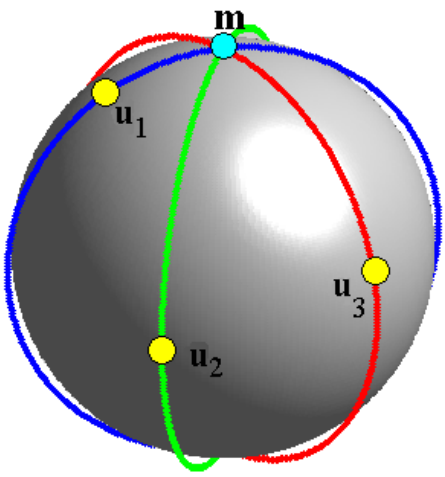


Figure 11.7. How a sphere is re-parameterised. For any point $\mathbf{u}_j$ on the sphere, a great circle is drawn through it and a fixed point $\mathbf{m}$ (the centre of the kernel). Each great circle can now be re-parameterised using the same function to give a symmetric sphere re-parameterisation.

shown later that this can be generalised to allow $\mathbf{m}$ to lie anywhere on the sphere surface.

To re-parameterise the latitude (θ) coordinate ($\theta \rightarrow g(\theta)$), $g(\theta)$ must be valid for ($0 \leq \theta \leq \pi$). This can be achieved by using the formulation introduced in (8.9):

$$g(\theta) = c^{-1}\left[\theta + A\arccos\left(\frac{(1+\omega^2)\cos\theta - 2\omega}{1+\omega^2-2\omega\cos\theta}\right)\right], \quad 0 \leq \theta \leq \pi, \quad 0 \leq g \leq \pi, \tag{11.11}$$

where c is a normalisation term: $c = 1 + A$.

We must now adapt this formulation to apply to any kernel position, $\mathbf{m}$. For any point $\mathbf{u}$, we wish to apply the transformation so that $\mathbf{u} \rightarrow \mathbf{u}'$, where $\mathbf{m}, \mathbf{u}$ and $\mathbf{u}'$ are all unit vectors since they lie on the surface of the unit sphere. First, we define a unit vector $\mathbf{n}$ that is in the same plane as $\mathbf{m}$ and $\mathbf{u}$ and orthogonal to $\mathbf{m}$:

$$\mathbf{m} \cdot \mathbf{n} = 0, \tag{11.12}$$
$$\mathbf{u} \cdot \mathbf{m} = \cos(\theta), \tag{11.13}$$
$$\mathbf{u} \cdot \mathbf{n} = \sin(\theta). \tag{11.14}$$

The position of $\mathbf{u}'$ can now be written as a linear combination of $\mathbf{m}$ and $\mathbf{u}$, see

figure 11.8:

$$\mathbf{u}' = \beta\mathbf{m} + \alpha\mathbf{u}. \tag{11.15}$$

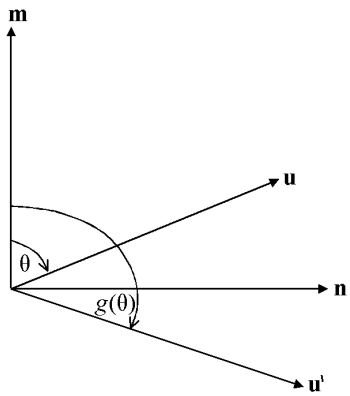


Figure 11.8. The relationship between $\mathbf{m}$, $\mathbf{q}$, $\mathbf{n}$ and $\mathbf{u}'$.

We must now find the values of β and α:

$$\begin{aligned} \mathbf{u}' \cdot \mathbf{m} = &\quad \beta + \alpha\mathbf{u} \cdot \mathbf{m} &&= \cos(g(\theta)), && (11.16)\\ \mathbf{u}' \cdot \mathbf{n} = &\quad \alpha\mathbf{u} \cdot n &&= \sin(g(\theta)), && (11.17) \end{aligned}$$

where $g(\theta)$ is the angle between $\mathbf{m}$ and $\mathbf{u}'$ as defined in (11.11). Rearranging and substituting in (11.13) and (11.14):

$$\begin{aligned} \alpha &= \frac{\mathbf{u}' \cdot \mathbf{n}}{\mathbf{u} \cdot \mathbf{n}} = \frac{\sin(g(\theta))}{\sin(\theta)}, && (11.18)\\ \beta &= \mathbf{u}' \cdot \mathbf{m} - \alpha\mathbf{u} \cdot \mathbf{m} = \cos(g(\theta)) - \frac{\sin(g(\theta))}{\sin(\theta)}\cos(\theta). && (11.19) \end{aligned}$$

We can now write:

$$\mathbf{u} \rightarrow \mathbf{u}', \tag{11.20}$$

$$\mathbf{u} \rightarrow \beta\mathbf{m} + \alpha\mathbf{u}, \tag{11.21}$$

$$\mathbf{u} \rightarrow \frac{\sin(g(\theta))}{\sin(\theta)}\mathbf{m} + \mathbf{u}\left(\cos(g(\theta)) - \frac{\sin(g(\theta))}{\sin(\theta)}\cos(\theta)\right), \tag{11.22}$$

$$\mathbf{u} \rightarrow \frac{\sin(g(\theta))}{\sin(\arccos(\mathbf{m}\cdot\mathbf{u}))}\mathbf{m} + \mathbf{u}\left(\cos(g(\theta)) - \frac{\sin(g(\theta))}{\sin(\arccos(\mathbf{m}\cdot\mathbf{u}))}\mathbf{m}\cdot\mathbf{u}\right). \tag{11.23}$$

This formulation allows us to find the re-parameterised position of any point, $\mathbf{u}$, given an arbitrary kernel position $\mathbf{m}$.

Figure 11.9 shows how the smooth representation of ϕ can re-parameterise a surface. The figure shows that points near the centre of the kernel are 'spread' over the sphere (hence spreading corresponding points on the surface). The bigger the kernel, the more the points are spread.

11.2.3 Further Extensions

The method described above has proved sufficient for all the practical experiments presented in the remainder of the thesis. This section considers more general sets of transformations and possible extensions to non-spherical topologies.

More General Transformations

Although the symmetric transformation described above provides considerable representational power it is interesting to consider a more general set of functions. For example, we can create an *asymmetric* transformation by making the amplitude of the Cauchy kernel a smooth, periodic function of the longitude parameter:

$$A \rightarrow A(\psi), \quad A(2\pi + \psi) = A(\psi). \tag{11.24}$$

We can also accomplish a *shearing* about an axis by using a transformation of the form:

$$\psi \rightarrow \psi + f(\theta). \tag{11.25}$$

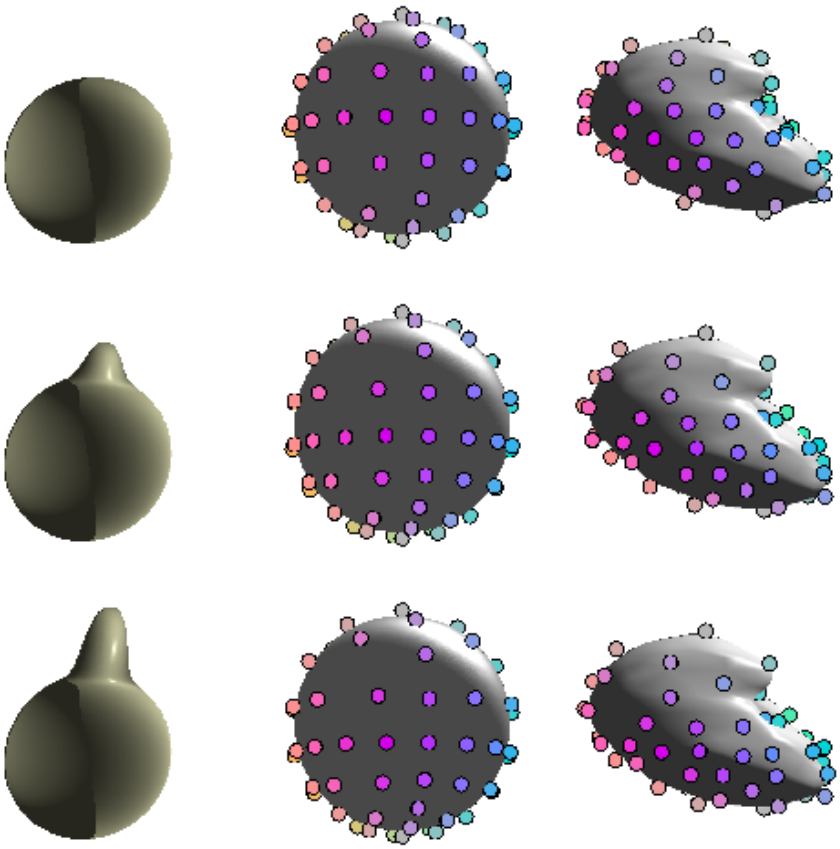


Figure 11.9. A demonstration of a symmetric theta transformation of a parameterised closed surface. *Left*: the distribution on the sphere; *middle*: the parameterisation on the sphere; *right*: the parameterised points on the surface (the colours denote the correspondence). The magnitude, A, of the Cauchy is 0 in the top row, 0.5 in the middle and 1 at the bottom. Points near the 'bump' get spread around on the sphere (hence spreading points on the surface).

These transformations have been described in more detail elsewhere [30] and implemented, but the symmetric theta transformations proved sufficiently general for all objects modelled in this thesis.

Other Topologies

The treatment so far has considered only objects of spherical topology. Although this was appropriate for all classes of objects studied, it is interesting to consider how we could treat other topologies. Section 8.1 of chapter 8 described how a 1D circle ($\mathbb{S}^1$) or 1D line ($\mathbb{R}^1$) could be re-parameterised. We can therefore re-parameterise any surface whose product space is a combination of ($\mathbb{S}^1$) and ($\mathbb{R}^1$). For example, if we represent the cylinder as the product space of $\mathbb{R}^1 \times \mathbb{S}^1$, we

can re-parameterise either $\mathbb{S}^1$ or $\mathbb{R}^1$. Similar treatments can applied to the torus ($\mathbb{S}^1 \times \mathbb{S}^1$) or the plane ($\mathbb{R}^1 \times \mathbb{R}^1$).

11.3 Optimisation in 3D

We now consider how the Cauchy kernel representation of re-parameterisation can be used to construct optimal 3D statistical shape models. The optimisation method is a direct extension of the 2D version described in chapter 9. A schematic giving an overview of the method is shown in figure 11.1.

We obtain an initial approximation to the optimal values of $\{s_i, \mathbf{R}_i, \mathbf{d}_i\}$ by aligning the training examples by hand although an automated method such as the iterative closest point (ICP) [6] algorithm could also be used.

As in 2D, the origin of each parameterisation is a free parameter. Moving the position of the origin is equivalent to rotating the parameterisation (i.e. the sphere). We can obtain an initial estimate of the origin as follows:

- assign a 'reference' example (we choose the most 'typical' example by hand [a]).
- **for** each example i (except the reference example):
  - **do**
    select a rotation for the parameterisation (i.e. the sphere) of the example.
    - uniformly sample n points on the sphere (i.e. the parameterisation), see appendix B for details.
    - project the sampled points onto the surface using Ω_i^{-1}.
    - measure the point-to-point squared distance between the current example and the reference example.
  - **until** the squared distance to the first reference shape is minimised.

[a] Although we could choose any example, an abnormal reference shape will slow the convergence of the optimisation.

The following algorithm is used to optimise the value of F with respect to a set of re-parameterisation functions $\{\phi_i\}$ and pose parameters $\{s_i, R_i\}$:

- **for** $i = 1 \ldots n_s$
  - $S'_i = s_i \mathbf{R}_i S_i(\phi_i)$
  - build the model from $\{S'_i\}$
  - evaluate the model using F

Given this and starting from the estimate given above, F is optimised with respect to $\{\phi_i, s_i, \mathbf{R}_i, \mathbf{d}_i\}$:

- **repeat**
  - Randomise the ordering of the examples
  - **for** each example i (except the reference example):
    - randomly place a kernel on the sphere, see appendix C for details.
    - randomly select a width for the kernel from the positive half of a Gaussian distribution with standard deviation $\sigma_G = 1/2$.
    - optimise the value of A_i, the amplitude of the kernel, using the Simplex algorithm
    - transform the re-parameterised points to the centre of gravity
    - optimise the values of s_i and $\mathbf{R}_i$
    - optimise the position of the origin (i.e. the rotation of the parameterisation)
- **until** convergence.

11.4 Summary

This chapter has shown that the methods of representing re-parameterisation developed in 2D can be extended to 3D. Given the experience gained in 2D, most attention has been paid to the method based on the use of the Cauchy kernels. There is only on free parameter – the width of the distribution from which the widths of Cauchy kernels are drawn. Results of applying the method to 3D training data are presented in the following two chapters.

Chapter 12

3D Results

"I didn't think; I experimented. ."
– *Wilhelm Roentgen.*

The previous chapter showed how the method of automatic model-building developed in 2D can be extended to deal with training sets of 3D shapes. This chapter presents the results of applying the method to four different 3D datasets. A quantitative evaluation shows that the proposed method constructs significantly better models than those built using alternative approaches.

12.1 The 3D Experiments

For each dataset, an MDL model was constructed using the algorithm described previously in §11.3 and compared to a model built using the SPHARM (spherical harmonic) method of Kelemen, Gerig *et al.* [55], which is arguably the best published approach to defining correspondence between sets of closed surfaces. It was shown in §4.3, that the method essentially spaces points over the surface according to the parameterisation, hence we have replicated a good approximation to the method by equally spacing points over the training surfaces. For the final dataset, (the hippocampus), Gerig *et al.* have kindly given us the model produced by their method, allowing a direct comparison to the MDL model. To distinguish between the two implementations, we refer to them as the Uniform and SPHARM models.

Quantitative comparisons of the models are reported by evaluating the models using the criteria described in §5.3 (generalisation, specificity and compact-

ness) and calculating the objective function value of each model. Qualitative results are presented by showing the first three modes of variation, independently varying the shape parameters by ± 2[standard deviations found over the training shapes].

Results are presented for four training sets of pre-segmented 3D objects: 23 induced stroke shapes, 8 brain ventricles (the anterior horn), 16 rat kidneys and 82 hippocampi. The details of the datasets with signposts to the compendium of results are set out below.

Stroke Model

Permanent focal cerebral ischaemia was induced in rats, and multi-slice T2-weighted magnetic resonance imaging (MRI) was performed *in vivo*, as described previously [107]. For the 23 examples used in this study, the shape of the infarct represents the territory of the middle cerebral artery. Qualitative and quantitative comparisons of the resulting models are shown in figures 12.1 (page 129) and 12.6 (page 134) respectively.

Brain Ventricle

T1-weighted magnetic resonance volume images of the brain were acquired for eight normal volunteers on a 0.5T GE Vectra scanner. For each image, the anterior horn of the right ventricle was segmented by a neuroradiologist. Qualitative and quantitative comparisons of the resulting models are shown in figures 12.2 (page 130) and 12.7 (page 135) respectively.

Rat Kidney

Wistar, Sprague-Dawley, and Fisher rats were imaged using a 4.7T Varian Inova MRI system. Multi-slice T2-weighted MRI was performed in the transverse plane with repetition time (TR) 2 sec; echo time (TE) 20 msec and slice thickness 1 mm, with 41 contiguous slices. Images were acquired with a 64 x 64 mm field of view and a 256 x 256 x 41 image matrix. Sixteen segmentations of the right kidney were used in this study. Since the manual segmentations were noisy, each example was smoothed, following triangulation, using Laplacian smoothing [85] (λ=0.1, 20 iterations) and decimated to reduce the number of triangles by 80%. Qualitative and quantitative comparisons of the resulting models are shown in figures 12.3 (page 131) and 12.8 (page 136) respectively.

Hippocampus

[1]Magnetic resonance images (MRI) of the brain were acquired on a GE Sigma Advantage 1.5T scanner (D IR Prepped Axial Spoiled Gradient; Fast SPGR Axial plane, 3D Acquisition TE min full; TR=15ms; flip angle=20^0; FOC = 24cm; Bandwidth =16KHZ; resolution = $256 \times 256 \times 124$; slice thickness = 1.5mm) as described previously [14, 40]. The hippocampus (hippocampus proper, subiculum, fimbria and subsplenal gyrus) was segmented as described in [14] and smoothed and parameterised using the method described in [8]. The data comprised 56 hippocampi of schizophrenic patients and 26 of age-matched control subjects. The two groups were combined to form a training set of 82 examples. A quantitative comparison of the resulting models are shown in figure 12.9 (page 137). The first three modes of the SPHARM and MDL models are shown in figures 12.4 (page 132) and 12.5 (page 133) respectively. These models are used in a shape discrimination experiment in the next chapter.

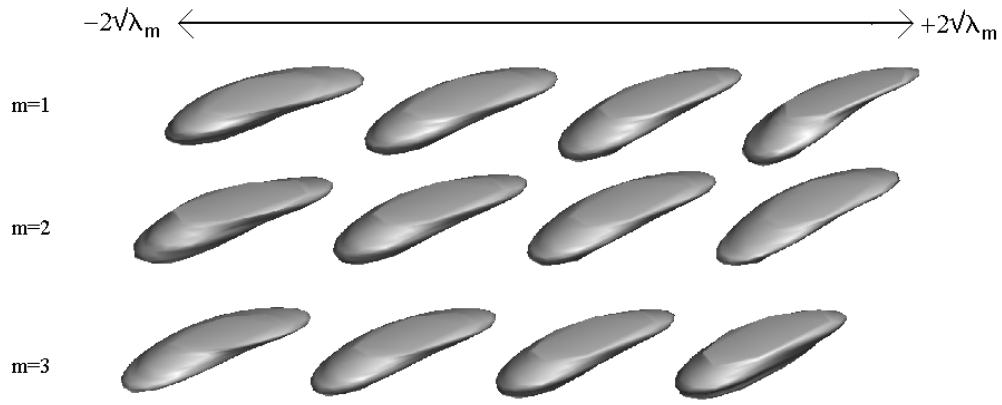
12.2 Discussion of Results

The qualitative results show that there are noticeable differences between the MDL and Uniform/SPHARM models. The quantitative results show that the MDL models have significantly better generalisation and specificity properties than the Uniform/SPHARM models for all datasets. The MDL model is also more compact for all datasets but the difference is only statistically significant for the Stroke (12.6) and Hippocampus data (12.9). The lack of significance for the Kidney and Ventricle data is probably due to the small sample size.

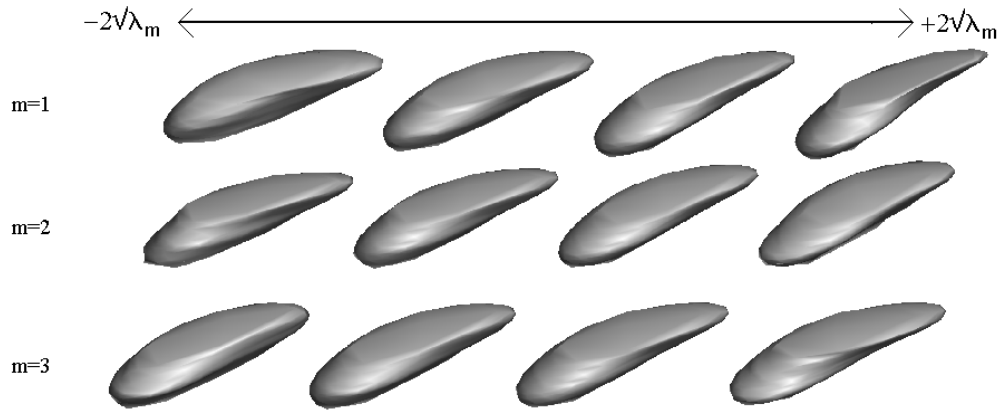
The 3D optimisation algorithm takes much longer to converge than the 2D version; for example, using a MATLAB implementation on a PC (Intel Pentium II – 450KhZ, 256Mb RAM) the ventricles (8 instances) converged after six hours but the hippocampi (82 examples) took over five days. Although model building is a one-off, off-line process these convergence times are inconveniently long. If, however, the algorithm was implemented in a lower-level language (e.g. C++, FORTRAN) on a modern PC, the convergence time would decrease by some orders of magnitude, allowing a large dataset (≈ 100 examples) to be optimised overnight.

For the hippocampus data, the MDL model was initialised using the SPHARM correspondences. This allows a direct comparison of the correspondences, as shown in figure 12.10. The most substantial differences are on the tail and the head, this is not surprising as these are the regions of highest curvature and

[1]The hippocampus data was kindly provided by Guido Gerig and co-workers, University of North Carolina, Chapel Hill



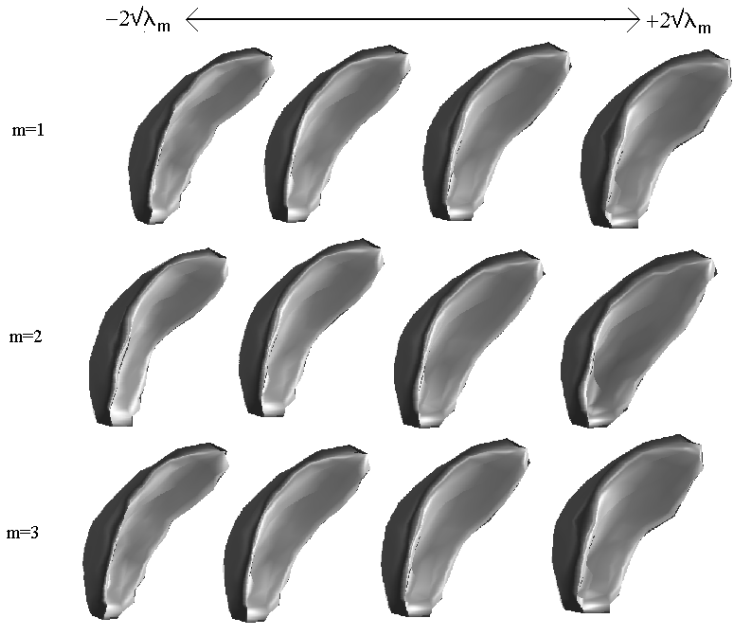
a: Uniform



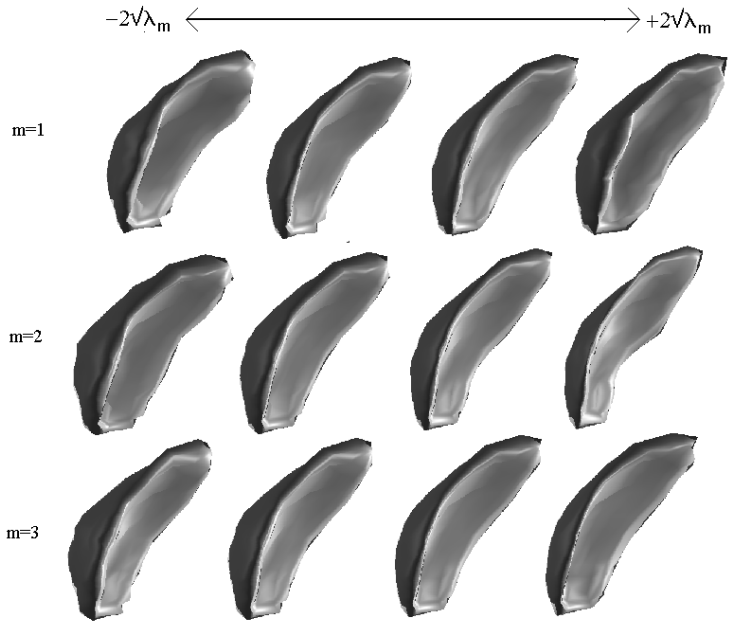
b: MDL

Figure 12.1. A qualitative comparison of the Stroke models.

demonstrate the most variation. Another interesting effect is the 'skewing' that occurs around the middle of the hippocampus body.

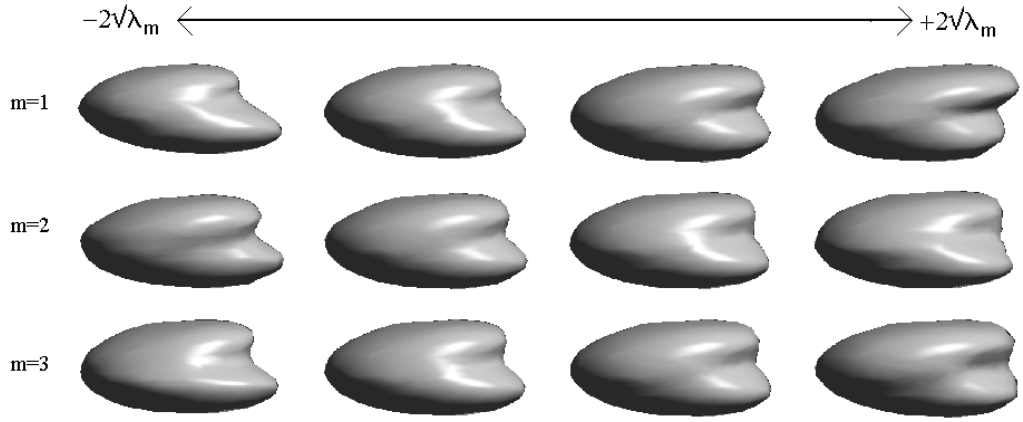


a: Uniform

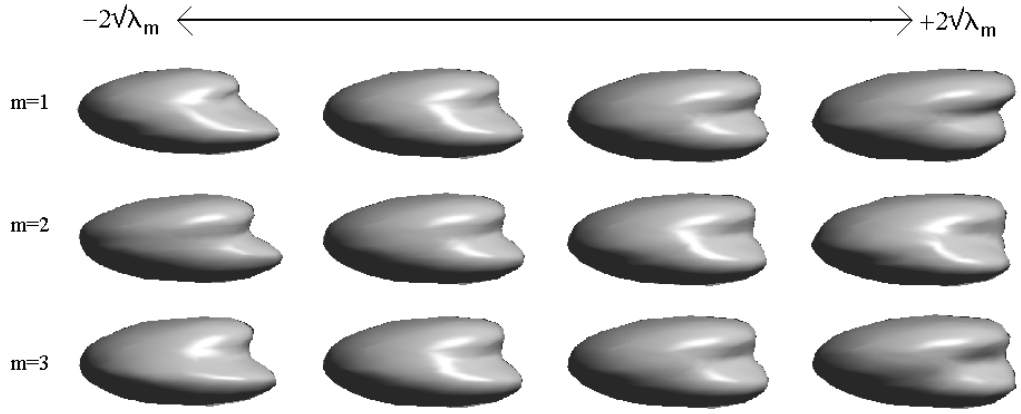


b: MDL

Figure 12.2. A qualitative comparison of the Brain Ventricle models.



a: Uniform



b: MDL

Figure 12.3. A qualitative comparison of the Rat Kidney models.

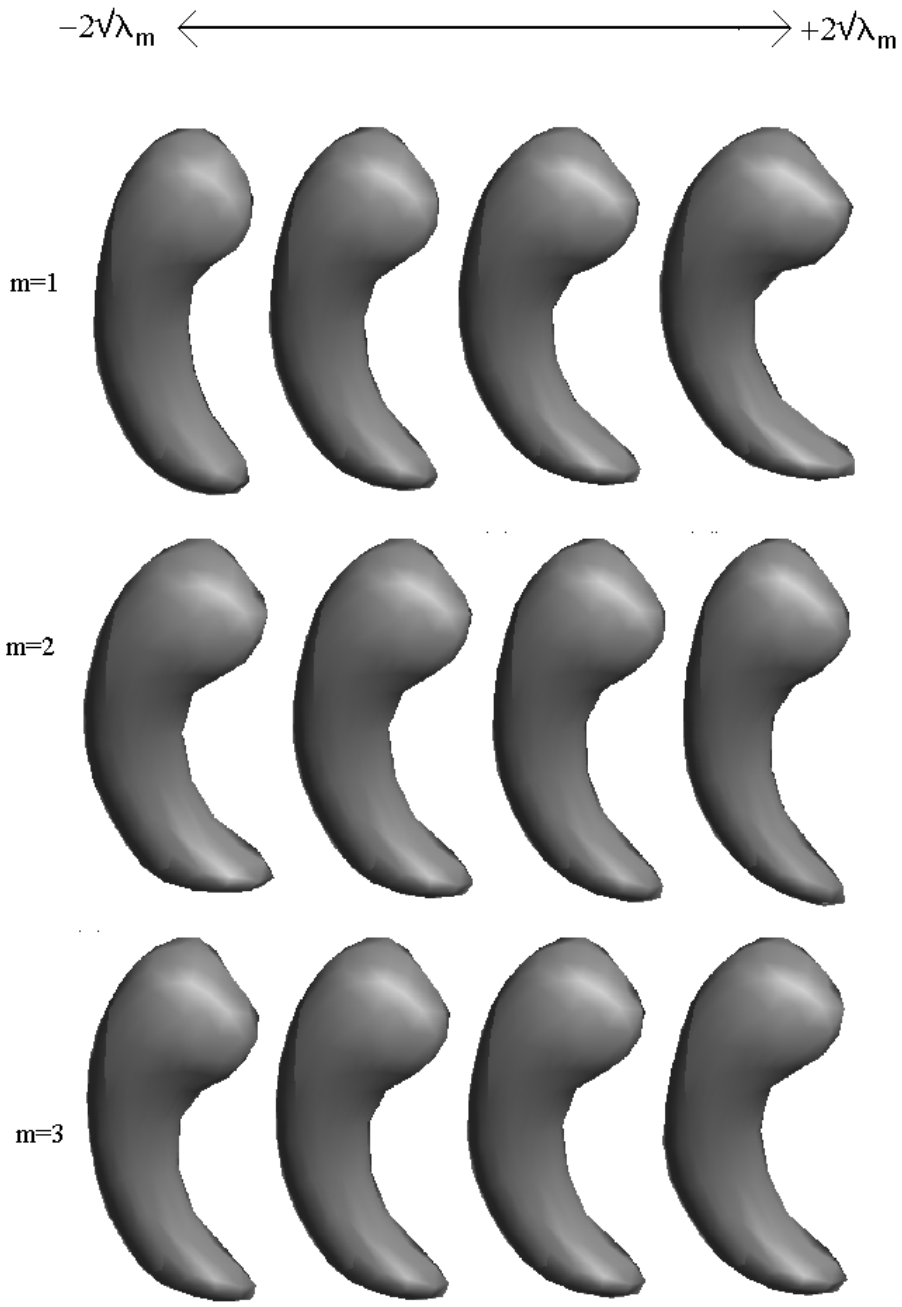


Figure 12.4. The first three modes of the SPHARM hippocampus model.

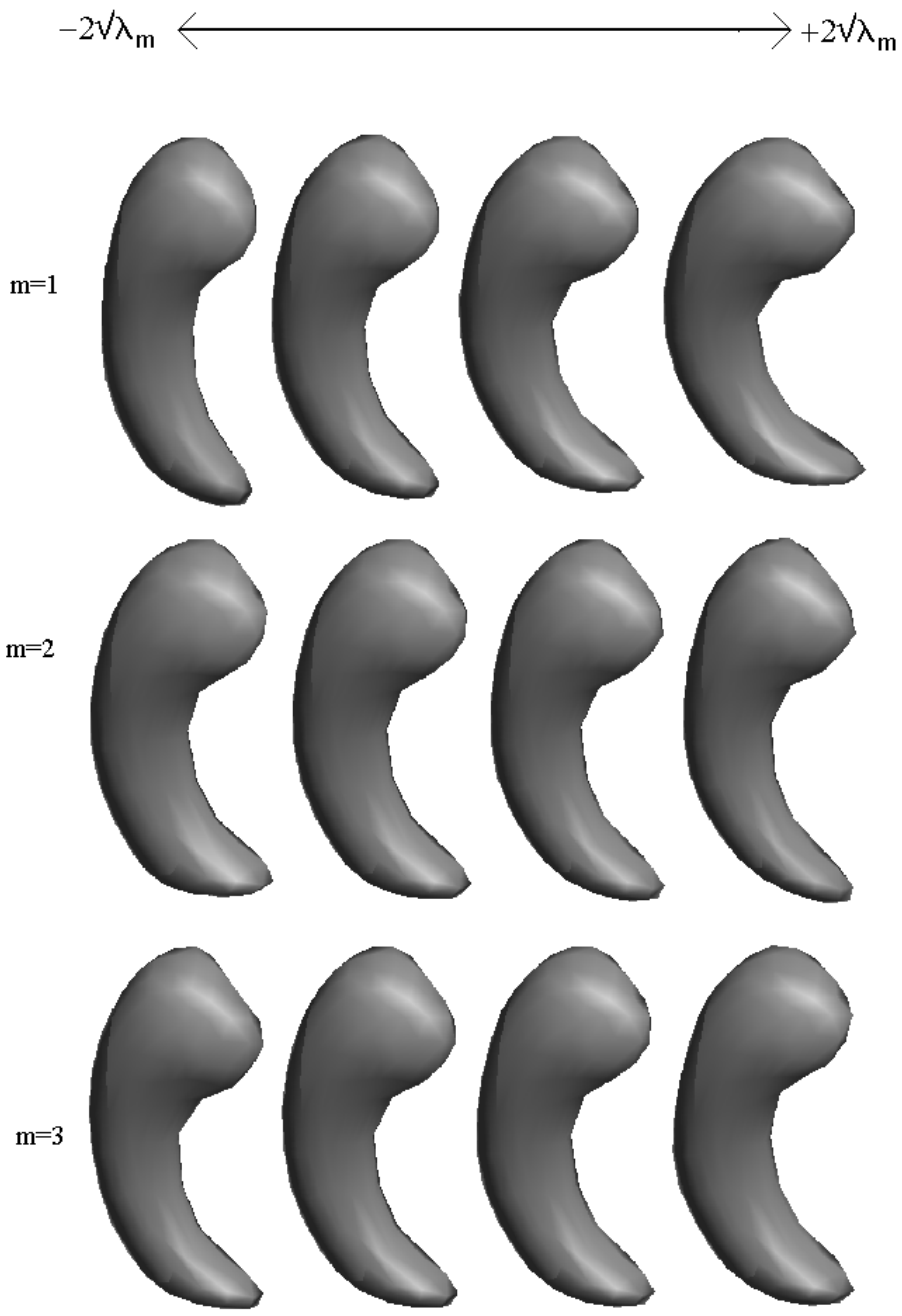
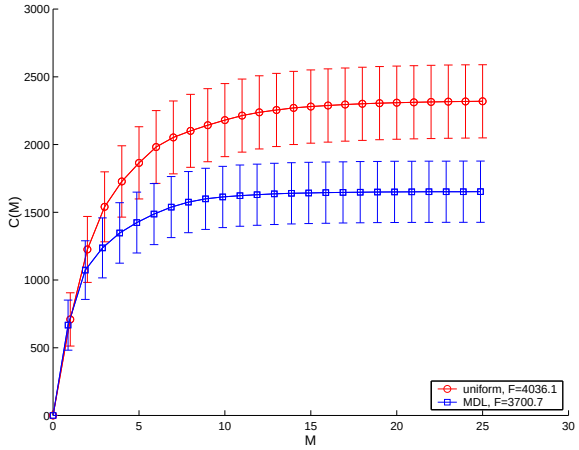
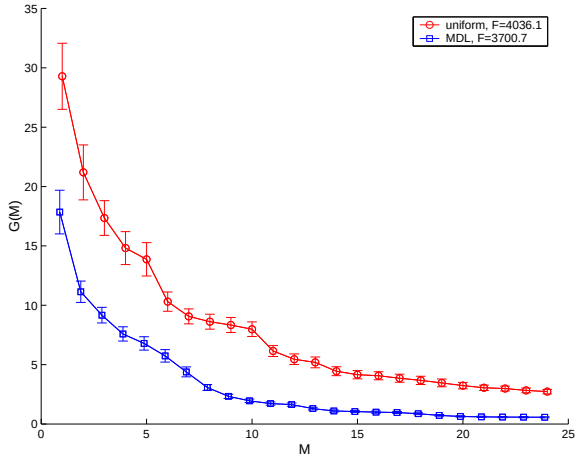


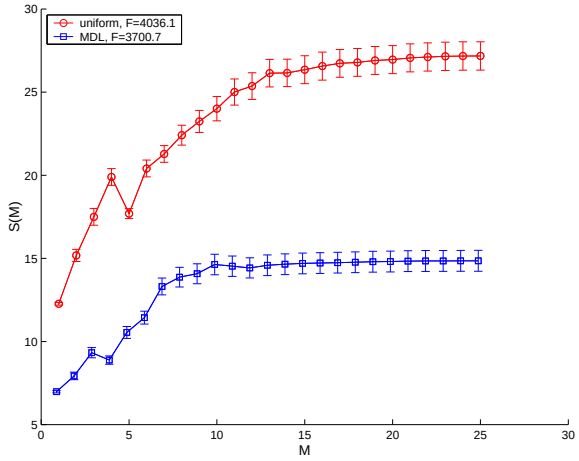
Figure 12.5. The first three modes of the MDL hippocampus model.



a: Compactness.

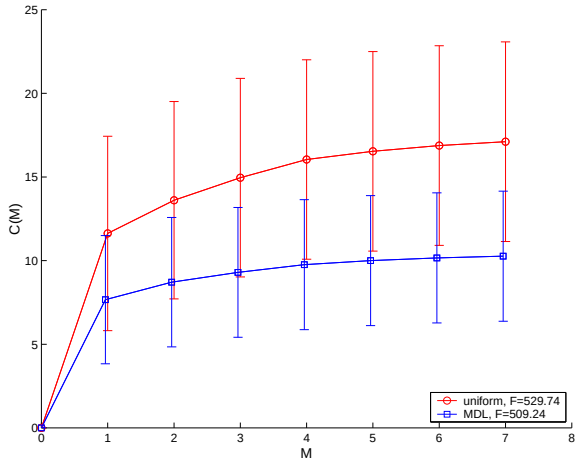


b: Generalisation ability.

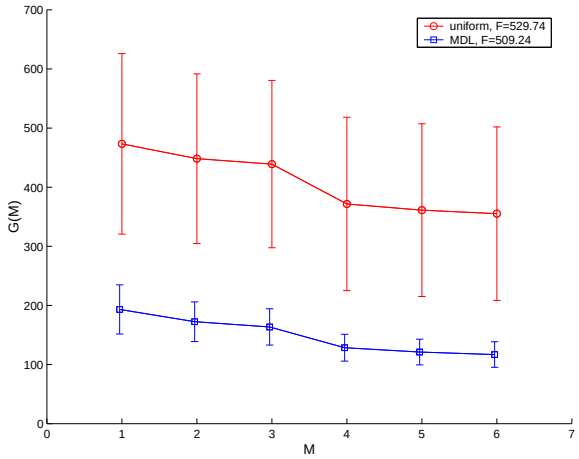


c: Specificity.

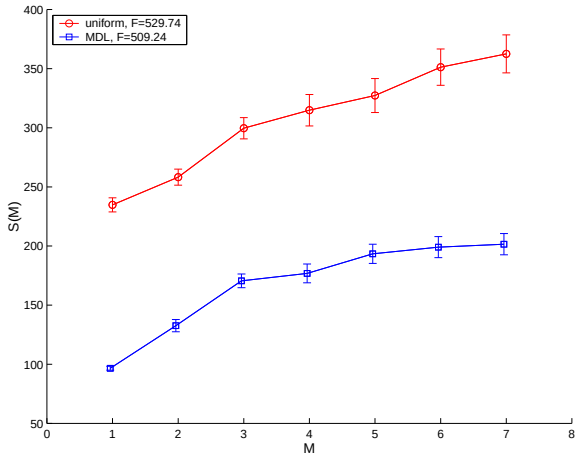
Figure 12.6. A quantitative comparison of the stroke models.



a: Compactness.

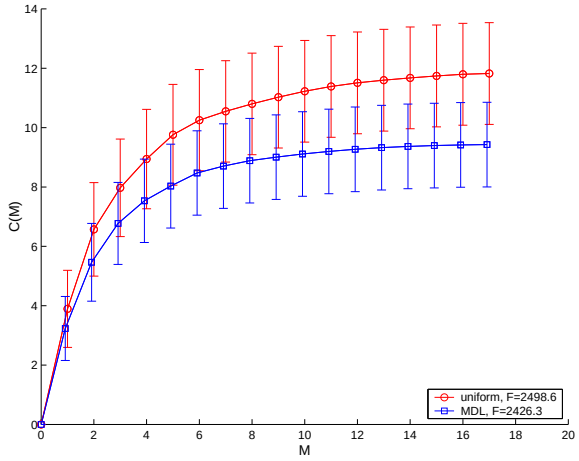


b: Generalisation ability.

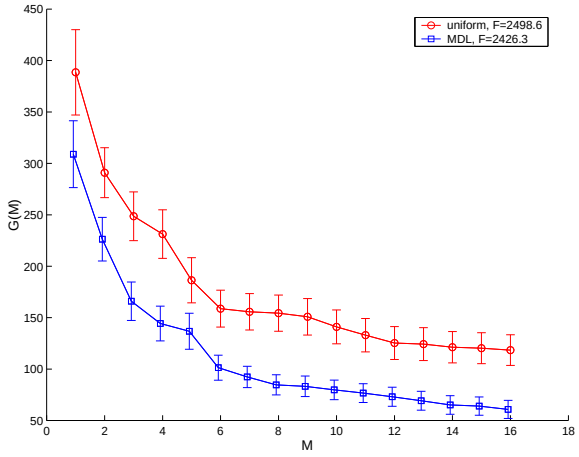


c: Specificity.

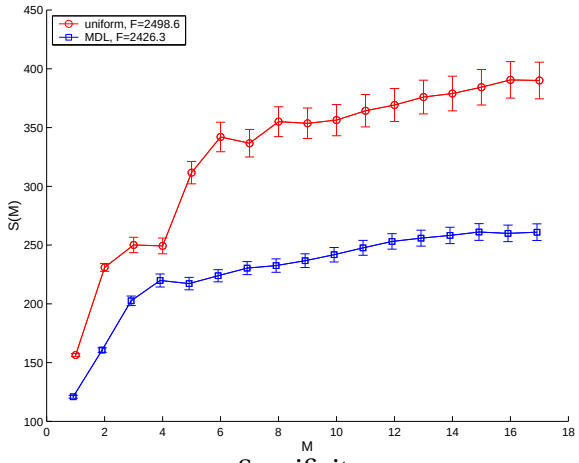
Figure 12.7. A quantitative comparison of the brain ventricle models.



a: Compactness.

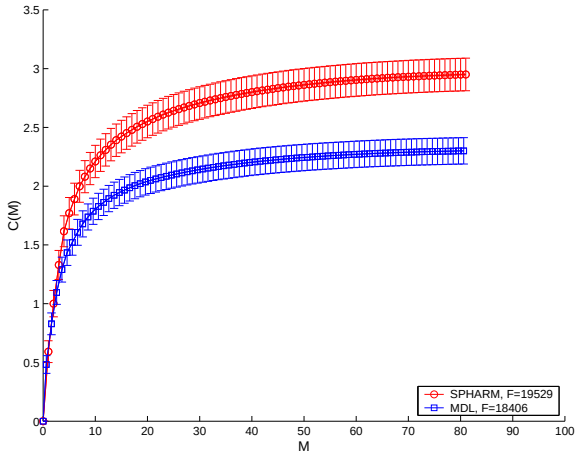


b: Generalisation ability.

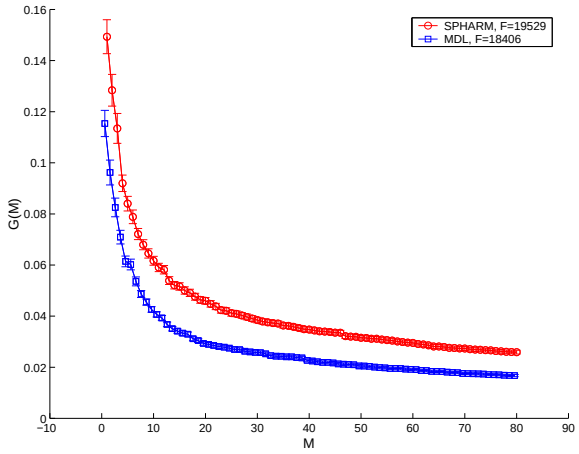


c: Specificity.

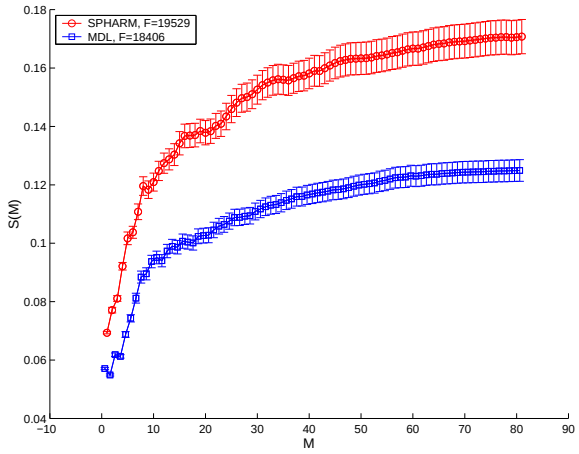
Figure 12.8. A quantitative comparison of the rat kidney models.



a: Compactness.

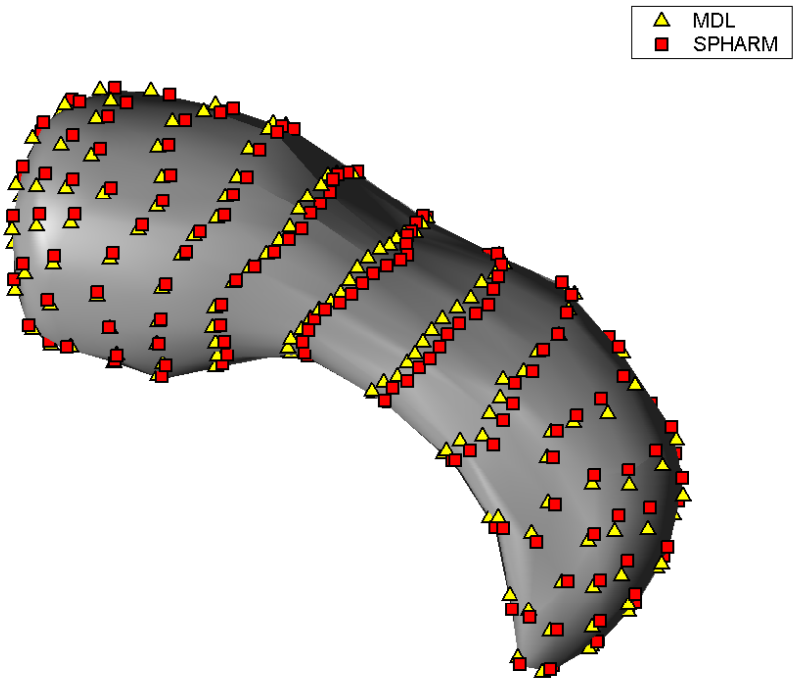


b: Generalisation ability.

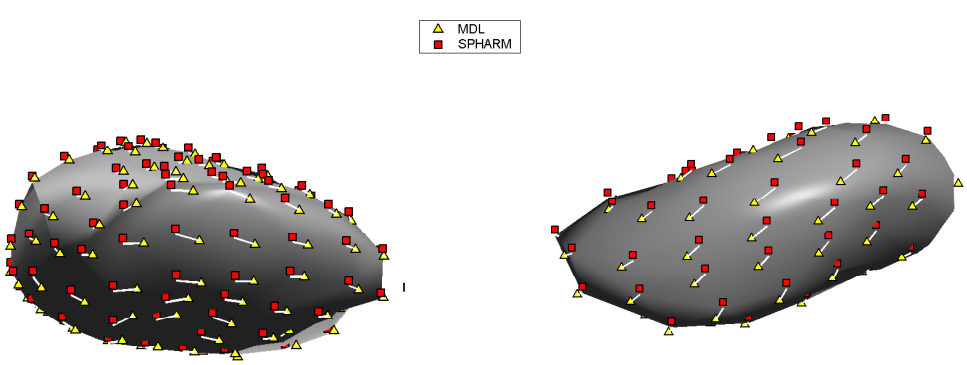


c: Specificity.

Figure 12.9. A quantitative comparison of the hippocampus models.



a: Side view.



b: The head.

c: The tail.

Figure 12.10. A sparse subset of the SPHARM and MDL correspondences.

Chapter 13

Application to Shape Discrimination

"To Thales the primary question was not what do we know, but how do we know it."
– *Aristotle.*

Previous chapters have shown how optimal statistical shape models may be constructed from sets of 2D or 3D training shapes. Quantitative criteria were used to demonstrate that the resulting models were, in a technical sense, better than those obtained using alternative methods. This leaves the important question 'does the technical superiority translate into a real benefit in practical applications?' This question is addressed in this chapter, where an automatically constructed MDL model is used as a basis for exploring differences in shape between the hippocampi of schizophrenic and control subjects. The results are compared to those obtained using a model of the same data generated by an internationally leading group in the field. We show that the MDL model provides significantly better discrimination

13.1 Motivation and Overview

There is much evidence to connect altered hippocampal form (e.g. [16, 62, 69, 109]) with schizophrenia. The connection is often studied using volumetric measurements but recent work (e.g. [45, 105, 41]) suggests that shape information provides better discrimination.

Shape discrimination can be performed by constructing an MDL model of the hippocampi for schizophrenic and healthy control subjects combined and performing linear discriminant analysis (LDA) on the model parameters of the two

groups. The discriminating characteristics can be visualised, statistically analysed, and compared to those obtained using a SPHARM model [55].

The hippocampus dataset consisted of 56 schizophrenic patients and 26 age-matched control subjects, segmented from magnetic resonance (MR) images. The schizophrenic group contains two subgroups: 34 patients with early onset of schizophrenia (< 5 years) and 22 chronic patients (> 10 years) but in our experiments these were combined to form a single schizophrenic group. The data was used to construct an MDL shape model of the right hippocampus using the algorithm described in §11.3. For comparative purposes, a SPHARM model was also constructed as described in [55]. Only the modes required to capture the first 95% of the shape variation were retained giving in each case, a set of 33-dimensional shape vectors: $\{\mathbf{b}_i : i = 1 \dots 82\}$. Following model construction, the parameter vectors were split into two groups: $C = \{\mathbf{c}_i : i = 1 \dots 26\}$ (the control subjects) and $S = \{\mathbf{s}_i : i = 1 \dots 56\}$ (the schizophrenic patients).

13.2 Discriminant Analysis

Linear discriminant analysis (LDA – using Fisher's criterion) was used to discriminate between C and S. Only the two-class case of LDA is discussed here, for a more general discussion, see [108] or [46].

13.2.1 Linear Discriminant Analysis

Fisher's criterion is maximised when the between-class separation of two groups is maximised and the within class separation is minimised. The vector, $\mathbf{w}$, that maximises Fisher's criterion can be found as follows:

Construct a scatter matrix for each group:

$$\mathbf{T}_c = \sum_{i=1}^{26} (\mathbf{c}_i - \bar{\mathbf{c}})(\mathbf{c}_i - \bar{\mathbf{c}})^T, \tag{13.1}$$

$$\mathbf{T}_s = \sum_{i=1}^{56} (\mathbf{s}_i - \bar{\mathbf{s}})(\mathbf{s}_i - \bar{\mathbf{s}})^T, \tag{13.2}$$

where $\bar{\mathbf{s}} = \frac{1}{56} \sum_{i=1}^{56} \mathbf{s}_i$ and $\bar{\mathbf{c}} = \frac{1}{26} \sum_{i=1}^{26} \mathbf{c}_i$ are the the mean vectors.

Construct a pooled within-class scatter matrix from (13.1) and (13.2):

$$\mathbf{T}_W = \frac{1}{82-2}\left(26\mathbf{T}_c + 56\mathbf{T}_s\right). \tag{13.3}$$

For 2 classes, Fisher's criterion can be written as:

$$J_F = \frac{|\mathbf{w}^T(\bar{\mathbf{s}} - \bar{\mathbf{c}})|^2}{\mathbf{w}^T\mathbf{T}_W\mathbf{w}}. \tag{13.4}$$

Differentiating J_F with respect to $\mathbf{w}$ and equating the derivative to zero gives:

$$\frac{\mathbf{w}^T(\bar{\mathbf{s}} - \bar{\mathbf{c}})}{\mathbf{w}^T\mathbf{T}_W\mathbf{w}}\left[2(\bar{\mathbf{s}} - \bar{\mathbf{c}}) + \left(\frac{\mathbf{w}^T(\bar{\mathbf{s}} - \bar{\mathbf{c}})}{\mathbf{w}^T\mathbf{T}_W\mathbf{w}}\right)\mathbf{T}_W\mathbf{w}\right] = 0. \tag{13.5}$$

Since $\frac{\mathbf{w}^T(\bar{\mathbf{s}}-\bar{\mathbf{c}})}{\mathbf{w}^T\mathbf{T}_W\mathbf{w}}$ is a scalar, the vector, $\mathbf{w}$, that maximises J_F is given by:

$$\mathbf{w} = \alpha\,\mathbf{T}_W^{-1}(\bar{\mathbf{s}} - \bar{\mathbf{c}}), \tag{13.6}$$

where α is a constant that denotes the length of $\mathbf{w}$. For the results reported below, α was chosen so that $\mathbf{w}$ had unit norm: $||\mathbf{w}|| = 1$.

13.2.2 Visualising the Discriminating Characteristics

LDA was applied to find a discriminating vector of the two groups of shape parameters of the MDL model. The discriminating characteristics can be visualised by creating shape parameters that produce a mode of variation along $\mathbf{w}$:

$$\mathbf{b} = \bar{\mathbf{b}} + \mathbf{w}d, \tag{13.7}$$

where d is the shape parameter of the discriminant mode.

Since the elements of $\mathbf{b}$ have zero mean, new shape instances can be created using:

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}\mathbf{w}d. \tag{13.8}$$

The discriminant mode of variation is shown in figure 13.1 for d varying over the range $[\min(\mathbf{w}^T\mathbf{b}), \max(\mathbf{w}^T\mathbf{b})]$which is roughly equivalent to ± 3 [standard deviations found over the training set]. The main effect, as the discriminant parameter

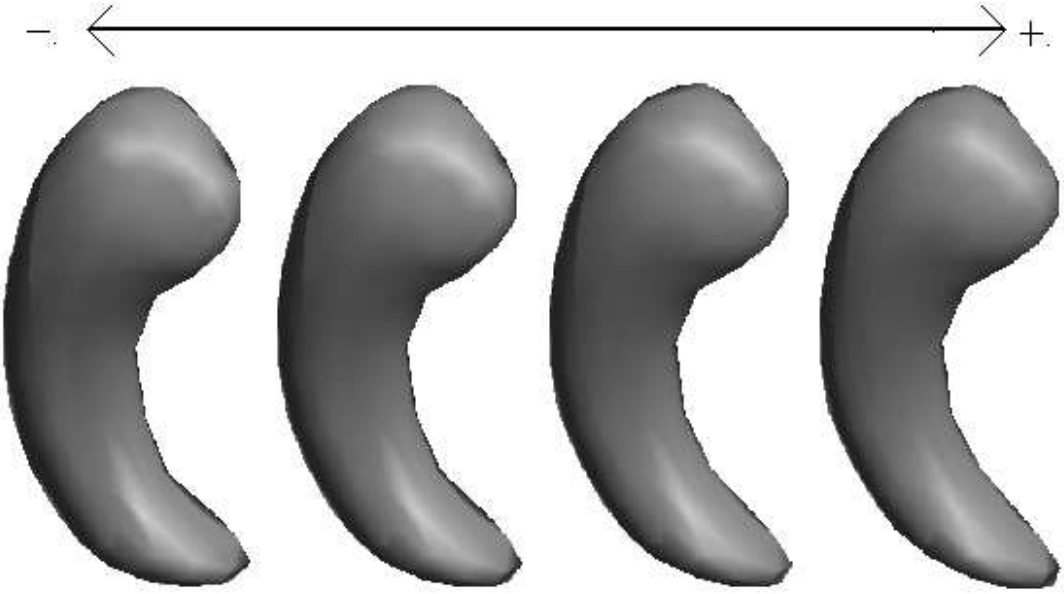


Figure 13.1. The discriminant mode of the hippocampal data varied from − (control) to + (schizophrenic). The most noticeable effect is a lengthening and narrowing of the 'tail'.

is moved towards the schizophrenic group, is a thinning and straightening of the 'tail'.

13.2.3 Statistical Analysis

An unbiased estimate of the separability of the two groups can be obtained using the following analysis:

- **for** each shape example $j = 1 \ldots n_s$
  - create the training set $\{\mathbf{x}_i\}_j$ with $\mathbf{x}_j$ excluded
  - build the model $(\bar{\mathbf{x}}_j, \mathbf{P}_j)$ from $\{\mathbf{x}_i\}_j$
  - find the shape parameters for each member $\{\mathbf{x}_i\}_j$:
    $$\mathbf{b}_i = \mathbf{P}_j^T(\mathbf{x}_i - \bar{\mathbf{x}}_j)$$
  - separate $\{\mathbf{b}_j\}$ into $\{\mathbf{s}_j\}$ and $\{\mathbf{c}_j\}$.
  - find the vector, $\mathbf{w}_i$, that maximises Fisher's criterion for $\{\mathbf{s}_i\}$ and $\{\mathbf{c}_i\}$.
  - estimate the parameters of the excluded example:
    $$\mathbf{b}_j = \mathbf{P}_j^T(\mathbf{x}_i - \bar{\mathbf{x}}_j);$$
  - project the parameters of $\mathbf{x}_j$ into discriminant space:
    $$y_j = \mathbf{w}_j^T \mathbf{b}'_j$$

The analysis was first performed using the MDL model. The projected values of the two groups are shown in figure 13.2. The difference between the means is statistically significant – a t-test shows that the probability that both means were sampled from the same underlying population is $p = 0.0042,\ (t = 2.944)$.

The analysis was repeated using a model constructed by the SPHARM method of Kelemen *et al.* [55]. The projected values for the two groups are shown in figure 13.3. A comparison of figures 13.2 and 13.3 suggests that the differences between the means of the two groups are similar but the within-class variances are smaller for the MDL model. A t-test shows that the difference between the two groups is still significant ($p = 0.0163, t = 2.454$), but with a lower level of significance than the MDL model. In order to determine whether this implies that the MDL results are significantly better than the SPHARM results, the sampling distributions of the t values must be estimated. To achieve this, the Bootstrap method [35] was used:

- **for** $j = 1 \ldots n$
  - separate $\{y_i\}$ into $\{s_i\}$ and $\{c_i\}$
  - let $\{s_i^j\}$ and $\{c_i^j\}$ be random subsamples (with replacement) of $\{s_i\}$ and $\{c_i\}$
  - calculate t_j, the t-value of the separation between $\{s_i^j\}$ and $\{c_i^j\}$
- estimate the standard deviation of $\{t_j\}$:
  $$\sigma = \sqrt{\tfrac{1}{n}\textstyle\sum_{j=1}^{n}[t_j - \mu]^2}, \quad \mu = \tfrac{1}{n}\textstyle\sum_{j=1}^{n} t_j$$

The distribution of t values was obtained for both the MDL and SPHARM models using $n = 1000$ bootstrap samples. A t-test shows that the difference between the

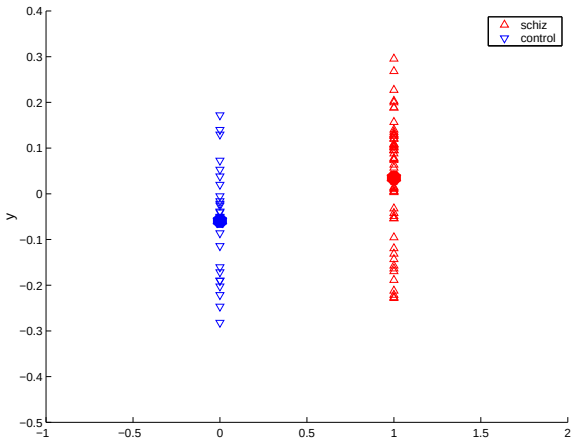


Figure 13.2. Leave-one-out projections of the control and schizophrenic data onto the discriminant vector using the MDL model.

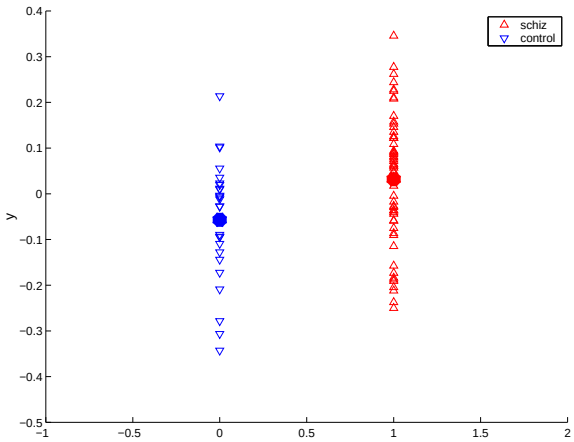


Figure 13.3. Leave-one-out projections of the control and schizophrenic data onto the discriminant vector using the SPHARM model.

models is highly significant ($p \approx 10^{-13}, t = 7.027$). This suggests that structural noise in the SPHARM model partially conceals the discriminating information present in the data.

13.3 Discussion

This chapter has shown how the statistical shape models described in this thesis can be used to characterise and quantify shape differences between two subsets of a class of objects. The method was applied to discriminating between the hippocampal shape of healthy and schizophrenic subjects. It was shown that the MDL model provided better discrimination than the SPHARM model of Gerig *et al.* – arguably the best existing method of constructing 3D statistical shape models.

Although the results show that neither model can be used to classify subjects reliably on an individual basis, both can be used to discriminate between populations. This could have important application in disease progression studies and drug trials. The results show that fewer subjects would be required to achieve a given level of significance using the MDL model rather than the SPHARM model (10% fewer subjects for this data), an important practical consideration.

It was assumed that a single multivariate Gaussian model was sufficient to model both classes of hippocampi. Two separate models could, in principle, have been used but this introduces the complication of defining a correspondence between

the two groups. The vectors of the groups would also be of different lengths, making comparison between them difficult. We intend that this will be the subject of future work along with an investigation of alternative discriminating statistics (e.g. the mean absolute difference to the class mean [41]).

For this study, the two schizophrenic subgroups (early and chronic) were combined to form a single group. A stronger discrimination might be achieved if the chronic subgroup was compared to the control group. It would also be interesting to investigate the shape differences between the early and chronically ill patients as this would give the shape variation that characterises the onset of schizophrenia.

Chapter 14

Conclusions

"Don't worry about the world coming to an end today. It's already tomorrow in Australia"
– *Charles Schultz*

This thesis has described a practical method of building both 2D and 3D statistical shape models automatically. This final chapter summarises the contributions and discusses ideas for future work.

14.1 Summary of Contributions

A principled approach to defining correspondences.
A generic optimisation framework has been presented based on the idea of selecting the correspondences that build the best model. The framework has been applied to automatically construct statistical models of shape.

An efficient method of manipulating correspondences.
It has been shown how correspondences can be manipulated by re-parameterising each training shape. This approach allows the correspondences to be straightforwardly constrained to be legal. Piecewise linear and continuous representations of re-parameterisation have been developed for both 2D (open and closed curves) and 3D (closed surfaces) objects.

An objective function that provides a principled measure of model utility.
A minimum description length (MDL) objective function has been derived that favours models with good specificity, generalisation ability and compactness.

An efficient method of optimisation.
An algorithm has been developed that is capable of routinely constructing shape models directly from large training sets of pre-segmented examples.

A demonstration of utility in a practical application.
The method has been applied to the problem of finding shape differences between the hippocampi of normal and schizophrenic subjects, demonstrating improved discrimination compared to the leading method in the literature.

14.2 Extension: MDL Appearance Models

There are two main limitations of the approach to automatic shape modelling described in this thesis:

- the topology of the modelled class of objects cannot vary;
- the training images must be segmented.

There are many cases where it is necessary to model objects of variable topology. For example, osteoarthritis (OA) causes articulating cartilage of load-bearing joints to erode thus changing the topology of the examples. To model such objects, the shape model must be able to accommodate topological changes.

Both limitations could be overcome by combining shape information with image intensity information to construct MDL *appearance models*. Appearance models [19] are normally constructed from a training set of corresponded images as follows:

- Define a warp function that registers each training image to a shape normalised coordinate frame
- Perform principal component analysis (PCA) on the shape data (i.e. the warp functions) to create a shape model.
- Perform PCA on the image data (in the shape normalised coordinate frame) to create an intensity model.
- Perform a further PCA on the shape and appearance parameter vectors[1] to create an appearance model.

The optimisation framework described in this thesis can be extended straightforwardly to automatically create such models:

[1] Finite differencing can be used to adjust the shape and appearance parameters to have the same units [19].

- The description length of the combined appearance parameter vectors can be calculated using (6.39) to give the total description length of the training set.
- The warp functions can be represented by extending the formulations of 1D curve re-parameterisation derived in §8.1.1 and §8.1.2, as described in §11.2.3.
- The optimal model can be found by optimising the MDL objective function with respect to the set of warp functions by extending the approach described in chapter 9.

14.3 Final Conclusions

This thesis has described a method of constructing optimal shape models *directly* from a training set of example shapes. The method has a theoretical grounding in information theory, has no arbitrary parameters and requires no user intervention (although this is easily incorporated if required). The effectiveness of the method has been evaluated empirically to show that it constructs significantly better models than alternative approaches.

Statistical shape models have been used widely to model and segment many classes of object and are now deployed both clinically and commercially. Their application has, however, been limited mainly to 2D objects, where manually annotated landmarks are easily obtained. The work presented here should allow much wider application of shape models, particularly to 3D biomedical objects.

Appendix A

Genetic Algorithm Optimisation

This appendix provides a brief overview of genetic algorithm (GA) optimisation as used in the work described in chapter 7. Only the aspects that are directly relevant to the problem at hand are discussed here, for a more general treatment see [74] or [42].

A genetic algorithm is a stochastic, global optimisation method based on the principles of natural selection. Much of terminology is borrowed from biology. The optimisation variables are called *genes* and a combination of genes makes a *chromosome*. A number of solution hypotheses (chromosomes) are generated and called a *population*. An objective function (sometimes called a fitness, or cost function), which is a function of the optimsation variables, measures the 'fitness' of each chromosome. A fitter individual (chromosome) has a higher probability of survival.

The idea behind GA search is that that when good chromosomes are continually combined, they will eventually produce the 'optimal' chromosome. The genetic operators repeatedly refine the population until a termination criterion is met. The following is an example of a simple genetic algorithm:

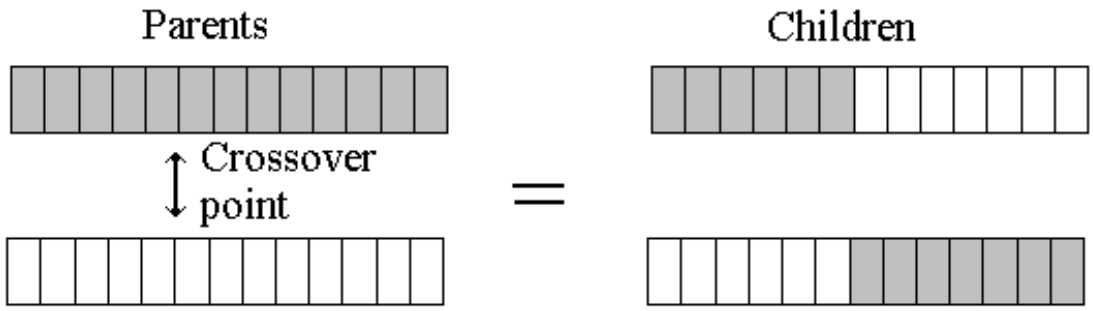


Figure A.1. How a crossover operator creates two child chromosomes from two parent chromosomes. The crossover point is the sixth gene.

- generate the initial population (randomly initialise chromosomes)
- **repeat**
  - calculate the fitness of each chromosome
  - build a new population using a selection operator
  - randomly select $N_{crossover}$ pairs of chromosomes from the population and use a crossover operator to generate new pairs
  - randomly select $N_{mutation}$ chromosomes and use a mutation operator to generate new chromosomes
- **until** convergence

Crossover, mutation and selection are discussed below.

A.1 Crossover Operator

A single crossover operator was used to obtain the results reported in this thesis. The crossover operator is analogous to the mating process of natural evolution. It combines the genes of two 'parent' chromosomes to produce two new, 'child' chromosomes. This is done by randomly selecting a 'crossover point' where each chromosome is split and combined to form two children. This is illustrated in figure A.1, where the crossover point is the sixth gene.

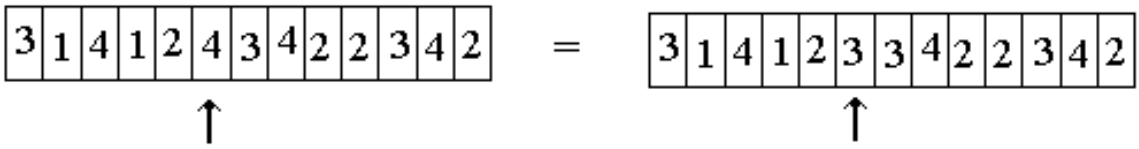


Figure A.2. How a mutation operator creates a new chromosome by replacing the value of a gene. The mutation point is the sixth gene.

A.2 Mutation Operator

The mutation operator ensures that the population remains genetically diverse. It takes a single chromosome and mutates a gene to produce a new individual. An example is shown in figure A.2, where the sixth gene is chosen and replaced.

A.3 Selection Operator

As the name suggests, the selection operator selects a number of chromosomes to build a new population. Because of the high-dimensionality of the solution space for our problem, the algorithm may converge prematurely unless a large population is chosen. Premature convergence can be avoided by applying minimal 'selection pressure'. We achieve this by 'sigma scaling' the fitness values of the chromosomes. Sigma scaling adjusts the fitness of the i^{th} chromosome to:

$$F_i' = \begin{cases} 1 + \frac{F_i - \bar{F}}{2\sigma} & \text{if } \sigma \neq 0; \\ 1 & \text{if } \sigma = 0; \end{cases} \tag{A.1}$$

where $\bar{F}$ is the mean fitness and σ is the standard deviation. The chromosomes of the new population are selected randomly with a probability proportional to their new fitness values, F_i'.

Appendix B

Uniform Tessellation of a Sphere

It is not strictly possible to position an arbitrary number of equidistant points on the surface of a sphere. An approximation, that is sufficient for our purposes, can be achieved by recursively subdividing a polyhedron (initially an octahedron) and projecting the points onto the sphere surface. At each level of recursion, each triangle is divided into 4 smaller triangles by placing 3 new vertices halfway along each edge and connecting them in a triangle as shown in figure B.1. Figure B.2 shows the tessellation after 3 iterations of the algorithm – giving 258 vertices and 512 faces.

This approach cannot generate an arbitrary number of points. If an arbitrary number of points are required, a method such as [27] can be used.

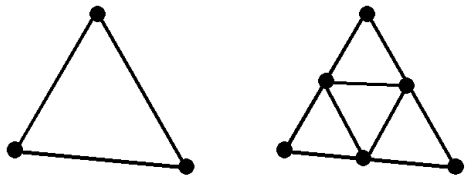


Figure B.1. How each triangle is subdivided. *Left*: initial triangle, *right* the four new triangles.

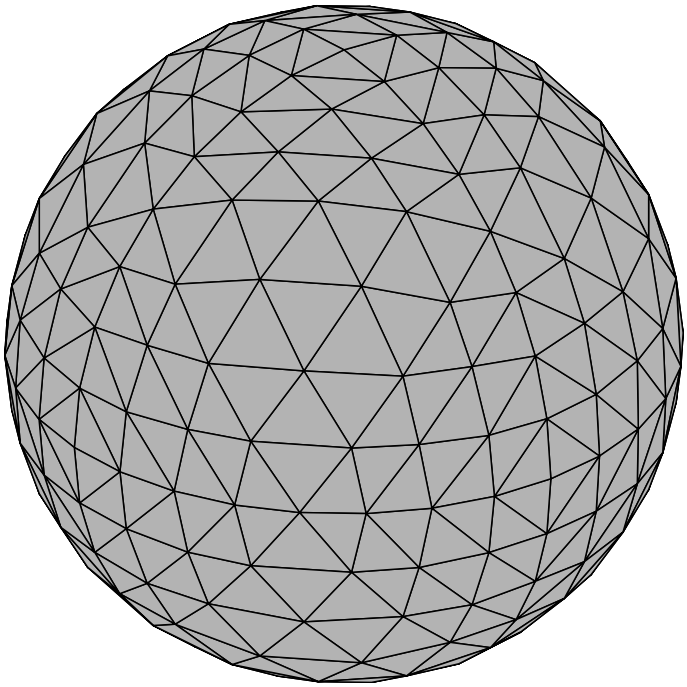


Figure B.2. A uniform tessellation of a sphere using three iterations of the subdivision algorithm.

Appendix C

Generating Random Points on a Sphere

We wish to generate random points on the sphere with a uniform probability over the entire surface. A simple method of doing this, using a unit sphere and Cartesian coordinates, is as follows[1]:

- Randomly sample the z coordinate from a uniform distribution of range $[-1, 1]$
- Randomly sample t from a uniform distribution of range $[0, 2\pi]$
- $r = \sqrt{1 - z^2}$
- $x = r\cos(t)$
- $y = r\sin(t)$

[1] adapted from entry 6.08 of the comp.graphics.algorithms FAQ. Available from http://www.faqs.org/faqs/graphics/algorithms-faq/

Bibliography

[1] S. Angenent, S. Haker, A. Tannenbaum, and R. Kikinis. On the laplace-beltrami operator and brain surface flattening. *IEEE Transactions on Medical Imaging*, 18:700–711, 1999.

[2] J. Ashburner and K. J. Friston. Voxel-based morphometry – the methods. *Neuroimage*, 11:805–821, 2000.

[3] Bajcsy and A. Kovacic. Multiresolution elastic matching. *Computer Vision, Graphics and Image Processing*, 46:1–21, 1989.

[4] A. Baumberg and D. Hogg. Learning flexible models from image sequences. In J.-O. Eklundh, editor, 3^{nd} *European Conference on Computer Vision*, volume 1, pages 299–308. Springer-Verlag, Berlin, 1994.

[5] A. Baumberg and D. Hogg. An adaptive eigenshape model. In D. Pycock, editor, 6^{th} *British Machine Vision Conference*, pages 87–96. BMVA Press, Sept. 1995.

[6] P. J. Besl and N. D. McKay. A method for registration of 3d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[7] F. L. Bookstein. Landmark methods for forms without landmarks: morphometrics of group differences in outline shape. *Medical Image Analysis*, 1(3):225–244, 1997.

[8] C. Brechbühler, G. Gerig, and O. Kübler. Parameterisation of closed surfaces for 3-D shape description. *Computer Vision, Graphics and Image Processing*, 61:154–170, 1995.

[9] C. Brechbühler-Miskuv. *Decription and analysis of 3-D shapes by parameterization of closed surfaces*. PhD thesis, Swiss Federal institute of technology, Zurich, Switzerland, 1995.

[10] A. Brett and C. Taylor. Construction of 3d shape models of femoral articular cartilage using harmonic maps. In *International Conference on Medical image computing and computer aided intervention*, pages 1205–1214, 2000.

[11] A. D. Brett, A. Hill, and C. J. Taylor. A method of automatic landmark generation for automated 3d pdm construction. *Image and Vision Computing*, 18:739–748, 2000.

[12] A. D. Brett and C. J. Taylor. A method of automatic landmark generation for automated 3d pdm construction. In P. Lewis and M. Nixon, editors, 9^{th} *British Machine Vision Conference*, volume 2, pages 914–923, Southampton, UK, Sept. 1998. BMVA Press.

[13] A. Caunce and C. J. Taylor. Using local geometry to build 3d sulcal models. In 16^{th} *Conference on Information Processing in Medical Imaging*, pages 196–209, 1999.

[14] M. Chakos, S. Schobel, H. Gu, G. Gerig, C. Charles, and J. Lieberman. Duration of illness and treatment effects on hippocampal voulume in male patients with schizophrenia. Research report, University of North Carolina at Chapel Hill, Feb. 2002.

[15] G. E. Christensen, S. C. Joshi, and M. Miller. Volumetric transformation of brain anatomy. *IEEE Trans. Medical Image*, 16:864–877, 1997.

[16] S. Chua and P. McKenna. Schizophrenia – a brain disease? a critical review of structural and functional cerebral abnormality in the disorder. *Br J Psychiatry*, 166:563–582, 1995.

[17] I. Cohen, N. Ayache, and P. Sulger. Tracking points on deformable objects using curvature information. In G. Sandini, editor, 2^{nd} *European Conference on Computer Vision*, pages 458–466. Springer-Verlag, May 1992.

[18] L. Cohen. On active contours and balloons. *Computer Vision, Graphics and Image Processing*, 53:211–218, 1991.

[19] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In H.Burkhardt and B. Neumann, editors, 5^{th} *European Conference on Computer Vision*, volume 2, pages 484–498. Springer, Berlin, 1998.

[20] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Comparing active shape models with active appearance models. In T. Pridmore and D. Elliman, editors, 10^{th} *British Machine Vision Conference*, volume 1, pages 173–182, Nottingham, UK, Sept. 1999. BMVA Press.

[21] T. F. Cootes and C. J. Taylor. Combining point distribution models with shape models based on finite-element analysis. *Image and Vision Computing*, 13(5):403–409, 1995.

[22] T. F. Cootes and C. J. Taylor. Data driven refinement of active shape model search. In 7^{th} *British Machine Vision Conference*, pages 383–392, Edinburgh, UK, 1996.

[23] T. F. Cootes and C. J. Taylor. A mixture model for representing shape variation. In A. Clarke, editor, 8^{th} *British Machine Vision Conference*, pages 110–119. BMVA Press, Essex, Sept. 1997.

[24] T. F. Cootes, C. J. Taylor, D. Cooper, and J. Graham. Training models of shape from sets of examples. In D. Hogg and R. Boyle, editors, 3^{rd} *British Machine Vision Conference*, pages 9–18. Springer-Verlag, Sept. 1992.

[25] T. F. Cootes, C. J. Taylor, D. Cooper, and J. Graham. Active shape models – their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, Jan. 1995.

[26] I. Craw and P. Cameron. Face recognition by computer. In D. Hogg and R. Boyle, editors, 3^{rd} *British Machine Vision Conference*, pages 489–507. Springer-Verlag, London, 1992.

[27] J. Cui and W. Freeden. Equidistribution on the sphere. *SIAM J. Sci. Comput*, 18:595–609, 1997.

[28] R. H. Davies, T. F. Cootes, C. J. Twining, and C. J. Taylor. An information theoretic approach to statistical shape modelling. In 12^{th} *British Machine Vision Conference*, pages 1–10, 2001.

[29] R. H. Davies, C. J. Twining, P. D. Allen, T. F. Cootes, and C. J. Taylor. Building optimal 2d statistical shape models. *Image and Vision Computing*, in press.

[30] R. H. Davies, C. J. Twining, T. F. Cootes, J. C. Waterton, and C. J. Taylor. 3d statistical shape models using direct optimisation of description length. In 7^{th} *European Conference on Computer Vision*, pages 3–21, 2002.

[31] R. H. Davies, C. J. Twining, T. F. Cootes, J. C. Waterton, and C. J. Taylor. A minimum description length approach to statistical shape modelling. *IEEE Transactions on Medical Imaging*, 21(5):525–537, May 2002.

[32] I. Dryden and K. V. Mardia. *The Statistical Analysis of Shape*. Wiley, London, 1998.

[33] N. Duta, A. Jain, and M. Dubuisson-Jolly. Automatic construction of 2d shape models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:433–445, 2001.

[34] G. J. Edwards, A. Lanitis, C. J. Taylor, and T. F. Cootes. Statistical models of face images: Improving specificity. In 7^{th} *British Machine Vision Conference*, pages 765–774, Edinburgh, UK, 1996.

[35] B. Efron, editor. *The Jacknife, the bootstrap and other resampling plans*. Society for Industrial and Applied Mathematics, 1998.

[36] N. I. Fisher. *Statistical analysis of circular data*. Cambridge University Press, 1993.

[37] M. Fleute and S. Lavallee. Building a complete surface model from sparse data using statistical shape models: Application to computer assisted knee surgery. In *International Conference on Medical image computing and computer aided intervention*, pages 878–887, 1998.

[38] A. F. Frangi, D. Rueckert, J. A. Schnabel, and W. J. Niessen. Automatic 3d asm construction via atlas-based landmarking and volumetric elastic registration. In 17^{th} *Conference on Information Processing in Medical Imaging*, pages 77–91, 2001.

[39] B. Geiger. Three-dimensional modelling of human organs and its application to diagnosis and surgical planning. Technical report no. 2105, Institut National de Recherche en Informatique et Automatique, Dec. 1993.

[40] G. Gerig, M. Styner, M. Chakos, and J. Lieberman. Hippocampal shape alterations in schizophrenia: results of a new methodology. In 11^{th} *Biennial Winter Workshop on Schizophrenia*, 2002.

[41] G. Gerig, M. Styner, D. Jones, D. Weinberger, and J. Lieberman. Shape analysis of brain ventricles using spharm. In *Mathematical Methods in Biomedical Image Analysis*, pages 171–178, 2001.

[42] D. E. Goldberg. *Genetic Algorithms in Search, Optimisation and Machine Learning*. Addison-Wesley, Wokingham,UK, 1989.

[43] P. Golland, W. E. L. Grimson, M. E. Shenton, and R. Kikinis. Small sample size learning for shape analysis of anatomical structures. In *International Conference on Medical image computing and computer aided intervention*, pages 72–82, 2000.

[44] J. Hajnal, D. Hawkes, and D. Hill, editors. *Medical Image Registration*. CRC Press, 2001.

[45] J. Haller, A. Banerjee, G. Christensen, M. Gado, S. Joshi, M. Miller, Y. Sheline, M. Vannier, and J. Csernansky. Three-dimensional hippocampal mr morphometry with high- dimensional transformation of a neuroanatomic atlas. *Radiology*, 202:504–510, 1997.

[46] D. J. Hand. *Discrimination and Classification*. John Wiley and Sons, 1981.

[47] A. Hill and C. J. Taylor. Model-based image interpretation using genetic algorithms. In P. Mowforth, editor, 2^{nd} *British Machine Vision Conference*, pages 265–274. Springer-Verlag, Sept. 1991.

[48] A. Hill and C. J. Taylor. Automatic landmark generation for point distribution models. In E. Hancock, editor, 5^{th} *British Machine Vision Conference*, pages 429–438. BMVA Press, Sept. 1994.

[49] A. Hill, C. J. Taylor, and A. D. Brett. A framework for automatic landmark identification using a new method of non-rigid correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):241–251, 2000.

[50] A. Hill, A. Thornham, and C. J. Taylor. Model-based interpretation of 3d medical images. In J. Illingworth, editor, 4^{th} *British Machine Vision Conference*, pages 339–348. BMVA Press, Sept. 1993.

[51] C. Houck, J. Joines, and M. Kay. A genetic algorithm for function optimization: A matlab implementation. Technical Report NCSU-IE TR 95-09, North Carolina State University, Department of Industrial Engineering, 1995.

[52] M. K. Hurdal, P. L. Bowers, K. Stephenson, D. W. L. Sumners, K. Rehm, K. Schaper, and D. A. Rottenberg. Quasi-conformal flat mapping the human cerebellum. In *International Conference on Medical image computing and computer aided intervention*, pages 279–286, 1999.

[53] K. C. K and J. Ghosh. A unified model for probabilistic principal surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:22–41, 2001.

[54] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In 1^{st} *International Conference on Computer Vision*, pages 259–268, London, June 1987.

[55] A. Kelemen, G. Szekely, and G. Gerig. Elastic model-based segmentation of 3D neurological data sets. *IEEE-TMI*, 18(10):828–839, 1999.

[56] A. Kelemen, G. Székely, and G. Guido Gerig. Three-dimensional Model-based Segmentation. Technical Report 178, Image Science Lab, ETH Zürich, 1997.

[57] S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[58] A. Kotcheff and C. J. Taylor. Automatic construction of eigen-shape models by genetic algorithm. In 15^{th} *Conference on Information Processing in Medical Imaging*, 1997.

[59] A. C. W. Kotcheff and C. J. Taylor. Automatic construction of eigenshape models by direct optimisation. *Medical Image Analysis*, 2(4):303–314, 1998.

[60] M. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37:233–243, 1991.

[61] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the nelder-mead simplex method in low dimensions. *SIAM journal on Optimisation*, 9:112–147, 1998.

[62] S. Lawrie and S. Abukmeil. Brain abnormality in schizophrenia: a systematic and quantitative review of volumetric mri studies. *Br J Psychiatry*, 172:110–120, 1998.

[63] M. E. Leventon, W. E. L. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *IEEE conference on Computer Vision and Pattern Recognition Conference*, pages 316–323, 2000.

[64] P. Lipson, A. L. Yuille, D. O'Keeffe, J. Cavanaugh, J. Taaffe, and D. Rosenthal. Deformable templates for feature extraction from medical images. In O. Faugeras, editor, 1^{st} *European Conference on Computer Vision*, pages 413–417. Springer-Verlag, Berlin/New York, 1990.

[65] C. Lorenz and N. Krahnstover. Generation of point-based 3d statistical shape models for anatomical objects. *Computer Vision and Image Understanding*, 77:175–191, 2000.

[66] J. B. A. Maintz and M. A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, 1998.

[67] K. V. Mardia. *Statistics of Directional Data*. Academic Press, 1972.

[68] K. V. Mardia, J. Hent, J. Wiffen, and D. Shah. Hierarchical bayesian models for biological objects. Stat 96/11, Department of Statistics, University of Leeds, 1996.

[69] R. McCarley, C. Wible, M. Frumin, Y. Hirayasu, J. Levitt, I. Fisher, and M. Shenton. Mri anatomy of schizophrenia. *Biol Psychiatry*, 45:1099–1119, 1999.

[70] T. McInerney and D. Terzopoulos. Topologically adaptable snakes. In 5^{th} *International Conference on Computer Vision*, pages 840–845, 1995.

[71] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: a survey. *Medical Image Analysis*, 1(2):91–108, 1996.

[72] D. Meier and E. Fisher. Parameter space warping: Shape-based correspondence between morphologically different objects. *IEEE Transactions on Medical Imaging*, 21:31–47, 2002.

[73] D. Meyers, S. Skinner, and K. Sloan. Surfaces from contours. *ACM Trans. Graphics*, 11:228–258, 1992.

[74] M. Mitchell. *An Introduction to Genetic Algorithms*. MIT press, 1996.

[75] C. Nastar and N. Ayache. Fast segmentation, tracking and analysis of deformable objects. In 4^{th} *International Conference on Computer Vision*, pages 275–279, Berlin, May 1993. IEEE Computer Society Press.

[76] A. P. Pentland and S. Sclaroff. Closed-form solutions for physically based modelling and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):715–729, 1991.

[77] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C (2nd Edition)*. Cambridge University Press, 1992.

[78] M. Quicken, C. Brechbhler, J. Hug, H. Blattmann, and G. Szekely. Parameterization of closed surfaces for parametric surface description. In *IEEE conference on Computer Vision and Pattern Recognition Conference*, pages 354–360, 2000.

[79] A. Rangagajan, H. Chui, and F. L. Bookstein. The softassign procrustes matching algorithm. In 15^{th} *Conference on Information Processing in Medical Imaging*, pages 29–42, 1997.

[80] A. Redhead, A. Kotcheff, C. Taylor, M. Porter, and D. Hukins. An automated method for assessing routine radiographs of patients with total hip replacement. In *Proc. Institution of Mechanical Engineers*, page 211, 1997.

[81] J. R. Rissanen. A universal prior for integers and estimation by minimum description length. *Annals of Statistics*, 11:416–431, 1983.

[82] J. R. Rissanen. *Stochastic Complexity in statistical inquiry*. World Scientific, 1989.

[83] D. Rueckert, A. F. Frangi, and J. A. Schnabel. Automatic construction of 3d statistical deformation models using non-rigid registration. In *International Conference on Medical image computing and computer aided intervention*, pages 77–84, 2001.

[84] D. Rueckert, L. Sonada, C. Hayes, D. HIll, M. Leach, and D. Hawkes. Non-rigid registration using free-form deformations: Application to breast images. *IEEE Transactions on Medical Imaging*, 18:712–721, 1999.

[85] W. Schroder, editor. *The Visualization Toolkit An Object-Oriented Approach To 3D Graphics*. Prentice Hall, 1992.

[86] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen. Decimation of triangle meshes. *Computer Graphics*, 26(2):65–70, 1992.

[87] S. Sclaroff and A. P. Pentland. Modal matching for correspondence and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6):545–561, 1995.

[88] G. L. Scott and H. C. Longuet-Higgins. An algorithm for associating the features of two images. *Proc Roy Soc Lond*, 244:21–26, 1991.

[89] C. E. Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, 27:379–423 and 623–656, 1948.

[90] L. S. Shapiro and J. M. Brady. A modal approach to feature-based correspondence. In P. Mowforth, editor, 2^{nd} *British Machine Vision Conference*, pages 78–85. Springer-Verlag, Sept. 1991.

[91] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *J. Opt. Soc. Am. A*, 4:519–525, 1987.

[92] S. Solloway, C. Hutchinson, J. Waterton, and C. J. Taylor. The use of active shape models for making thickness measurement s of articular cartilage from MR images. *magnetic resonance in medicine*, 37:943–952, 1997.

[93] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, analysis, and machine vision*. Brooks/Cole, 1999.

[94] M. Spiegel and L. Stephens, editors. *Schaum's Outline of Statistics*. Schaum, 1998.

[95] L. H. Staib and J. Duncan. Parametrically deformable contour models. In *IEEE conference on Computer Vision and Pattern Recognition Conference*, pages 98–103, 1989.

[96] C. Studholme, D. Hill, and D. Hawkes. Automated 3d registration of mr and ct brain images by multireolution optimisation of voxel similarity measures. *Med Phys*, 24:25–35, 1997.

[97] C. Studholme, D. Hill, and D. Hawkes. An overlap invariant entropy measure of 3d medical image alignment. *Pattern Recognition*, 32:71–86, 1998.

[98] G. Szekely, A. Kelemen, C. Brechbuhler, and G. Gerig. Segmentation of 2-d and 3-d objects from mri volume data using constrained elastic deformations of flexible fourier contour and surface models. *Medical Image Analysis*, 1:19–34, 1996.

[99] G. Szeliski and S. Lavalée. Matching 3-D anatomical surface with non-rigid deformations using octree-splines. *International Journal of Computer Vision*, 18(2):171–186, 1996.

[100] H. D. Tagare. Shape-based nonrigid correspondence with application to heart motion analysis. *IEEE Transactions on Medical Imaging*, 18:570–579, 1999.

[101] D. Terzopoulos and D. Metaxas. Dynamic 3d models with local and global deformations: Deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):703–714, 1991.

[102] D. W. Thompson. *On growth and form*. Cambridge University Press, 1917.

[103] P. Thompson and A. Toga. Anatomically drive strategies for high-dimensional brain image warping and pathology detection. In *Brain Warping*, pages 311–336, 1999.

[104] A. W. Toga, editor. *Brain Warping*. Academic Press, San Diego, CA, USA, 1999.

[105] L. Wang, C. Sarang, M. Miller, U. Grenander, and J. Csernansky. Statistical analysis of hippocampal asymmetry. *Neuroimage*, 2002.

[106] Y. Wang, B. S. Peterson, and L. H. Staib. Shape-based 3d surface correspondence using geodesics and local geometry. In *IEEE conference on Computer Vision and Pattern Recognition Conference*, pages 644–651, 2000.

[107] J. C. Waterton, B. J. Middleton, R. Pickford, C. P. Allott, D. Checkley, and R. A. Keith. Reduced animal use in efficacy testing in disease models with use of sequential experimental designs. *Developments in Animal and Veterinary Sciences*, 31:737–745, 2000.

[108] A. Webb. *Stastical pattern recognition*. Arnold, 1999.

[109] I. Wright, S. Rabe-Hesketh, P. Woodruff, A. David, R. Murry, and E. Bullmore. Meta-analysis of regional brain volumes in chronic schizophrenia. *Am J Psychiatry*, 157:16–25, 2000.

[110] A. L. Yuille, D. Cohen, and P. Hallinan. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8:104–109, 1992.

[111] P. Zhu and P. M. Chirlian. On critical point detection of digital shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):737–748, 1995.

"If we knew what it was we were doing, it would not be called research, would it?"
– *Albert Einstein*.