Statistical Shape Modelling: Automatic Shape Model Building

Zihua Su

Submitted to the University College London for the Degree of

Doctor of Philosophy

Department of Medical Physics and Bioengineering

University College London

2011

Declaration

I (Zihua Su), confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Signature:

Table of Contents

List of Figures

List of Tables

Abstract

Statistical Shape Models (SSM) have wide applications in image segmentation, surface registration and morphometry. This thesis deals with an important issue in SSM, which is establishing correspondence between a set of shape surfaces on either 2D or 3D. Current methods involve either manual annotation of the data (current ‘gold standard’); or establishing correspondences by using segmentation or registration algorithms; or using an information technique, Minimum Description Length (MDL), as an objective function that measures the utility of a model (the state-of-the-art). This thesis presents in principle another framework for establishing correspondences completely automatically by treating it as a learning process. Shannon theory is used extensively to develop an objective function, which measures the performance of a model along each eigenvector direction, and a proper weighting is automatically calculated for each energy component. Correspondence finding can then be treated as optimizing the objective function. An efficient optimization method is also incorporated by deriving the gradient of the cost function. Experimental results on various data are presented on both 2D and 3D. In the end, a quantitative evaluation between the proposed algorithm and MDL shows that the proposed model has better Generalization Ability, Specificity and similar Compactness. It also shows a good potential ability to solve the so-called “Pile Up” problem that exists in MDL. In terms of application, I used the proposed algorithm to help build a facial contour classifier. First, correspondence points across facial contours are found automatically and classifiers are trained by using the correspondence points found by the MDL, proposed method and direct human observer. These classification schemes

are then used to perform gender prediction on facial contours. The final conclusion for the experiments is that MEM found correspondence points built classification scheme conveys a relatively more accurate gender prediction result.

Although, we have explored the potential of our proposed method to some extent, this is not the end of the research for this topic. The future work is also clearly stated which includes more validations on various 3D datasets; discrimination analysis between normal and abnormal subjects could be the direct application for the proposed algorithm, extension to model-building using appearance information, etc.

Acknowledgements

I would like to thank my first PhD supervisor Professor Andrew Todd-Pokropek for his original derivation of this project, outstanding support and supervision. My second PhD supervisor and my best friend Dr. Tryphon Lambrou made numerous and invaluable contributions to this work. His words are encouraging for both research and my life. Again, I would like to say thank you Tryphon. I also would like to mention my friends in the lab, Jing, Irving, Munir, and the other Richard. You guys made my life in London quite fascinating and joyful.

感谢我的父亲苏德昌和母亲孟凡青在给予我的帮助，支持，关怀，和爱。这篇文章献给你们，作为你们培养我多年一个成果。

另外，感谢我的岳父母还有我的妻子常卓对我的理解和关怀。

最后，我希望把这篇文章献给我的儿子，希望他能茁壮成长。

Chapter 1 Introduction

This PhD thesis deals with a fundamental issue in Statistical Shape Models (SSM), which is the “Correspondence Problem”. In previous approaches of using SSM, researchers [1] [13] have been using manual marking correspondences to facilitate this procedure. However, this subjective and error-prone manual work is against the spirit of image processing which is pursuing an automatic fashion. Recently, Minimum Description Length (MDL) [2] has emerged as the state-of-the-art approach to find correspondence points across datasets automatically. However, due to the pitfall of the MDL cost function, the original cost function often conveys wrong results, which have some of the landmarks congested or overlapped on one location. This behaviour is reported in several references [2], [32], [38], [49], and is named as “Pile Up” problem. Therefore, this drawback of the cost function will be the main issue discussed in the next few chapters. The author of MDL also proposed three objective metrics in order to evaluate automatic correspondence finding methods from different researchers. We will use those three metrics as the main evaluation methods to compare our proposed method with MDL.

In the scope of the thesis, we are aiming to develop a new framework of established correspondence across datasets automatically. The new framework should hold some desired features:

1) Automatic, no human inference is needed;

2) Flexible, the methods can work on both 2D and 3D datasets;

3) Robust, the proposed algorithm should have good potential to deal with complex shapes, in which case the state-of-the-art algorithm encounters "Pile Up" problem;

4) Better, by using the same accepted metrics, the new proposed method should convey better results compared with the state-of-the-art algorithm.

In this thesis, "correspondence" is referred to the meaningful anatomy correspondence points. An illustration of correspondence is shown in Figure 1.1

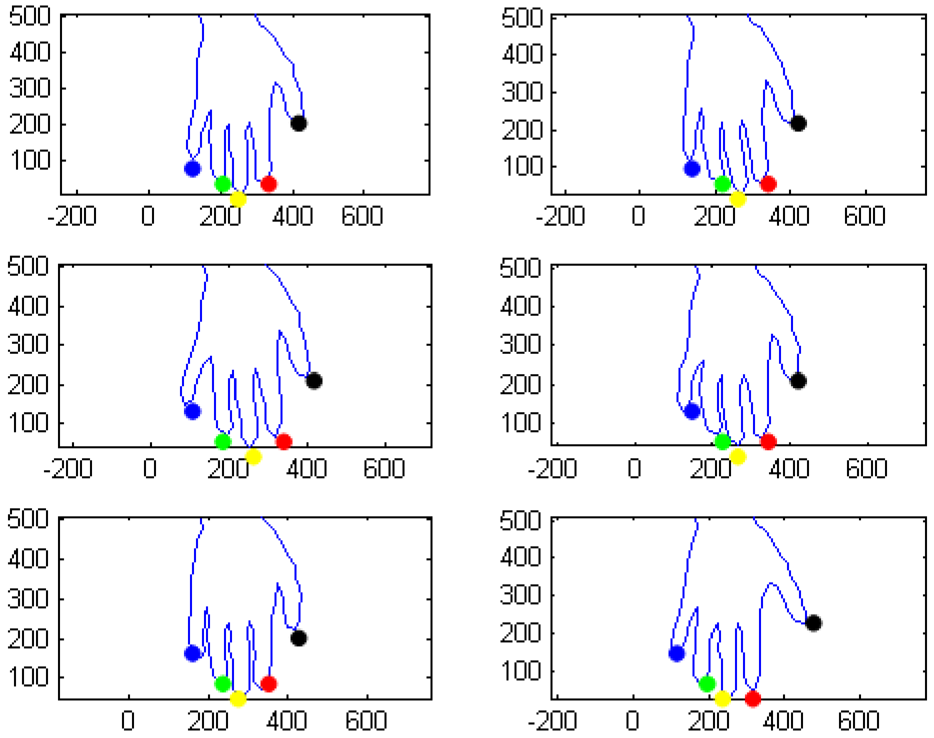


Figure 1.1 Six hand outlines are shown here, the "correspondence" in the thesis is interpreted as anatomy correspondence. For example, in this graph, fingertips of different hands are corresponding to each other. The correspondences are identified by using the same colour.

Applications of using SSM are vast. For example, the work can deal with image segmentation, registration, and shape modelling. This work can also be applied to both 2D and 3D, by using different shape parameterization schemes. The aim of the proposed work is to find an optimal automatic Statistical Shape Model building method, which can help the shape model achieve better performance in the applications we just mentioned. In this sense, we will introduce the applications of SSM in the next paragraphs, to Image Segmentation, Morphological Analysis, and Image Registration.

1.1 Model Based Shape Segmentation

In computer vision, segmentation refers to the process of partitioning a digital image into multiple segments (sets of pixels or voxels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image so that pixels with the same label share certain visual characteristics. Segmentation is also an important procedure for future image processing. For example, identifying and/or evaluating anatomical areas of interest, pre-processing for image registration, preparation for analysis with respect to functional metrics, preparation for surface extraction.

Segmentation can be performed manually by drawing along the area of interest by an expert. However, in practice, we will not use this method since it is prone to operator bias, fatigue, and is time-consuming. Although manual segmentation is recognized as the gold standard, Davies *et al* have shown that it will not always hold right under some evaluation properties [2] due to the operator’s bias, and making an algorithm run in an automatic manner is the ultimate goal for image processing.

Some low level methods, such as edge detection and region growing (for example snakes) will often be trapped in an invalid, incomplete and erroneous shape. They will perform even worse when parts of the boundary are missing. An additional remedy has to be performed for ensuring a valid shape. Shape constrains are often used for this purpose. For example, in 2D cases, researchers often make the shape contour first order or second order smooth. In the bottom of Figure 1.2, Canny Edge Detector (CED) is used to capture ROIs, which are hidden in the noise background. The key point for CED algorithm is that a Gaussian convolution is applied before edge detection. The convolution performs as a low pass filtering so that all the high frequency information is recognized as non-edge information. By designing a suitable low pass filter, the ideal edge information can be easily found by an edge detection method. From the segmentation results, we can see that ROI is still mixed with noise. Therefore, there is no guarantee for good quality segmentation. In addition, prior knowledge such as the cut off frequency for the low pass filter is hard to estimate.

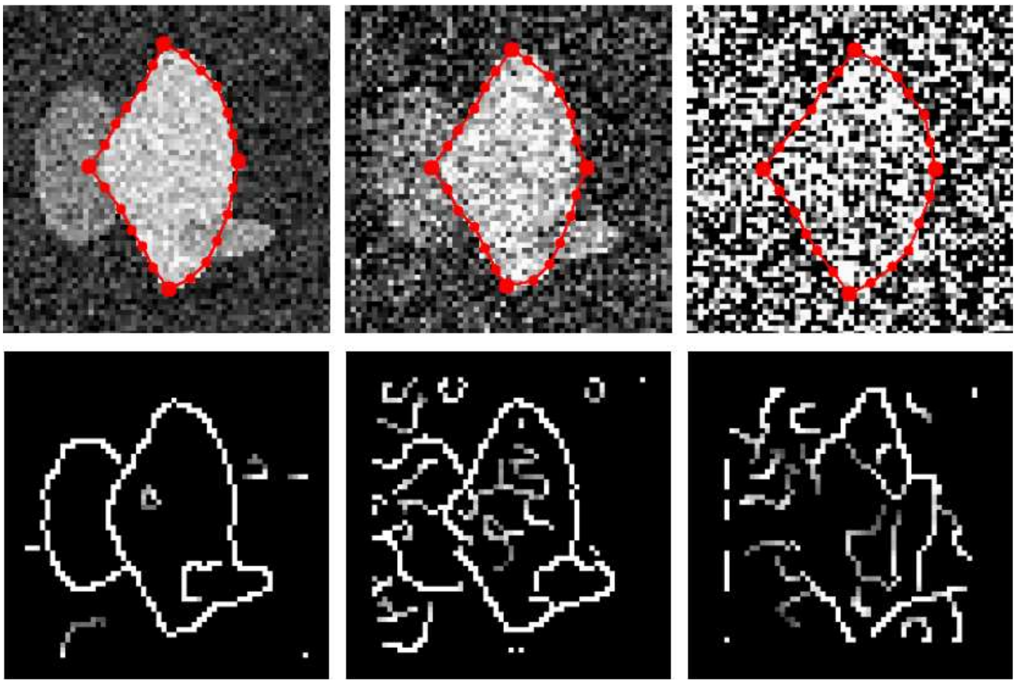


Figure 1.2 Top, left to right: images with Gaussian noise added with SNR 5.0, 2.5, 0.5 (with each shows the final contour found by ASM); Bottom, left to right: corresponding contour found by Canny edge detector [3]. This graph is from Reference [26].

A promising approach is to segment shapes using a priori information or knowledge. For example, properties of shape such as shape variations, position, scale, and rotation. Bearing this information in mind, we can achieve a resultant shape that looks the closest to valid shape(s) in the training set. This becomes useful in blurred or incomplete data, see Figure 1.2.

Another example of model-based segmentation is shown in Figure 1.3. Active Shape Model (ASM) is used for this particular task [99]. Statistical Shape Model based segmentation [37, 61] uses information from both edges from images and prior information of shapes. For example, the new shape segmented by ASM will look like the ones in the training set. Therefore, in the segmentation task, even though parts of the shape boundary is mixed in the image noise, a proper segmentation result can still be achieved.

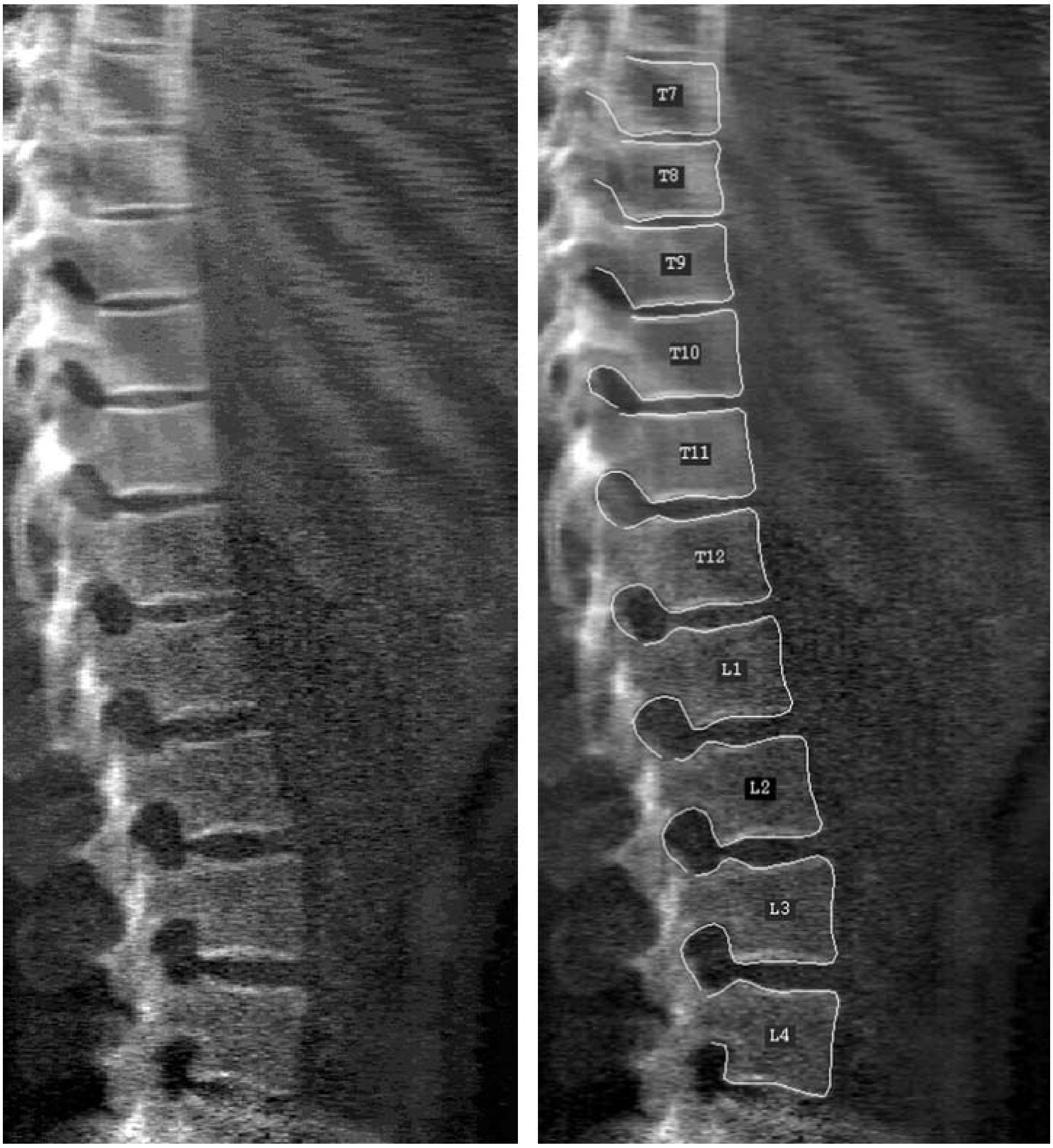


Figure 1.3 This graph shows a segmentation example performed by using Active Shape Model, **which is from reference [99]. Left: a DXA image shows the spine from the seventh thoracic vertebra (T7) to the fourth lumbar vertebra (L4). Right: the segmentation achieved by using the Active Shape Model.**

1.2 Shape Model for Morphological Analysis

Morphology comes from the calssical Greek concept *morphé*, meaning shape or form.

Morphology is the study of similarities and differences in the anatomy of organisms. Within this thesis, there are two major concepts of Morphological Studies:

1. Studies on homologous structures within the same group. For example, datasets collected from the same subject but in different time, it is quite useful to analyze shape changes within time.
2. Studies on homologous structures between groups with same genetic but different characteristics. In medical imaging, brain is an interesting subject, which attracted many researchers. One of the reasons is that, as an example, the structure of hippocampus is related with many illnesses. It is useful to identify the shape difference between subjects with illness and control/normal subjects [22], since it will become easier to identify and/or predict subjects with illness.

Morphology can have a direct application in the study of shape, which will be presented in this thesis. An example of morphological analysis is the study on hippocampus shape of healthy and schizophrenic patients [4] as in Figure 1.4. By using the Minimum Description Length algorithm proposed in [44], ASM can be used as a classification tool in examining the difference between the control subjects and a new dataset.

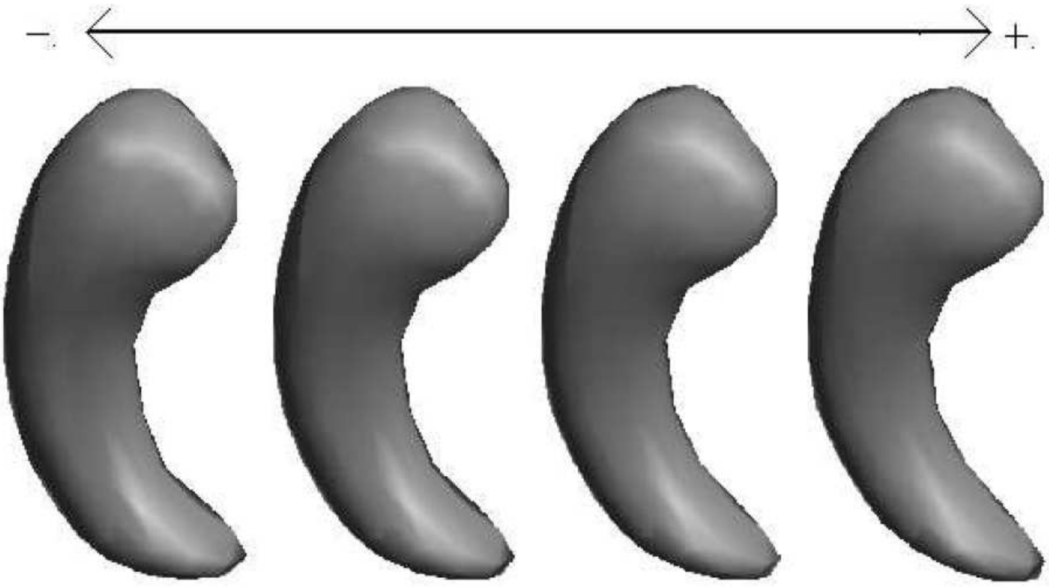


Figure 1.4 The shape variation discriminates between hippocampus shape of a group of healthy (-) and schizophrenic (+) patients. This graph is from [44].

Different approaches have been implemented for morphological studies, many of which are only applicable to sets of manually annotated landmarks. Although some automatic methods exist, they have their advantages and disadvantages. We wish to develop a new automatic algorithm, which can keep their excellence and reduce or get rid off the disadvantages of the previous automatic methods.

1.3 Shape Model of Surface-based Group Registration

Registration is the process of alignment of medical imaging data, usually for the purpose of comparisons or measurement [5]. The registration technique can be used in a broad range of applications. For example: image guided surgery, analysis of function images, characterization of normal and abnormal anatomical variability, detection of change in disease stages over time, visualization of multimodality data and atlas guidance for anatomical interpretation. Many previous algorithms rely heavily on landmarks that have been placed during scan or surface features like pixel intensity, curvatures. However, sometimes there are no locatable anatomical landmarks available. We want to build a framework that can find these landmarks or features automatically, so surfaces can be registered easily by using these landmarks.

1.4 Evaluations Methods

Validation is a very important process of checking whether the newly built model satisfies certain criteria, and compares the results with the ones from other models. Therefore, in the thesis, we have to compare our model with models built by other researchers. In 2D, most researchers compare the optimal results with manual landmarks, which is the so-called “Gold Standard”. However, this process is subjective and error prone. Moreover, it becomes almost impossible to mark data in 3D cases, where landmarks with high curvatures are rare. Fortunately, there are some general accepted approaches for evaluating different algorithms, namely Generalization Ability, Specificity and Compactness [22].

Briefly, Generalization Ability of a model measures its capability to represent unseen instances of the class of the object modelled. Specificity is the ability to measure whether the model can generate instances of the object that are close to those in the training set. Finally, Compactness is the ability to use fewer parameters to cover more shape instances in the training sets. Therefore, we can adopt these three approaches to measure the performance. More details about these three evaluation criteria will be presented in section 4.1.

1.5 Conclusions and Contributions

The main contributions of this thesis are listed as follows:

1) A new objective function, which provides a measurement of model utility

Different from the previous MDL approach, we propose using a new entropy measurement of model utility to find the correspondence points. During experiments, we discovered that there are some good features in our proposed approach compared with MDL approach.

2) The gradient of the MEM cost function is derived for a faster convergence

In this thesis, I also discuss the problem of deriving the gradient of the cost-function. With the help of gradient information, some more sophisticated optimization method can be used in the algorithm. Compared with optimizing the cost-function with Simplex, steepest gradient optimization method can achieve stabilization faster.

3) A more shape feature preserving shape parameterization and re-parameterization method

Unlike the simplified version of spherical harmonics used in the MDL approach by Davies, we use conformal mapping as our 3D shape parameterization method. If moving a point on the shape in the original shape space, the point in the corresponding sphere will move in the coherent direction. A more efficient re-parameterization method is used, unlike MDL approach, a local kernel is used therefore, a new kernel can be added without disturbing other landmarks.

4) The framework is applicable to both 2D and 3D datasets

The proposed algorithm does not have restriction on the dimensionality of the datasets; therefore, it can be flexibly applied to both 2D and 3D.

5) Comparisons with MDL algorithm based on objective metrics

Comparisons between MDL and the proposed method are performed on different datasets and different scenarios, both for 2D and 3D. In 2D, closed shapes, open shapes, and shapes with and/or without free ends are examined. In 3D, both artificial cuboids and real medical image datasets are used.

6) Using MEM method to perform automatic classification and make comparisons with similar schemes built by other methods

Classifiers are constructed by using correspondence points found by MEM, MDL and human manual landmark placing. These three schemes, together with direct human guessing, are performed on facial contours to perform gender prediction. The conclusion is that the MEM can help automatic building classification and make classification accuracy on the datasets tested relatively more accurate than the other three methods.

7) Solving the “Pile Up” problem

In the MDL approach, there is a well known “Pile Up” problem. When the problem happens all the points or parts of the points will pile up at some area. This is intrinsically wrong behaviour since that a points is found in the existing points and should not be overlapped with other points. This well reported problem is effectively eliminated by using the MEM cost-function.

1.6 Thesis Organization

The thesis will be organized as follows:

Chapter 2 Deformable Models.

Chapter 3 Building Statistical Shape Models.

Chapter 4 Solving the Correspondence Models.

Chapter 5 A 2D Minimum Entropy Approach.

Chapter 6 Experiments and Results in 2D

Chapter 7 A 3D Minimum Entropy Approach and Experiment Results

Chapter 8 Applications of Using MEM & MDL for Classification

Chapter 9 Conclusions and Future Work

Chapter 2 Deformable Models

In this chapter, we will describe briefly the available literature on the topics, which are relevant to this work. Over the years, a number of different approaches have been developed for deformable models.

2.1 Snakes

The use of deformable models in image segmentation became popular when Kass *et al* [6] created the ‘Snakes’. They describe an active contour model (so-called ‘snake’), that deforms inside of an image forming an ideal contour. The snake is driven by a combination of forces: an image force, an internal force and an external force.

The internal energy represents the smoothing forces on the curve, and the image forces represent the image-derived forces that contain the curve to take the shape of the feature. The image force attains a minimum when the snake matches an image contour. The snake converges when the forces achieve equilibrium.

The problem with the snake’s original version was that if the initial curve was not close enough to an edge, it had difficulty in being attracted to the optimal edge. In addition, the curves in the original snake had a tendency to shrink on themselves. To improve the convergence properties, dynamic programming can be used for energy minimization [7].

This also allows the objective function to include hard local constraints. Some robustness to the problem of finding a good initial guess was achieved by the addition of an extra force, which made the contour have a more dynamic behaviour. In another paper [8], the curve was considered to be a balloon that was being inflated. From an initial oriented curve, an extra pressure force was added, which pushed the curve out as if air was being introduced inside the closed contour. Model-based snakes [9] allow deformations based on a template model, but take its shape information into account only in a very limited and general way.

2.2 Thin-Plate Splines

Mathematically, a thin-plate spline $f(x,y)$ is a smooth function, which interpolates a surface that is fixed at the landmark points P_i at a specific height h_i. If one imagines this surface as a thin metal plate, then this plate will take a shape in which it is least bent. Bending energy is defined here as the integral over R^2 of the squares of the second derivatives,

$$I[f(x,y)] = \iint_{R^2} \left(f_{xx}^2 + 2 f_{xy}^2 + f_{yy}^2 \right) dxdy \qquad \textbf{(2.1)}$$

Bookstein [10] proposed this method. The solution for bending energy function is given by a solution of linear equations. In below we will discuss some implementation details of the Thin Plate Splines (TPS) algorithm. For example, there are two sets of points, which are corresponding to each other on different plates and each has *n* points.

$$\begin{Bmatrix} c_{i1} \\ c_{i2} \\ c_{i3} \end{Bmatrix}, i \in [1 \ldots n] \equiv C_{n\times 3} = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_n & y_n & z_n \end{bmatrix} \tag{2.2}$$

The TPS weights x and a can be obtained from solving the linear system:

$$\begin{bmatrix} K & P \\ P^T & O \end{bmatrix} \square \begin{bmatrix} w \\ a \end{bmatrix} = \begin{bmatrix} v \\ o \end{bmatrix} \tag{2.3}$$

The definitions of the symbols are given below:

$$\begin{gathered} K_{ij} = U\left(\left\| [c_{i1}, c_{i2}] - [c_{j1}, c_{j2}] \right\|\right) \\ U(r) = \begin{cases} r^2 \log(r) \,||\, r > 0 \\ 0 \,||\, r = 0 \end{cases} \end{gathered} \tag{2.4}$$

$$\begin{gathered} P = \begin{bmatrix} 1 & c_{11} & c_{12} \\ 1 & c_{21} & c_{22} \\ 1 & \vdots & \vdots \\ 1 & c_{n1} & c_{n2} \end{bmatrix}, O = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ v = \begin{bmatrix} c_{13} \\ c_{23} \\ \vdots \\ c_{n3} \end{bmatrix}, o = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}, a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \end{gathered} \tag{2.5}$$

Once the above linear system is solved, the coordinates along z-axis can be interpolated from:

$$z(x, y) = a_1 + a_2 x + a_3 y + \sum_{i=1}^{n} w_i U(\left\| [c_{i1}, c_{i2}] - [x, y] \right\|) \tag{2.6}$$

With bending energy given by:

$$F_b = w^T K w \tag{2.7}$$

The problem for this method is that a large number of homologous pairs of anatomic point landmarks should be set. So this method becomes difficult when there are not enough true landmark points that can be accurately marked. Landmarks are typically needed throughout the image because local changes to the spline have global effects. These landmarks must be reliable and robust, because the thin-plate spline forces these points exactly into correspondence. Also, while the transformation has interesting statistical properties, especially in terms of decomposing the warp, it can result in impossible wrappings. For example, the space can fold over itself, resulting in a non-homeomorphic warping, when the geometry is particularly warped. This technique is intended primarily as a method for the statistical comparison of shape through the location of homologous landmarks, and not as a general technique for image warping [11].

2.3 Statistical Shape Models

Statistical models try to capture the actual patterns of variability found in a class of objects, rather than making arbitrary assumptions. In this section, we will present a brief review of the Statistical Shape Models technique.

In 1989, Staib and Duncan [12] used a Fourier decomposition to form an orthogonal shape basis for a set of contours. Normalization 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 invalid examples, which means that the new example is not like any other in the training set. In addition, there may be important information in the boundary over a large range of frequencies, leading to a non-compact model.

In 1995, Cootes, *et al* [13，33] constructed the “Point Distribution Models (PDM)” from training sets of 2D boundaries. Given a set of labelled training examples, Procrustes Analysis [14] is first applied to minimize the sum of squared distances to the mean of the set. The aligned training set forms a cloud in the two dimensional space. Then, they used Principal Component Analysis (PCA) [15] to pick up the main axes of the cloud, and model the first few, which count for the majority of the variations. The shape model is then:

$$X = \overline{X} + Pb \tag{2.8}$$

Where $\overline{X}$ is the mean of the aligned training examples, P is a matrix whose columns are unit vectors along the principal axes of the cloud, and b is a $2n$ element vector of shape parameters with n is the number of points used in the examples. New shapes are generated by sampling independently from the distribution along each axis and reconstructing using the principal vectors. In most cases, Cartesian coordinates are sufficient, but in cases where parts of shapes can rotate, it may be useful to use angular coordinates instead [66].

In 1998, Cootes, *et al* [16] created a more powerful tool, the Active Appearance Model (AAM), which is a generalization of the widely used ASM approach. Rather than using only the shape contour coordinates, AAM uses all the information including the intensities in the image region covered by the target object. The model is generated by combining a model of shape variation with a model of the appearance variations from the training set. To build a model of the grey level appearance, they wrap each example images so that its control points match the mean shape. For reducing the global lighting variations, the grey information is filtered out by normalizing the examples. Again, PCA is applied to the normalized data; a new model can be obtained:

$$g = \overline{g} + P_g b_g \tag{2.9}$$

Where $\overline{g}$ is the mean moralized grey level vector, P_g is a set of orthogonal modes of variation and b_g is a set of grey level parameters.

The shape and appearance of any example can thus be summarized by the vectors b and b_g . For each example, the new shape vector is composed by:

$$\begin{pmatrix} W b \\ b_g \end{pmatrix} = \begin{pmatrix} WP^T (x - \overline{x}) \\ P_g^T (g - \overline{g}) \end{pmatrix} \tag{2.10}$$

2.4 Physical Models

First, there is no true physical model for deformation between individuals, for example, one individual's anatomical structure does not literally result from the deformation of another individual. Researchers use analogous physical models to enforce topological properties on the deformation. Without them, the results could be almost completely arbitrary.

Broit *et al* developed the Fluid model [17], which is based on the theory of elasticity developing restoring forces, which are proportional to the deformed distance. The basic idea for this method is that it simulates the physical deformation of fluid except for the smallest deformations. Such elastic transformations prevent the atlas from being fully deformed into the shape of the study. This slight flaw is overcome by the viscous fluid method [18], which allows the restoring forces to relax over time. For viscous fluids, the viscosity depends on the relevant velocity and scale length of the flow and the viscosity is the reciprocal of the Reynolds number.

The linear elastic model is based on Hooke's law. The loading modulus and unloading modulus are the same for the model. The restoring force holding the template together grows proportionately with the displacement from the original configuration of the atlas. The force is proportional to the displacement. Pentland and Sclaroff [19], and Nastar and Ayache [20] 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 generalized eigenproblem to obtain the 'modes of vibration' of the structure.

Therefore, Physical models try to give more intuitive shape variations, but the resulting shape is often "invalid"， which means the new generated shape does not look like what it should do. This is due to the fact that shape variations are not from prior knowledge of training set but from simulations of physical world. Therefore, this physical deformation does not necessarily result in a valid shape.

2.5 Group Registration

Image registration is not directly related to deformable models. However, since registration can result in a deformable model indirectly, for example, in references [102], [103], the authors used registration to find correspondence points of internal organs and use these correspondence points to construct a deformable model to analyze the functionality of the organs. Therefore, we will discuss the basic ideas behind registration algorithms. Image registration is the process of establishing point-by-point correspondence between two or more images of a scene. This process is very important in medical image processing. Many algorithms exist to solve this difficult problem. A good survey can be found in [56]. This family of algorithms are mainly composed of two parts: one is transformation and the other one is a metric for local matching.

In simple cases, the transform class $[x', y'] = \Gamma([x, y])$ can be defined by a set of parameters such as translation, angle of rotation, scale etc. A simple representation of rotation, scale and translation in 2D is shown below.

$$\begin{vmatrix} x' \\ y' \end{vmatrix} = a \begin{vmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{vmatrix} \begin{vmatrix} x \\ y \end{vmatrix} + \begin{vmatrix} t_x \\ t_y \end{vmatrix} \qquad \textbf{(2.11)}$$

Where parameter θ is the rotation angle; t_x and t_y are translations along x and y direction; a is the scale factor. It may be a linear affine transformation or a non-linear transformation. Shape deformation can also be represented by B-splines coefficients on a regular grid. The details on B-spline registration can be found in reference [57]. The physical model we mentioned in the previous paragraph can also be used as a deformation method.

The metric or body force encourage images to move and fit better to each other. Options for the metrics can be information theoretic measures e.g. mutual information, sum of squared difference or template intensity difference etc.

Based on the discussions above, many algorithms [58], [77], [78] have been proposed and published. For example, in [58], Crum et al, uses partial differential equations to model the properties of viscous water as a driving force.

2.6 Point Correspondence by Using Flow Field

Another big family of finding correspondence methods, is by using flow field theory to identify point correspondence between images. The shape correspondence can thus be achieved by defining a shape contour explicitly on one image. In this family of point correspondence, Optical Flow (OF) [89] and Particle Velocity Imaging (PIV) [90] showed good performance. These two algorithms were originally used to recover motion filed between images taken as times series, for example see references [91], [92], [93]. The difference between OF and PIV is that PIV simulates the nature of fluid physics. The PIV assumes that the fluid is in a tunnel (boundary condition) and the volume of the fluid is uncompressible (constrains). On the other hand, OF simulates the human instinct that motion is brought by and identified by motion of light. Therefore, the basic rule for OF is that the intensity between corresponding points should hold constant. However there are two main problems in OF and PIV methods. First, the intensity constancy will not always hold, even though Brox et al [94] have proposed using both intensity and intensity gradient as a remedy method, which he called "highest accuracy of method". However, the method seems to be quite sensitive to the weighting components that can force the results to be very random. Therefore, in order to cope with one particular application, the method has to be previously trained on some prepared samples. Moreover, in order to suppress the aperture problem, some extra constrain components, such as isotropic, anisotropic or bilateral filter, has to be incorporated. In this way, the method becomes more sensitive to weighting. In addition, the algorithm may not recover the motion field on edge areas; therefore, it is not a good option for performing shape analysis. In order to use this particular model, the weighting component has to be trained using the training set. Since our goal is to recover the correspondence field between segmented shape contours, we will not discuss more details about image correspondence.

2.7 Conclusions

Although each approach has its own distinguishing advantages and application conditions, the Active Shape Model (ASM) of Cootes *et al* have shown some good properties, which other algorithms do not have. For example: 1) Without using prior information of the shape, other methods or algorithms often results in a invalid shape which does not look like genetically the same as examples in the training set. 2) Unlike ASM, which have explicitly incorporated correspondence points into training set for use, other methods do not use and define the correspondence directly in the algorithm.

Despite of the advantages in the ASM, the critical difficulties in ASM on how to construct the correct correspondences across datasets automatically are not well explored. An ill-defined correspondence can result in ill alignment, so that a new shape represented by the model will turn out to be invalid and will not look like shapes in the training set. In practice, manual annotation can solve this problem. However, this time consuming and tedious work is very subjective and error prone. It may be possible for an expert to mark 2D datasets, but it becomes mission impossible in 3D. Therefore, an automatic correspondence building method becomes so demanding. This PhD thesis will discuss some of the relevant review work on how to find correspondence points automatically in 2D and 3D cases, and propose a new framework for solving the automatic correspondence-finding problem.

The new proposed method is to some extent parallel to and at the same time quite different from the current state-of-the-art algorithm Minimum Description Length (MDL) [2]. The similarity exists in the way that both the new proposed approach and MDL are using a “learning” process to find the optimal correspondence across the datasets. For example, the correspondence is found through optimizing the predefined cost function. The properties of the correspondence have already been predefined in the cost function. Therefore, optimizing the cost function will be equal to find the optimal correspondence given the current condition/datasets. There are two main differences between the proposed method and MDL. Firstly, the proposed method uses Entropy rather than Description Length as the main metric to measure the utilities of the Active Shape Model. Secondly, the new proposed method chooses an automatic calculated weighting component for each energy component rather than just summing them up as MDL does. In order to investigate the potential ability of the new proposed model, a quantitative analysis was conducted between our proposed model and the state-of-the-art model by using some generally accepted metrics, which will be disclosed in the next few chapters.

Chapter 3 Building Statistical Shape Models

3.1 Shape Parameterization

Statistical Shape Models can be extracted from a set of *n* shapes. Each example object is marked with a fixed number of landmarks in Cartesian coordinates ($x_i, y_i \quad i = 1,2,3\ldots n$), each of which marks a particular point on the object, which we call correspondence points. Then, the shape is represented by concatenating all the node coordinates into one single vector. The 2D case example of the shape vector is in Equation 3.1 and the extension to 3D is quite straightforward as in Equation 3.2.

$$x = (x_1, y_1, \ldots, x_n, y_n)^T \tag{3.1}$$

$$x = (x_1, y_1, z_1, \ldots, x_n, y_n, z_n)^T \tag{3.2}$$

The training examples are aligned into a common co-ordinate frame by using Procrustes Alignment [14] to reduce correspondence error introduced by rotation, translation and scaling. Shape error is directly measured by the absolute distance between corresponding points.

3.2 Procrustes Analysis

In this section, we will give out some details about the frequently used Procrustes

Analysis (PA) algorithm [14]. The PA algorithm has many forms; the Generalized Orthogonal Procrustes Analysis (GPA) is the most useful one. After landmarks are placed, GPA is used to filter out the effects from location, scale and rotation, and retain geometrical information. A general description of this algorithm is shown here:

1) *Select one example from the training set as the target shape. (i.e. the first shape in the set is taken as the approximate mean shape)*
2) *Align the shapes with reference to the target one*
    a. *Calculate the centroid of each shape*
    b. *Move all centroids to the origin*
    c. *Normalize each shape to unit size*
    d. *Rotate each shape to the newest approximate mean shape*
3) *Calculate the new approximate mean from roughly aligned shapes*
4) *Go back to step 2, if mean shapes from step 2 and 3 are different*
5) *The end (shapes are aligned)*

Centroid is achieved by calculating the mean of all landmarks on each shape. In here, we will discuss the rotation in 3-dimension case. Given a point with coordinate vector [x, y, z], the rotation will be attained by multiplying this vector with rotation matrix.

Rotation with angle *a* about x-axis is:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(a) & -\sin(a) \\ 0 & \sin(a) & \cos(a) \end{bmatrix} \tag{3.3}$$

Rotation with angle *b* about y-axis is:

$$\begin{bmatrix} cos(b) & 0 & \sin(b) \\ 0 & 1 & 0 \\ -\sin(b) & 0 & \cos(b) \end{bmatrix} \tag{3.4}$$

Rotation with angle *c* about z-axis is:

$$\begin{bmatrix} cos(c) & -\sin(c) & 0 \\ \sin(c) & \cos(c) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

Then, the rotation about a general axis can be achieved by a combination of the above rotations about x-axis, y-axis and z-axis.

An example of 2D Procrustes Analysis on datasets of facial profiles are shown in Figure 3.1.

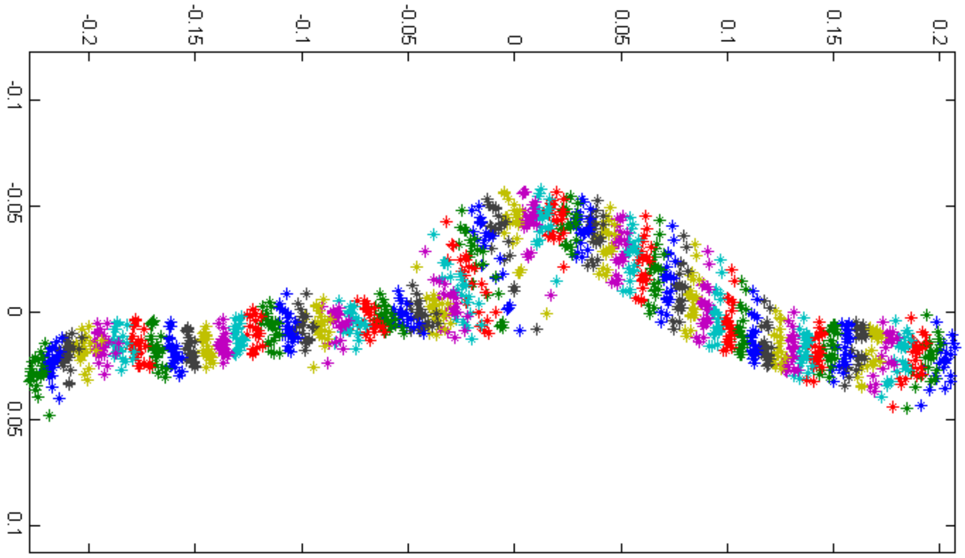


Figure 3.1 22 facial profiles are marked by an expert. They are aligned by using Procrustes Analysis. The corresponding landmarks can be identified by the same colour.

3.3 Statistical Analysis in Active Shape Modelling

The main contribution for the active shape modelling is achieved by Cootes *et al* [40].

They made an assumption that shape vectors x have a Gaussian distribution for the training set. We can build a linear model as follows

$$x = \overline{x} + Pb \tag{3.6}$$

Where, $\overline{x}$ is the mean shape, P is the orthogonal shape variations matrix, and b is the weighting shape vector.

The approach will be explained in details here. First, the mean shape can be calculated using Equation 3.7, where n_s is the number of samples. Shape vector is composed by concatenating landmark coordinates into a single vector.

$$\overline{x} = \frac{1}{n_s}\sum_{i=1}^{n_s} x_i \tag{3.7}$$

Then the normalized covariance matrix will be calculated from:

$$D = \frac{1}{n_s}\sum_{1}^{n_s}\left(x_i - \overline{x}\right)\left(x_i - \overline{x}\right)^T \tag{3.8}$$

At last, eigenvalue (λ^m) and eigenvector (p) will be extracted from matrix D

$$Dp^m = \lambda^m p^m \tag{3.9}$$

The eigenvalues are ranked naturally in a descending order according to their values. The eigenvector represents the directional variations for ASM and the normalised corresponding eigenvalue shows the proportion of variations is undergoing along this eigenvector. Normally, we choose the first M eigenvector to cover enough variations (say 98%).

It is calculated by Equation 3.10.

$$\frac{\sum_{i=1}^{M} \lambda_i}{\sum_{i=1}^{N_p} \lambda_i} \geq 98\% \qquad \textbf{(3.10)}$$

Where N_p, is the total number of shape variations captured by PCA and λ_i is the i^{th} eigenvalue from the shape covariance matrix.

3.4 Performance of Active Shape Model

An example of performance of ASM is shown here. We use a training set of 22 silhouettes of faces as in Figure 3.2.

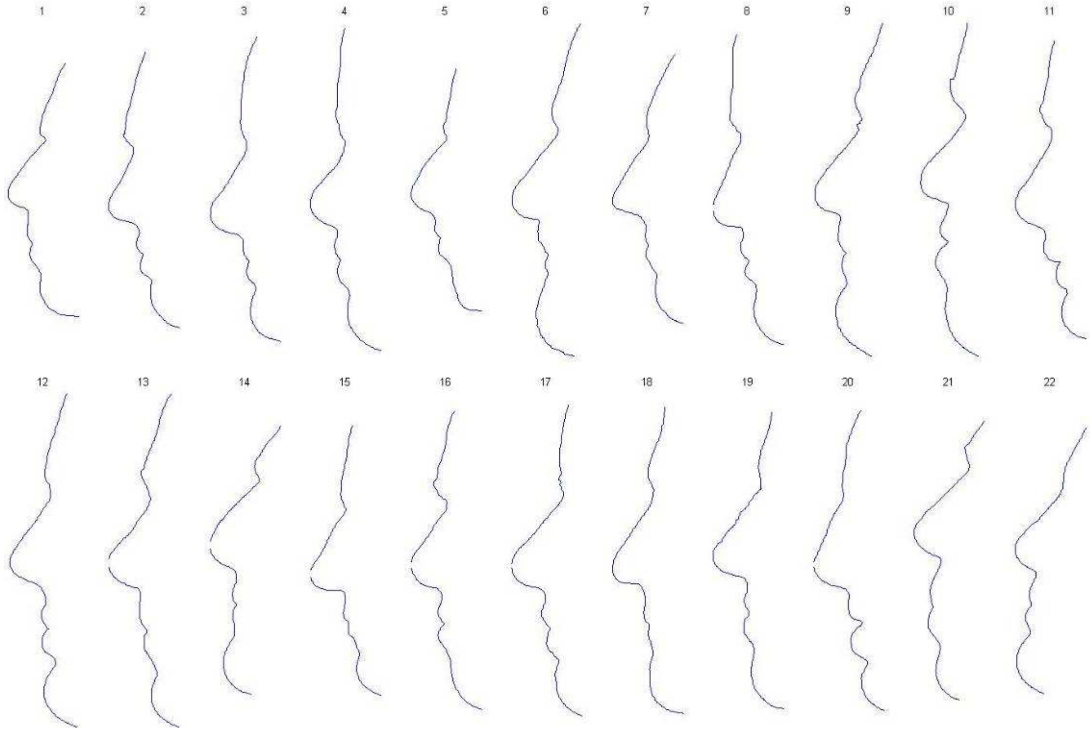


Figure 3.2 The training set of silhouettes.

These examples are taken by a digital camera. An expert segmented and marked these datasets by hands. (The datasets are from Thodberg after personal communication; these datasets are also used in reference [38]). From the description above, it can be concluded that by adjusting the weighting parameter in Equation 3.6., we can see the effect of different variations captured from training set (Figure 3.3).

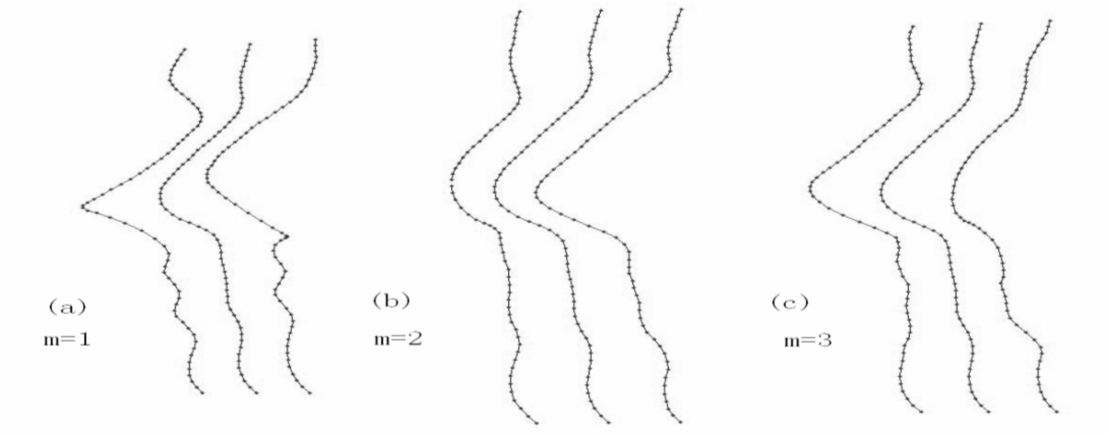


Figure 3.3 From left to right, this graph shows the effect of moving the first three weighting components independently. The shapes show that mean shape minus $3\sqrt{\lambda^m}$, mean shape and mean shape plus $3\sqrt{\lambda^m}$. M is the order of the weighting component.

Therefore, the new shape we want to match to, can be generated by selecting different values of $\{b^m\}$. We only select $\{b^m\}$ in the interval of $[-3\sqrt{\lambda^m}, 3\sqrt{\lambda^m}]$ for generating a valid shape. The probability distribution for this $\{b^m\}$ is assumed to be a Gaussian shape $P_{b^m} = \frac{1}{\sqrt{2\pi\lambda_m}} e^{-{(b^m)^2}/{2\lambda_m}}$. Figure 3.3 shows the first three modes of variation by independently varying the values of the first three shape parameters, (b^m, m=1, 2, 3) in equation (3.6), by $\pm 3\sqrt{\lambda^m}$. In this shape model, 65 marks are used and 7 modes can capture 98% of the variation of the training set (Equation 3.10).

3.5 Conclusions

In this chapter, we introduced the technique details of statistical shape model. The main idea behind Active Shape Model is to use Principal Component Analysis (PCA) to derive shape variations from training sets. Firstly, shape is represented by concatenated shape coordinates. After Procrustes Alignment shape difference introduced by translation, rotation and scaling has been removed. Therefore, after applying PCA to the shape covariance matrix, the shape variation distribution can be extracted easily. Finally, the new shape can be represented by using a mean shape plus shape variations with weighting component. From Figure 3.3, we can observe that the Active Shape Model can capture the true shape variations from the training set. However, the essential problem of using this technique is how to establish correspondence points automatically. Traditionally, manual labeling has been employed to tackle this problem. However, it is quite error prone, subjective and time consuming. Manual labeling on 3D datasets may take weeks to accomplish. Therefore, it is within the spirit of Computer Science to derive an automatic technique to find correspondence points automatically. In the next chapter, we will discuss in detail solving this correspondence problem.

Chapter 4 Solving the Correspondence Models

Point correspondence is a fundamental problem in using Active Shape Model to solve image processing problem. It is thus a critical problem, which can also be related to many medical imaging applications. For example, the whole work for registration either pair-wise or group-wise is finding the correspondence on either the surface or the interior of the object. With the correct correspondence, we can register different images from different modalities or different time acquisitions. Since information gained from different images in the clinical track of events can usually offer complementary nature. For segmentation problems, it is quite easy to solve; if we know the correspondence points and can construct a Shape Model. Normally, when the boundary is blurred or mixed with other tissue or organ, Shape Model can offer a better solution. In reference [59], readers can find some segmentation results by using Active Shape Model.

In this chapter, several approaches for solving this correspondence problem will be discussed. Especially, the last one (Minimum Description Length approach), which builds the correspondence in a learning process, has shown many advantages over other algorithms. Therefore, MDL will be the main target for comparison with our proposed method. Before introducing the techniques in literature, we start from some neutral and general accepted comparison criteria, which can be utilized to evaluate the performance of different techniques.

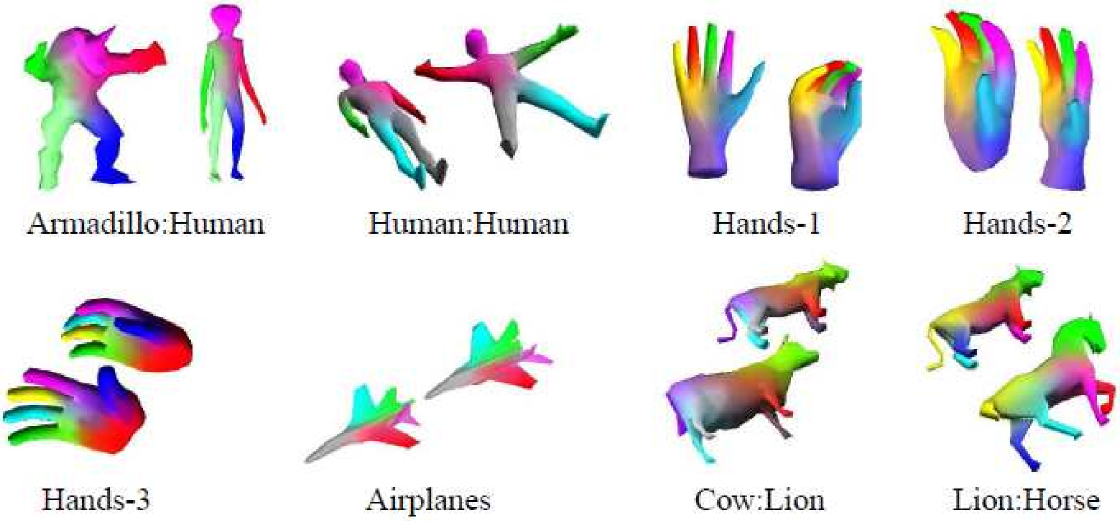


Figure 4.1 Shown is an example of “Corresopondence Problem”. The correspondence points are identified by using the same colour. These images are from reference [100].

4.1 Comparison Criteria

In the development of Statistical Shape Modelling Methods, due to lack of general accepted ground truth, it is very important for different researchers to have an objective basis for comparing different approaches. For example, in Figure 4.1, authors in reference [100] have found some correspondence results across shapes. The correspondence is identified by the same colour. However, just from subjective evaluation, it is quite difficult to evaluate the correspondence’s correctness. In this section, we will describe the ‘benchmark’ comparison criteria, which were first introduced in Davies’s paper [44].

The following paragraphs will describe three properties of an ideal model: Generalization Ability, Specificity and Compactness. All measures described allow meaningful comparisons between different models constructed from the same training set. Figure 4.2 will illustrate the notion of Generalization Ability and Specificity. On the left is the example of training sets. In the middle, the blue shows a kind of shape model, which can capture parts of the pentacle accurately, but not the whole pentacle. Therefore, the middle model is specific but not general. On the right, it shows another shape model which can cover all the shape examples in the training set but the shape model does not look like a pentacle. Therefore, the right model is general but not specific.

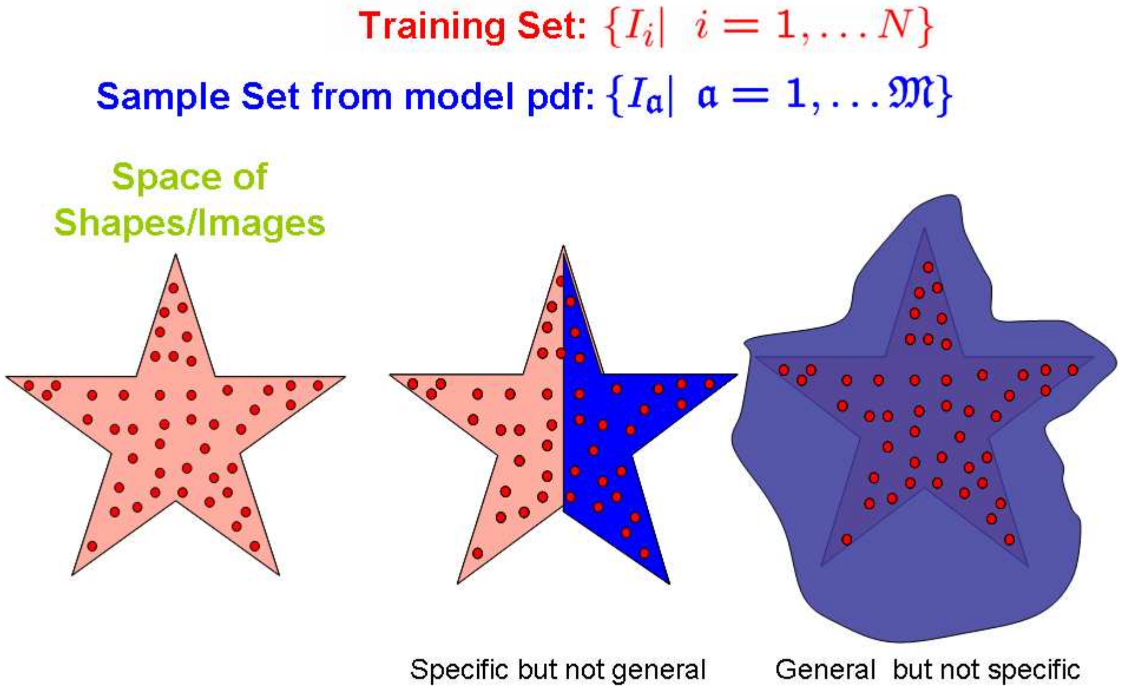

Training Set: $\{I_i| \ i = 1, ... N\}$
Sample Set from model pdf: $\{I_{\mathfrak{a}}| \ \mathfrak{a} = 1, ... \mathfrak{M}\}$
Space of Shapes/Images
Specific but not general
General but not specific

Figure 4.2 This graph is a brief and vivid introduction of the concept Generalization ability and Specificity. Red is training set and blue is sample set generated from model's probability distribution function.

4.1.1 Generalization Ability

The generalization ability (G) measures a model's ability to represent unseen instances of the class of object. This is a fundamental property as the goal of building a model is to fit the model to a new example. If a model is over-fitted to the training set, it will be unable to generalize to unseen examples.

A leave-one-out reconstruction can be used to measure the ability of each model from the training set. It means that a model is built using all but one member of the training set and then fitted to the one member excluded previously. The error with which the model can describe the unseen example is measured and the process is repeated until all examples are tested. The approximation error representing this generalization ability can be the averaged error over the complete set of examples.

The pseudo code below represents the whole process: the Generalization Ability is measured as a function of the number of shape parameters *M*, used in the reconstructions, N is the number of dataset members:

```
For M=1…N-2
  For i=1…N
       Build the ASM model from the training set, with $x_i$ removed;
       Reconstruct the unseen shape using M shape parameters:
                  $x_i'(M) = \bar{x}_i + \sum_{m=1}^{M} P_i^m b_i^m$
       Calculate the sum of squares approximation error:
                  $\varepsilon_i^2(M) = \left| x_i - x_i'(M) \right|^2$
       Calculate the mean squared error:
                  $G(M) = \frac{1}{N} \sum_{i=1}^{n_s} \varepsilon_i^2(M)$
  End of i
End of M
```

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 correspondence is obtained by considering all of the training shapes that have been used. This action will tend to overestimate the absolute error, but allow an unbiased comparison of different models.

Therefore, for comparison of two models A and B, if $G_A(M) \leq G_B(M)$ for all *M* or for a given *M*, we can conclude that the Generalization Ability of method A is better than that of *B*.

In order to evaluate the significance of differences when using different *M*, we estimate the likely error in $G(M)$. The standard error of $G(M)$, which is derived from the sampling distribution for a mean is given by [104]:

$$\sigma_{G(M)} = \frac{\sigma}{\sqrt{n_s - 1}} \qquad \textbf{(4.1)}$$

Where σ is the sample standard deviation of $G(M)$, and M is the number of modes/shape variations used in the evaluation.

4.1.2 Specificity

Specificity (S) measures a model's ability to generate instances of the object class that are similar to those in the training set. It will be spontaneous to evaluate this ability by generating a population of instances using the model and comparing them to the members of the training set. We define this measure (as a function of *M*, where *M* is the number of shape variations/mode used in the measurement) as:

$$S(M)=\frac{1}{N}\sum_{j=1}^{N}\left|x_j(M)-x'_j\right|^2 \qquad \textbf{(4.2)}$$

Where x_j are shape examples generated by the model (by choosing values of *b* in ASM in the range over the training set randomly) and x'_j is the nearest member in the training set to x_j. Therefore, if $S_A(M)\le S_B(M)$ for all *M* or a given *M*, *A* is more specific. The standard error of $S(M)$ is defined as:

$$\sigma_{S(M)}=\frac{\sigma}{\sqrt{N-1}} \qquad \textbf{(4.3)}$$

Where σ, is the sample standard deviation of $S(M)$, *N* is the number of samples (in our experiment $N=10000$), and M is the number of shape variations/modes used in this evaluation.

4.1.3 Compactness

Compactness (C) measures a model's ability that uses as few parameters as possible to cover the same variance. It is helpful to calculate this ability as a plot of cumulative variance:

$$C(M)=\sum_{m=1}^{M}\lambda^m \qquad \textbf{(4.4)}$$

Where λ^m is the m^{th} largest eigenvalue and *C(M)* is the cumulative variance of the M^{th} mode. If $C_A(M) \le C_B(M)$ is true for all shape modes or some of the shape modes, *A* is more compact.

As for Generalization Ability and Specificity, the likely error in *C(M)* is also given. The standard deviation of the sampling distribution of the variance of the m^{th} node is given as:

$$\sigma_{\lambda^m} = \sqrt{\frac{2}{n_s}}\lambda^m \qquad \textbf{(4.5)}$$

where λ^m is the m^{th} largest eigenvalue of the covariance matrix. The standard error of *C(M)* will be defined as:

$$\sigma_{C(M)} = \sum_{m=1}^{M}\sqrt{\frac{2}{n_s}}\lambda^m \qquad \textbf{(4.6)}$$

where M is the number of shape variations/modes used.

4.2 Manual Landmark Placing

The first Active Shape Model was built by Cootes *et al* [21]. In this approach, they built correspondence by manual landmark placing on hand shape outlines. Although manual annotation has been accepted as a ground truth and the Shape Model built by these points also often lead to a valid shape, there is no guarantee for good performance because it is subjective, and error-prone. Manual landmark placing is also a very time consuming process. In some 3D cases, it may take a month for a specialist to mark these datasets. Another disadvantage can be seen from Davies's paper [22]. In this paper, the so-called ground truth has been verified that it will not always show desired properties. According to the results of reference [22], Minimum Description Length (MDL) outperformed the “Ground Truth” in the metric of Generalization ability, Specificity and Compactness. In Figure 4.3, shows a set of manually placed landmarks on MR image.

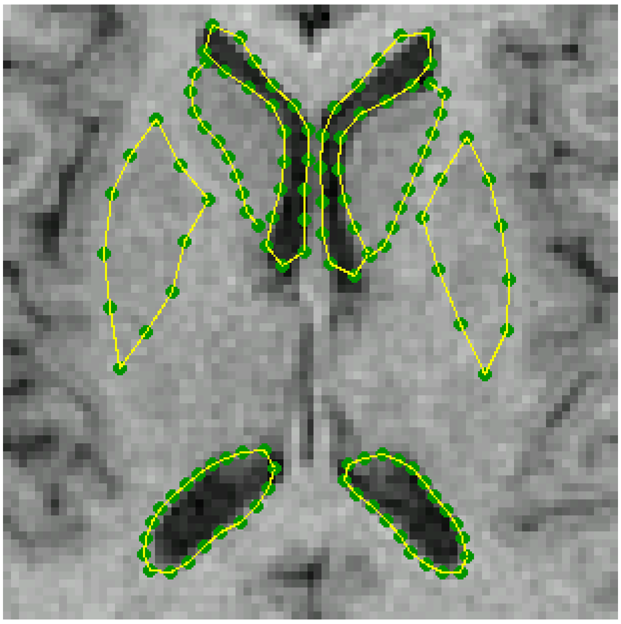


Figure 4.3 Here is an example of labelled brain MR image from Cootes’s (reference [16]) website.

4.3 Iterative Closest Point Algorithm

Iterative closest point (ICP) is a straightforward registration algorithm to find correspondences where shapes are close to each other when shapes are roughly aligned. Besl *et al* [24] describe this method as a way to register a pair of shapes and define correspondences between them. An initial correspondence is established by finding the closest points between shapes and an initial transformation is also defined. An iterative procedure is then adopted for finding the convergence to a local minimum. The cost function for this convergence is defined by the squared distance between shapes. Finally, the best solution is selected from results starting from different initial positions. In Figure 4.4, shows the results of two curve registered by ICP.

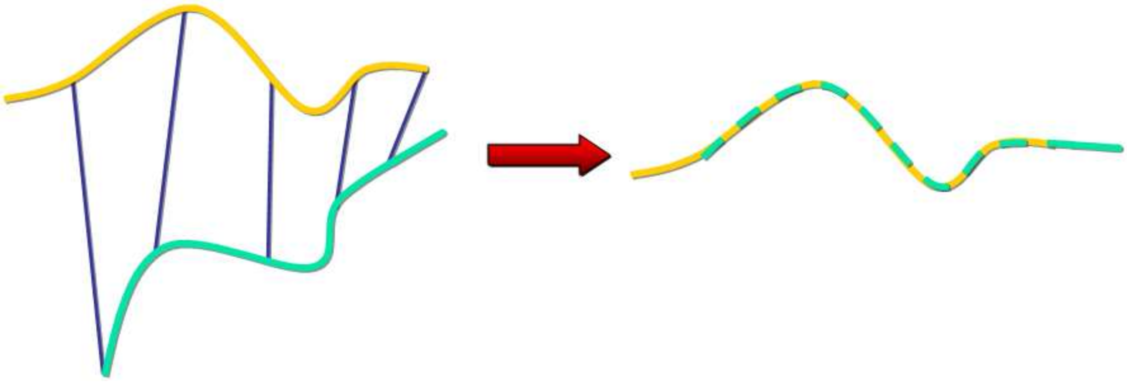


Figure 4.4 An example of using ICP to roughly register two shapes is shown.

The correspondence solution for this kind of method is straightforward and simple to implement. However, it is quite arbitrary to claim correspondence after shapes are roughly aligned. Therefore, it is unacceptable to say that correspondence is defined by the closest distance.

4.4 Shape Matching-based Correspondence

Stalib *et al* [25] applied flexible constraints on deformable shape, in the form of a probabilistic deformable model, to the problem of segmenting 2D shapes and finding correspondence. The parametric model is based on the elliptic Fourier decomposition of the boundary. Probability distributions on the parameters of the representation bias the model to a particular overall shape while allowing for deformations. Boundary finding is formulated as an optimization problem of maximizing a posterior objective function.

The problem for this algorithm is that when the training set is limited, the final segmented shape will be unlike any shape from training set (bad Generalization Ability).

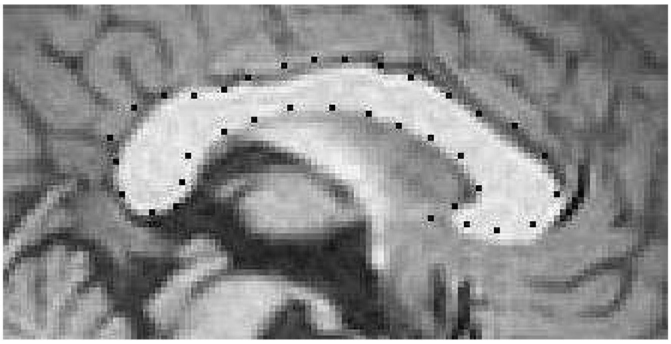


Figure 4.5 An example of result by applying Wang et al's algorithm to a set of hippocampus. Details please see [26]).

Wang *et al* [26] overcame this problem by adding on a purposely trained matrix onto the covariance matrix from training set. In this way, the shape variations become more global and rigid. However, this global variation is often objective and not subjective to the datasets.

In reference [27], Su *et al* used Independent Component Analysis (ICA) [63] to capture shape variations, they claimed that by using ICA, more local and accurate boundaries can be detected. A Markov Random Fields [86, 87, 88] based cost function was used to facilitate the relations between points, and final optimal segmentation results can be achieved in maximizing the posterior probability fashion [85]. In Figure 4.5, we show a demonstration of difference between ICA and PCA. From Figure 4.6, it can be seen that PCA assumes that distributions are normal to each other; ICA can capture real undergoing distributions which are not orthogonal to each other.

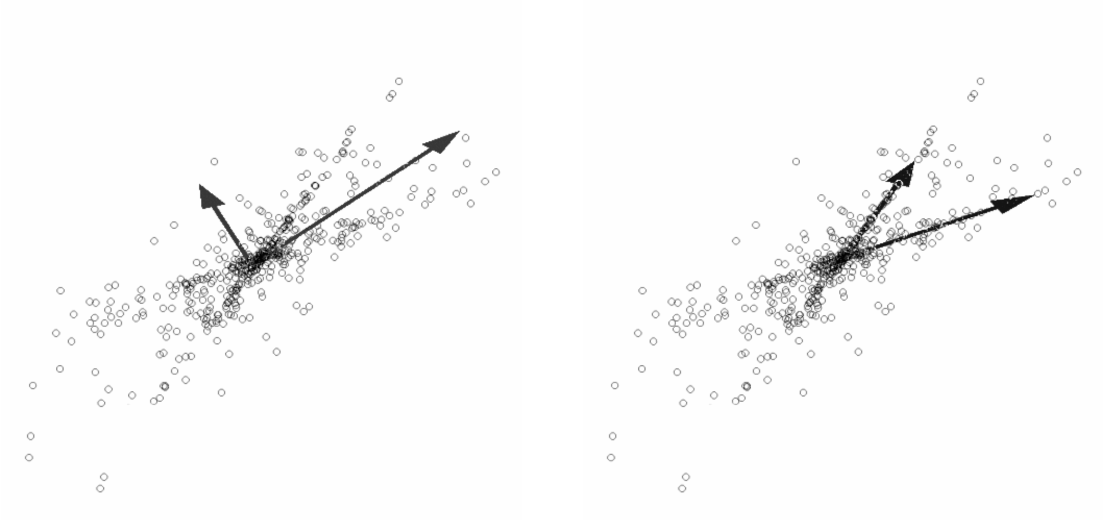


Figure 4.6 Here is an example of using PCA and ICA to extract the underlying distribution from a cluster of datasets. On the left, red arrows show the distribution found by PCA and on the right shows the distribution found by ICA.

Although compared with Wang's [26] results, Su's [27] results were improved in terms of accuracy, the essential problem of automatic landmark placing is not solved. Since, during experiments, they used training datasets marked by an expert and correspondence is achieved only when they use trained ASM to segment a new dataset.

4.5 Shape Properties-based Correspondence

Another intuitive approach to establish correspondence is to use similar local shape features. Curvature is the most often used criterion. For example, such an approach is established by Varun *et al* [28]. In the paper, a shape descriptor based on curvature distribution along a geodesic neighbourhood is used. Thresholds of the curvature are adopted to make the descriptor more robust against non-rigid shape deformation. Once the descriptor is computed for every point or feature vertex of two shapes to be matched, a one-to-one correspondence can be built. The advantage for this technique is that it can achieve correspondence relevantly quicker. Moreover, this algorithm is based on a very intuitive way of people's general idea about the world. Therefore, it can solve some simple shapes with distinguish features, like buildings with sharp corners, etc. However, for medical images, it is always not that simple. Since, medical images are involved with normal organs with smooth surface (e.g. liver) or abnormal organs with random appeared curvatures (e.g. tumour). Therefore, it is still not ideal for solving the finding correspondence problem by using curvatures.

Hill *et al* [29] built a framework for automatic landmark identification. It employs a binary tree of corresponding pairs of shapes to generate landmarks automatically on each set of example shapes. The correspondence algorithm locates a matching pair of sparse polygonal approximations, one for each of a pair of boundary by minimizing a cost function using a greedy algorithm. The greedy algorithm produces a set of points that lie on regions of high curvature. The cost function expresses the dissimilarity in both the shape and representation error (with respect to the defining boundary) of the sparse polygons. Therefore, minimizing the cost function will convey the correspondence. The method, however, allows invalid correspondences between the examples. Nevertheless, this pitfall has been overcome by flattening the surface before establishing correspondence, which is using an angel preserving technique to map a shape to a sphere. Although some results can be achieved automatically, the correspondence is still in a completely arbitrary manner, since different correspondence can be achieved according to the same theory. All in all, using the curvature to build up the initial correspondence set may also jeopardise the solidity of correspondence.

4.6 Finding Correspondence in a Learning Process

The state-of-the-art technique for finding correspondence, so far, is to treat correspondence as an optimization problem and find correspondences by optimizing an explicit objective function. This will actually allow good properties according to the criteria for building the cost function.

Different authors [24, 31, 36,] have proposed using the trace of the model covariance as the objective function.

$$Function_{Trace} = \sum_{m} \lambda^{m} \qquad \textbf{(4.7)}$$

where $\{\lambda^{m}\}$ are the eigenvalues of the covariance matrix. By minimizing this function, landmarks are moved towards the mean, directly minimizing the total variance of the model. Therefore, it is the same as the Compactness (C metric) measure. An example of using this technique can be seen from Figure 4.8. From Figure 4.7, we can see that the standard deviation has been decreased. As discussed in Davies [35], this model preferred a model with equal spaced landmarks and it is sensitive to initialization positions. The other problem is that although a compact model can be guaranteed, its Compactness is also overestimated such that the proposed model scored less on Generalization ability and Specificity than Minimum Description Length [35]. Although the trace-model can use less parameters, however the model's bad Generalization and

Specificity is not ideal.

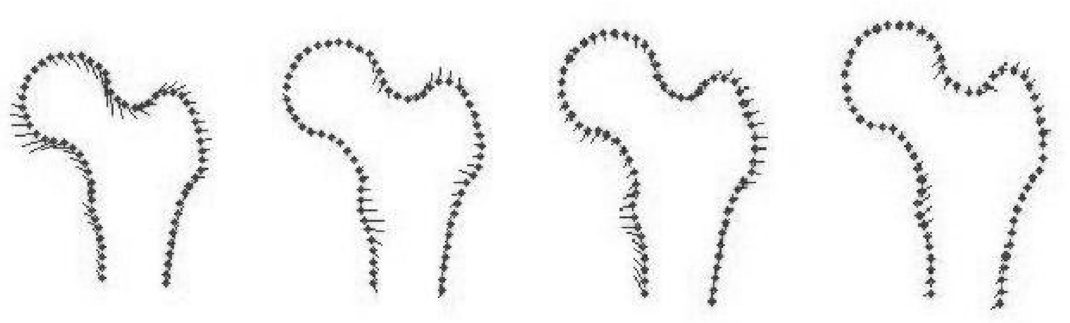


Figure 4.7 Shown is the mean shape with red marks, the whiskers starting from the marks indicate three standard deviation of the first two principal components. Left: The model is built from equal spaced initialization. Right: The model is built with traces of shape covariance matrix.

Hill *et al* [30] adopt a curvature matching dynamic programming algorithm to obtain an initial correspondence and then optimize an objective function for the final correspondence. The objective function is the trace of the model covariance matrix plus a correction term that penalizes points for moving off the original shape boundary.

$$Function_{Trace} = \sum_{i=1}^{n} a_{ii} + \varepsilon \qquad (\mathbf{4.8})$$

Where in Equation 4.8, if the shape covariance matrix is *A*, the cost-function is the sum of the elements on the main diagonal of *A* plus a correction term. Though some of the plausible results have been achieved, there is some potential drawback in this algorithm. Points can be moved off the boundary due to bad reconstructions. Several other authors also reported using trace of model as objective function to find correspondence points as in [24], [36], [62].

The determinant of the model covariance matrix as an objective function, is adopted by Kotcheff *et al* in reference [31]. The cost-function is shown in Equation 4.9.

$$Function_{Det} = \sum_{m} \log(\lambda^{m}) \qquad \textbf{(4.9)}$$

Where $\{\lambda^{m}\}$ are the *m*th eigenvalues of the covariance matrix. Similar to the previous cost function based on trace of the covariance matrix, this cost function effectively measures and minimizes the volume spanning in the variation space but still it favours a compact model. Moreover, it achieves a meaningless minimum when any eigenvalue approaches zero. To overcome this problem, Kotcheff *et al* add a small constant ε, and then the cost function becomes:

$$Function_{Det} = \sum_{m} \log(\lambda^{m} + \varepsilon) \qquad \textbf{(4.10)}$$

They argue that an appropriate value of ε can be estimated from the noise on the training shapes.

Kotcheff's model [31] will actually lead this model to a more compact one. The problem for Kotcheff's work is that it degenerates minima and thus requires an arbitrary parameter to keep it well defined. The arbitrary parameter will affect the convergence properties of the model. Correspondences are using a piecewise-linear function. Strong constraints have been used on the re-parameterization functions to make sure the final converged shape is valid. However, this shape re-parameterization can not be extended into 3D easily. Though a genetic algorithm (GA) has been used to optimize the objective function by manipulating the parameterization function, it is still relatively slow. Normally it takes a day to run the whole algorithm on 2D and can not cope with complex objects or large datasets. Although, the Minimum Description Length (MDL) also cast the finding correspondence problem in a learning process, since MDL is our main target for comparison, the background theory of this algorithm is in the next section.

4.7 Minimum Description Length Approach

Most recently, the Minimum Description Length (MDL) has become the state-of-the-art approach for solving the correspondence problem. The first paper using MDL as an objective function to solve correspondence problem is published by Davies *et al* [23]. This very first paper publication about MDL model soon attracted a lot of attention to this field [32], [49], although the algorithm is still quite hard to duplicate and implement. In addition, Davies quantitatively demonstrated that MDL has better performance than Hill's *et al* [30] and Kotcheff's model [31]. In reference [49], the author provided a simpler form of the MDL cost function and the gradient of the cost function, which makes the algorithm easier to implement. Although, some improvements have been made to the MDL approach, the model is not flawless yet. As have been reported by Davies in [23] and also Thodberg in [32], the cost function will be trapped in a meaningless local minimum, when parts of the landmarks will be trapped in a one place. We will introduce the MDL theory by first discussing the rationale behind it.

4.7.1 Correspondence and Statistics

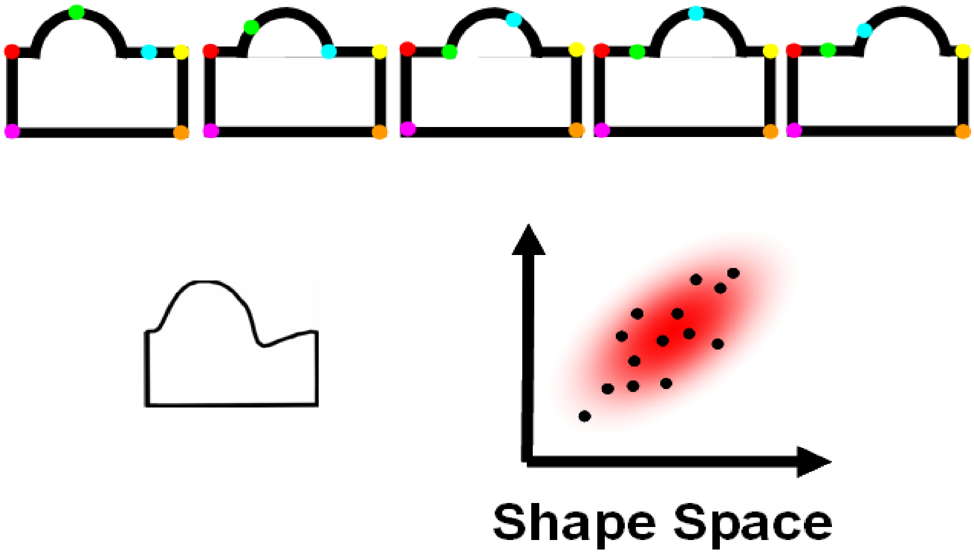


Figure 4.8 This is a demonstration of wrong correspondence. On the top row it shows the dataset with manual marks. Correspondence is achieved by placing nodes with the same colour. Bottom row left: A new shape generated by the Shape Model, which was constructed from the datasets. It can be seen that due to the wrong correspondence, the new generated shape does not look like any example in the dataset. Bottom row right: The statistics in feature space is formed as an elliptical shape.

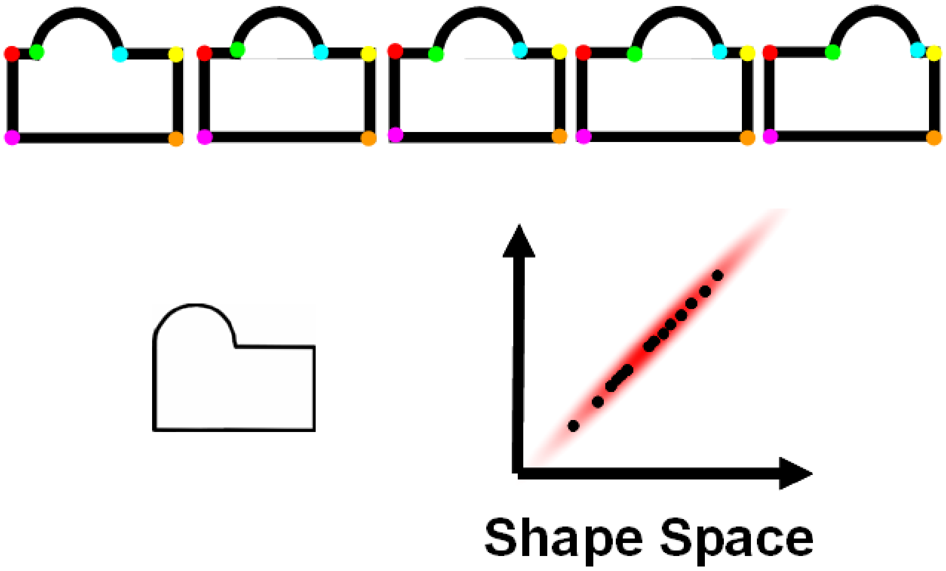


Figure 4.9 This is a demonstration of correct correspondence. On the top row, it shows the dataset with manual marks. Again, correspondence is identified by the same colour. Bottom row left: A new shape generated by the Shape Model, which was constructed from the datasets. It can be seen that with right correspondence, the new generated shape share the same genus with dataset. Bottom row right: The statistics in feature space is formed as a line.

Statistics is often used to describe the properties of group behaviour. Therefore, it is straightforward to think about finding correspondence as manipulating its statistics. In Figures 4.8 and 4.9, a demonstration of a plausible connection between right correspondence and statistics is shown. It can be seen from the above figures that, with correct correspondence points marked on the training set, the shape space will become more “compact” and orderly. Therefore, varying correspondence will be “equal” to varying statistics.

Davies *et.al.* use an information based function to describe the information amount used to represent this shape model parameters and data. The “best” model, which defines the correspondence, is characterized as the one that minimizes the description length of the training set, arguing that this leads to models with good properties. They argue that the simplest description of the training set will interpolate and/or extrapolate best. The notion of the ‘simplest description’ is formalized by Minimum Description Length (MDL), which is from the Shannon Coding codeword length [39]:

$$Descriptionlength = -\log_2 p \text{ bits or } -\ln p \text{ nats} \qquad \textbf{(4.11)}$$

The basic idea is to minimize the length of a message required to transmit a full description of the training set, using the model to encode the data. The whole cost function is based on the measurement of two parts: the first part is the information needed to describe the encoded model, which includes the mean shape and the shape modes; the second is the information for the training shapes, which is the p_i from the model Probability Density Function (PDF). Since, the information to describe the mean shape $\overline{X}$ and the eigenvector can be assumed to be constant for a given training set, thus only information needed to describe training shapes will be calculated.

The term of "Description Length (DL)" is defined as follows: for example, a set of possible events {i} with probabilities { p_i }, thus, the codeword description length of event {i} is equal to $-\log p_i$. The whole training set of shapes will be encoded in this way which includes Encoded Model (mean shape, model modes etc) and each training shape (p_i from model probability distribution function). It is quite reasonable to assume that information amount for describing the mean shape and modes is assumed to be constant for a given training set, thus only the second term will be calculated. Therefore, the total cost function is simply the sum of the descriptions for all weighting variables as in Equation 4.12.

$$Total\text{DL} = \sum DL \qquad \textbf{(4.12)}$$

For example, when all the shapes are roughly marked. An Active Shape Model can be built according to the coordinates of these landmarks. Each shape can then be constructed by the mean shape, shape variations and weighting vector (recall Equation 2.8, $X = \overline{X} + Pb$). As have been discussed in previous chapters, each component in the weighting vector can be assumed to comply with a Gaussian distribution. Therefore, the Description Length for each component of each weighting vector can be easily calculated by the definition of Description Length and the total Description Length is sum of Description Length of components in each weighting vector.

Thodberg [38] and Ericsson [49] improve this MDL technique and derive a new form of the final description length such as:

$$\text{Description Length} = \sum L_m$$

$$L_m = 1 + \log(\lambda_m / \lambda_{cut}) \quad \text{for} \quad \lambda_m \geq \lambda_{cut} \qquad \textbf{(4.13)}$$

$$L_m = \lambda_m / \lambda_{cut} \qquad \text{for} \quad \lambda_m < \lambda_{cut}$$

This cost function has the property that it tends to zero when all eigenvalues tend to zero and both L_m and $dL_m / d\lambda_m$ are first order derivative continuous at the cut-off λ_{cut}. In other words, when λ_m falls below λ_{cut}, the benefit of decreasing it further is no longer logarithmic, but levels off and reaches a minimum one unit below the transition point. A mode with eigenvalue λ_{cut} contributes on average a variance of λ_{cut} / N per mark, and since the *rms* radius of the aligned shapes is $1/\sqrt{N}$, the mode contributes a standard deviation per *rms* radius of $\sigma_{cut} = \sqrt{\lambda_{cut}}$. So λ_{cut} can be evaluated by σ_{cut} that is defined by the noise level of the training set.

In summary, MDL cost function is based on the assumption that all shape variations are independent and equally weighted for each other. Each shape variation is normal to each other but they are not independent to each other. For example, in the *N* dimensional space, given (*N*-1) shape variations we can use the rule that the sum of all squared shape variations is equal to one to calculate the remaining shape variation. What is more, in reality, the assumption “might” work for some applications, however when the training set is loose or the covariance matrix is ill defined, the assumption will not hold right. For example, the shape variations with smaller eigenvalue will most likely to be blurred with noise. This may be one of the main reasons why researchers use the first few eigenvector to construct the shape model. We therefore suggest using automatic calculated statistical weights for each energy component, which will be discussed later.

4.7.2 “Pile Up” Problem

The results shown by Davies [35] were encouraging, and soon attracted a lot of attention. However, it has drawbacks. The problem is the reported “Pile Up” problem [22], [32]. The problem is a situation when all or parts of the landmarks in the Active Shape Model piled up in some locations and fail in conveying a reasonable correspondence shape model. This is due to the fact that the optimal MDL result is a local minimum, which has to be near to the initial position. It is quite straightforward to realize that when all landmarks pile up into one point, the cost function of MDL will achieve global minimum. This problem happens quite often when the landforms of shapes are complicated [32].

In Davies’s thesis [22], he suggested a possible solution in avoiding this “Pile Up” problem. He suggested using the single master example method. In the optimization method, one of the shapes in the training set is set to be fixed. For this particular example, the landmarks have been marked and used as a reference shape. Therefore, this shape will influence other shapes. What was hoped is that the solution will somehow relate to the result with the accuracy of the first master shape. However, applying this manual reference shape is against the sprit of automatic shape modelling. Other researchers [32] have reported that this method of a single fixed master example is not sufficient to keep the whole set in place. For example, the free endpoints of open curves can drift systematically to one side or the other, neglecting the master example. Another example is in closed curve, if we have a large dataset, say 100, the statistical weight of the majority can outweigh the single master example and gain of the run-away exceeds the cost of a single outlier easily.

Another solution to this problem is a programming technique that we can reinitialize parameters until the right optimal result is found. For example, if one initialization parameterization meets the “Pile Up”, another initialization can be used until no problem is met. However, we need observer interference to select the correct correspondence result, which otherwise can lead the algorithm into an arbitrary and subjective manner, and the results will be prone to error as well. Therefore, several credible results will be achieved according to different initialization and it is hard to justify which one holds the right correspondence.

Hans [32] has reported that using a curvature based external function or a node penalty can avoid this “Pile Up” problem. Therefore, the cost function is now composed of using both MDL and an external term. The theory is that this external cost function will favour some areas (for example, areas with high/low local curvature) so that it will trap landmarks and limit the landmark’s moving ability into a small area. However, the “Pile Up” problem might be solved in this way. Adding an external cost function changes the correspondence problem back to an arbitrary manner. Furthermore, in complex cases, where high curvature is present (for example, face profiles), the external term may be overweighed. Whereas is an organ like the liver there is lack of change on the surface. The MDL term can overwhelm the curvature term easily. Therefore, in practice weighting those two terms in different circumstances will be a very hard problem.

The “Pile Up” problem can be imagined clearly as the points that run away from the correct positions and pile up in some locations; therefore, the shape model fails in describing the rest of the shapes and in reaching a smaller description length than the optimal correct one. The reason for Minimum Description Length’s misbehaviour is quite complex. However, we can attribute this to the drawback of Description Length. For example, if all the points on the shapes fall into one point, it will attain minimum, which is actually a global minimum. Here, we perform an experiment regarding to this “Pile Up” problem.

Twenty-four artificial datasets as the ones used in Figures 4.8 and 4.9 were generated. The shapes are rectangles with a bump moving from the left to right. Since the ground truth of datasets is known, we can make a comparison between the ground truth and the MDL results. From Figure 4.10, it can be observed that the MDL results are different from the ground truth. We therefore calculate the MDL output for both ground truth position and the MDL converged position. In the ground truth position, the MDL output is 20.135 and for the converged position the MDL cost function output is 18.591.

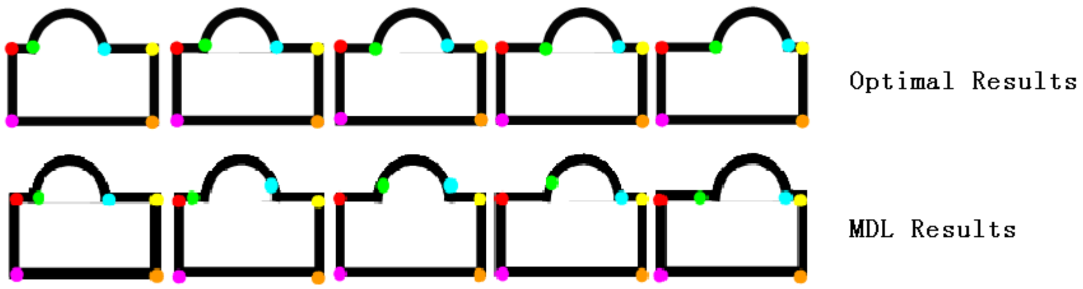


Figure 4.10 A comparison between optimal results and results from MDL. The top row shows the optimal results; the bottom row, shows the MDL results.

Another experiment is performed on the datasets of human face profiles. We compare the MDL converged results and some of the manual landmarked results as in Figure 4.11.

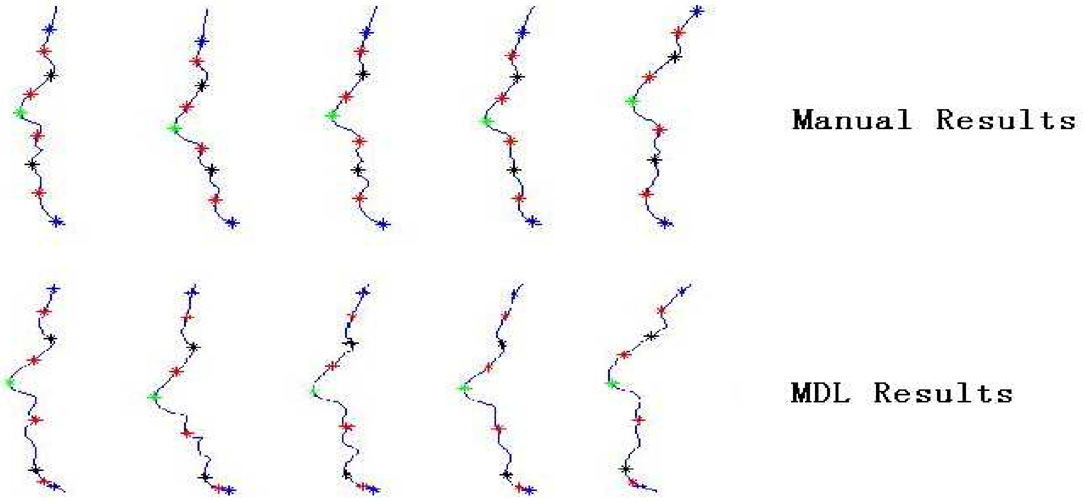


Figure 4.11 A comparison between manual results and MDL results. On the top it shows the manual landmarks results; On the bottom, it shows the MDL results.

It can be seen that some of the points in the chin area piled up. The MDL cost function output for both cases: manual results is 29.43 and the MDL result is 27.6643.

4.8 Conclusions

Based on the knowledge offered by reviewing the published literature on finding correspondence across shapes, we can see that some promising results have been achieved, although they suffer from a number of problems. For example, the manual landmark placing is too error prone and time consuming, especially in 3D cases; ICP registration provides a quick way to find correspondence where shapes are close to each other. However, close in distance does not necessarily mean that they are corresponding to each other. By giving different initialization positions, several converged results can be found and there is no evidence shown how to pick up the correct correspondence from these convergence results. For shape matching based algorithm, the essential problem of how to automatically find correspondence points across datasets is not solved. Therefore, it is still a segmentation technique. Correspondence points will be achieved in the same time, when a new shape is segmented by using an Active Shape Model constructed from manually marked datasets. For shape properties based techniques, it can be seen that for a soft organ, such as liver, there are no obvious shape properties to model. Davies *et al* [35] showed the most appealing and intuitive solution to tackle the correspondence problem so far. They find the correspondence across the datasets in a learning process. The objective cost function is based on information theory which mainly measures the utilities (Description Length) used for one model. They argued that the model, which has the smallest Description Length, would hold the correspondence. However, problems have been reported by Thodberg [38] and Davies [35], that due to a small pitfall of the cost function, shapes will "Pile Up" from the right correspondence and pile up in some areas. Davies [35] suggested using one marked example to influence other examples in the training set. In theory, given a large dataset, one single marked example is not enough to influence the rest of the examples. Thodberg [38] suggested using external cost function as a solution to original MDL cost function to constrain the moving ability of nodes during optimization. The cost function can be extracted from shape properties such as curvature. Then again, this external cost function becomes hard to define according to different landforms of datasets. Therefore, it allows different correspondence to be achieved given different weights to the cost function.

As a result, a new method that can have the good correspondence properties and do not suffer the problem of “Pile Up” without using external energy function is required. In term of good correspondence properties, we are referring to the performance of Active Shape Model constructed from the automated identified correspondence landmarks. This performance evaluation can be achieved by common accepted measurements, for example the Generalization Ability, Specificity and Compactness. These three criteria have been used by many other researchers in [95] [96] [97] [98]. Due to lack of ground truth, researchers often use manual marked results to evaluate the performance of different methods. However, these manual results are quite subjective, and dependent on user’s experience. Different users may conclude different manual results. Rather than trying to retrieve the ground truth, the three comparison methods evaluate the performance of the model from a different perspective, which evaluates the properties of the shape model built from correspondence points found by different algorithms. In validations, all these three estimates are measuring the error quantity of the corresponding ability. Something we should keep in mind is that, for example in Generalization Ability, the smaller Generalization Ability value is, the more general the model is. Within those three criteria, Generalization Ability and Specificity are more important, since in most of the cases, researchers care more about the performance of a model rather than how many parameters were used. For example, if model *A* with 12 parameters and model *B* with 10 parameters can represent same variations, which means $C(A) > C(B)$, but if $G(A) < G(B), S(A) < S(B)$, in general case, we will still choose model *A*. Another possible approach to evaluate different algorithms is to cast the constructed shaped model in a real medical image application such as image segmentation or shape classification/recognition.

The following chapters will describe our new proposed method, which is also based on information theory and group-wise optimization that can hopefully deal with the limitations of both Davies's and Thodberg's paper to solve the "Pile Up" problem.

Chapter 5 A 2D Minimum Entropy Approach

This chapter will provide a brief description of our proposed approach, which builds correspondences in a learning process and in a group-wise manner. We also note that a pair-wise method exists in literature [60]. In this chapter, we will focus on 2D cases, and discussion on 3D will be in later chapters.

Our goal is to develop a method for solving the problem of finding correspondence automatically and solving problems left by the original MDL approach, for example the “Pile Up” problem. We also treat the correspondence problem, as a part of shape leaning process, by doing this the desired properties will be achieved, in terms of Generalization Ability, Specificity and Compactness. This approach will involve several steps in the framework, such as surface/shape parameterization, manipulating correspondence, efficient optimization, an objective function to optimize and criteria to evaluate the performance of the objective function. Among these steps, the objective function is essential. We seek an objective function that has the following properties:

(1) Achieve comparable or better score in the three evaluation criteria compared with “the state-of-the-art” method;

(2) Guarantee that the optimal result offers valid correspondences as an example: Solve the “Pile Up problem”;

(3) Applicable to both 2D and 3 D;

(4) Efficient to optimize: achieving convergence status in a relevant shorter time

compared with “the state-of-the-art” method

This chapter is organized as follows:

Section 5.1 will focus on the technique used for shape parameterization and correspondence manipulation in 2D. Details of correspondence manipulation will be further presented in sections 5.1.1 and 5.1.2, respectively.

Section 5.2 will discuss our method of composing a cost function and some of the relevant work.

Section 5.3 will give the optimization strategy in 2D cases.

Section 5.4 a brief conclusion will be drawn.

5.1 2D Shape Parameterization and Correspondence Manipulation

Many approaches have been developed for surface or boundary parameterization in the literature. The proposed approach should guarantee that only valid correspondences would be achieved during optimization. Intuitively, one plausible way is that we can put a number of points along the boundary and move them respectively. It is, however, very hard to converge to valid correspondence locations and also inefficient. A specified order of points must exist to make sure of the points correspondence. This will become an even more difficult problem in 3D cases. In 2D cases, we use the normalised shape length to parameterize landmarks; therefore, from start-landmark to end-landmark, the parameter runs from zero to one. A hierarchical way is adopted here for shape parameterization. This approach acts similarly to a multi-resolution approach. The figure below shows the method which is used to place nodes along a 2D shape. Figure 5.1 gives the demonstration of placing landmarks along a 2D shape.

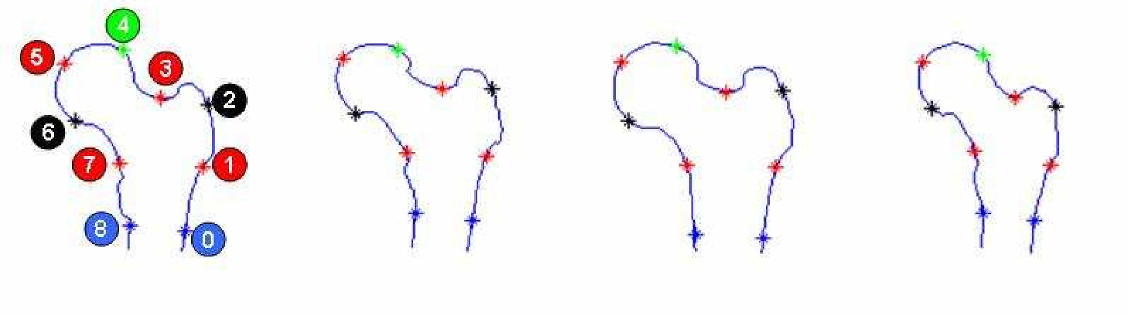


Figure 5.1 The node placement method is shown here. Nodes are numbered from 0 to 8 with four levels. First level is blue, which include node 0 and 8. Second level is in between blue nodes (parents nodes), which is node 4. Third level is black, which are node 2 and 6. Fourth level is red, which are nodes 1, 3, 5 and 7.

The advantage of this hierarchical node placement is that optimization can be performed to a specified level, which is defined by the accuracy requirement of the particular application.

For correspondence manipulation, one approach proposed by Kotcheff and Taylor [31] is very promising. They use re-parameterization to manipulate correspondence along curves or surface. Each node is defined by a monotonously increasing parameter (*u*), in the case of *N* landmarks *u* has values $\{ \frac{1}{N}, \frac{1}{N} * 2, \ldots, \frac{1}{N} * N \}$. A different re-parameterization function $\Phi_i(u)$ is defined for each shape $S_i(u)$, with a diagonal line as the re-parameterization function for the equal-spaced case. Both *u* and $\Phi_i(u)$ will be in the range of zero to one and of monotonic increase. For a valid correspondence, this objective function has to be diffeomorphic, which means folds or tears of shapes are not allowed. An example of this shape re-parameterization is shown in Figure 5.2 on corpus callosum data. By defining different curve scenarios, we can have different marks allocated along the boundary. In closed curve case, the last point is identical to the first point on the boundary. The $\Phi_i(u)$ value for the last point should be assigned to be equal to $\Phi_i(0)$. On the contrary, in open curve case the first point is different from the last point on the boundary. Therefore, the optimization problem can be sorted out by looking for the correct mapping curve to represent the correct correspondence.

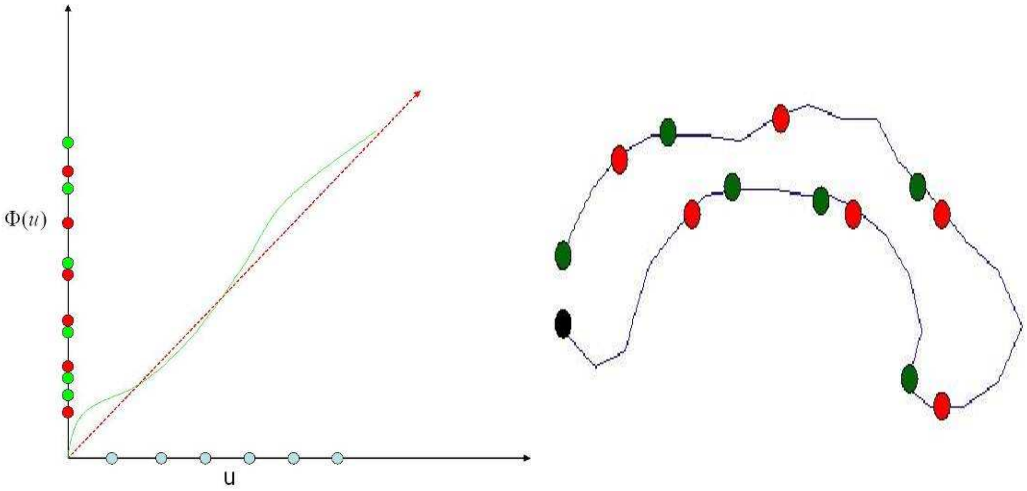


Figure 5.2 On the left, it shows the parameterization and re-parameterization function. The dilute blue is equally spaced on *U* coordinate representing function parameter. The *y* direction represents the normalized corresponding curve length. On the right is an outline of corpus callosum segmented from brain data. The curve length is calculated from the black dote. It can be seen that the red dotes are extracted from the dashed line and the green dotes are from the dilute blue curve.

Floater and Hormann [34] have investigated several different representations of shapes. Here we will use a recursive, piecewise-linear representation that is related to and extends the work of Kotcheff and Taylor [31], which is detailed in the next section.

5.1.1 A Piecewise-Linear Representation of Re-Parameterization

Kotcheff and Taylor used a piecewise-linear representation for re-parameterization function Φ. In 2D case, a set of nodes { p_i } are defined and placed along the function Φ curve, linear interpolation is used to evaluate values between nodes. An example can be seen from Figure 5.3. A linear function defined by nodes { p_i } will be used to approximately estimate function Φ.

$$\Phi(u) = \Phi(u_i) + (u - u_i)\frac{\Phi(u_{i+1}) - \Phi(u_i)}{u_{i+1} - u_i}, \quad u_i \le u \le u_{i+1} \qquad (5.1)$$

Where u_i and $\Phi(u_i)$ are function parameters standing for the path-length parameterization and re-parameterization of node p_i respectively.

The diffeomorphic properties of the function means that both u_i and $\Phi(u_i)$ must be monotonically increasing:

$$\begin{gathered} 0 \le \Phi(u_1) \le \ldots \Phi(u_i) \le \Phi(u_{i+1}) \ldots \le \Phi(u_n) \le 1, \\ 0 \le u_1 \le \ldots \le u_i \le u_{i+1} \ldots \le u_n \le 1 \end{gathered} \qquad (5.2)$$

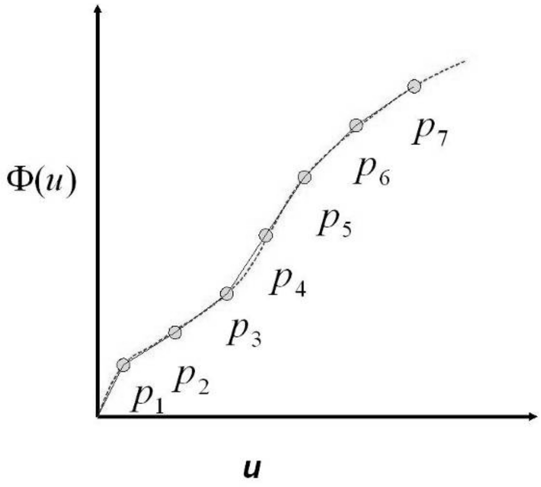


Figure 5.3 Seven points are used to describe the re-parameterization function. Each node can be moved along the curve and a linear function will be used to approximate the curve.

5.1.2 A Recursive Definition of Re-Parameterization

A more efficient and recursive algorithm is extended by Davies [35]. We are seeking a set of 2^L+1 marks on each curve to represent the shape. For closed shapes, the start and end points (number 0 and 2^L) are identical and there are only 2^L points on each shape. The hierarchical method to define points will be described as follows: the position of new node (say, level *i*+1 and *i* is an integer between 1 and L-1) is coded as its fractional distance κ_{ij} between its two parent nodes (level *i*) p_i and p_j. So κ_{ij} lies in the range of [0, 1] where it achieves 0 if it is placed on its left parent and 1 if it is placed on the right parent. For a closed curve with 65 marks, we specify on the first level the coordinates of mark 0 and 32 by their absolute arc length position. On the second level, mark 16 and 48 are specified by parameters between 0 and 1. For example mark 16 can be anywhere on the curve between mark 0 and 32, corresponding to the extremes 0 and 1. On the third level the marks 8, 24, 40 and 56 are specified in between already fixed marks. This is continued until level six so that all marks are placed. A clear example can be seen in Figure 5.4.

In Figure 5.4b, K_{00} is the fractional distance between origin and end. Figure 5.4c (i.e. level two): two other points are inserted, $\mathrm{K}_{02} \neq \mathrm{K}_{20}$. It shows an example of how to place 5 points of 3 levels on an open curve.

As the representation of re-parameterization is performed in a hierarchical manner, it allows optimization performed up to a specific level and makes the points of remaining levels inserted in the middle of parents points. The optimization details will be revealed in section 5.3.

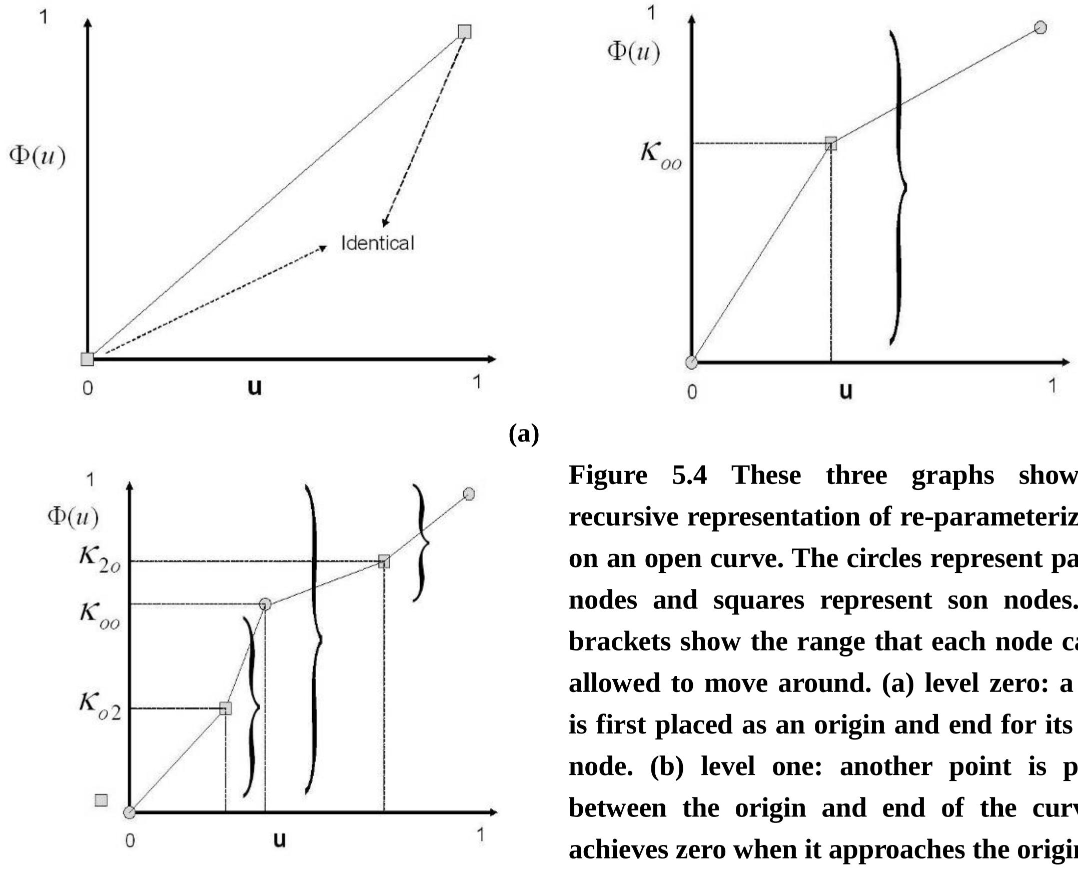


Figure 5.4 These three graphs show the recursive representation of re-parameterization on an open curve. The circles represent parents nodes and squares represent son nodes. The brackets show the range that each node can be allowed to move around. (a) level zero: a node is first placed as an origin and end for its child node. (b) level one: another point is placed between the origin and end of the curve; it achieves zero when it approaches the origin and one when it approaches the end.

5.2 An Entropy Based Objective Function

The essential property of an objective function is that it has a guaranteed minimum or maximum. By finding this optimal result, correspondence can be achieved at the same time. This section will give a glimpse of the previous work on the subject of finding correspondence in a learning process. Then we will represent our proposed entropy based objective function, which is from another branch of information theory.

5.2.1 Previous Work

5.2.1.1 A Minimum Description Length Approach

Davies *et al* [35] developed an information theory based approach for building the cost function. Moreover, they have shown that their Minimum Description Length model has better performance than the previous two models. They argued that the simplest description of the training set will interpolate and/or extrapolate best. The notion of the 'simplest description' is formalized by Minimum Description Length (MDL), which is by the Shannon Coding codeword length [39]:

$$Descriptionlength = -\log_2 p \text{ bits or } -\ln p \text{ nats} \qquad (\mathbf{5.3})$$

The basic idea is to minimize the length of a message required to transmit a full description of the training set, using the model to encode the data. The whole cost function is based on the measurement of two parts: the first part is the information needed to describe the encoded model, which includes the mean shape and the shape modes; the second is the information for the training shapes, which is the p_i from the model Probability Density Function (PDF). Since the information to describe the mean shape $\bar{X}$ and the eigenvector can be assumed constant for a given training set, only information needed to describe training shapes will be calculated.

After coding the parameters and data, a very complicated form is achieved, which is very hard to manipulate and understand. Therefore, Thodberg [38] and Ericsson [49] improve this technique and derive a new form of the final description length like:

$$\text{Description Length} = \sum L_m$$

$$L_m = 1 + \log(\lambda_m / \lambda_{cut}) \quad \text{for} \quad \lambda_m \geq \lambda_{cut} \qquad \textbf{(5.4)}$$

$$L_m = \lambda_m / \lambda_{cut} \qquad \text{for} \quad \lambda_m < \lambda_{cut}$$

This cost function has the property that it tends to zero when all eigenvalues tend to zero and both L_m and $dL_m / d\lambda_m$ are first order derivative continuous at the cut-off λ_{cut}. In other words, when λ_m falls below λ_{cut}, the benefit of decreasing it further is no longer logarithm, but levels off and reaches a minimum one unit below the transition point. A mode with eigenvalue λ_{cut} contributes on average a variance of λ_{cut} / N per mark, and since the *rms* radius of the aligned shapes is $1/\sqrt{N}$, the mode contributes a standard deviation per *rms* radius of $\sigma_{cut} = \sqrt{\lambda_{cut}}$. So λ_{cut} can be evaluated by σ_{cut} that is defined by the noise level of the training set.

In summary, MDL has become a benchmark for automatic shape model building. It achieves better values of the three evaluation properties than the previous models; however it still has some problems, which have not been totally clarified, one is the so-called “Pile Up” problem.

5.2.1.2 Solution for the Pile Up Problem

Some methods have been proposed to stop the “Pile Up” effect. One way to avoid this undesirable effect is by selecting a single shape as a master example (introduced by the MDL author Davies in [23]). The master example is the one that all points on the shape have been manually placed by an expert and these points are not allowed to move during optimization. This method is trying to make the master influence the behaviour of the other datasets. In this way, the MDL author is hoping that the points will not pile up in some locations. However, in some cases, a single fixed master is not sufficient to keep the whole set in place. For example, the MDL algorithm will be run on 100 datasets. One is selected as the master example; the other 99 datasets will be optimized by MDL. During optimization, the statistical weight of 99 datasets will overweight the only one master example easily. In this case, the “Pile Up” problem could happen. In another example, the free endpoints of the open curves, points can drift systematically to one side or the other easily; Thodberg reported this in [38].

Another remedy for the “Pile Up” problem is to add a stabilizing term to the MDL cost, which was introduced by Thodberg [38]. Instead of fixing the node parameters of the master example, he used a node cost instead:

$$NodeCost = \sum (a_i^{average} - a_i^{t\arg et})^2 / T^2 \qquad (\mathbf{5.5})$$

Where $a_i^{t\arg et}$ and $a_i^{average}$ representing parameters of target and average shapes. As defined previously, the parameter *a* is the parameter which defines the node’s absolute position between existing nodes. ${}^{1}\!/\!{}_{T^2}$ is a weighting component for the external energy component. Therefore, if the average drifts e.g. T =0.05 sway from the target, one unit of cost function value is added to the cost.

Rich shape information such as curvature plays an important role in image processing, therefore it is straightforward to use it as a complementary component to the existing MDL cost function:

$$t_i = r_{i+1} - r_{i-1},\ c_i = \pi N(r_{i+1} - r_{i-1} - 2r_i)\hat{t}_i / t_i^2 \qquad \textbf{(5.6)}$$

Where r_i is the 2D-vector of points *i*, t_i is the tangent of the shape contour, and $\hat{t}_i$ is the normal of the shape contour. From the expression of this equation, it can be seen that curvature expression is independent of the pose of the shape and it is one for a circle. The curvatures c_i are then convolved with a Gaussian filter. The smoothed curvature value at mark *i* of the *rth* example is denoted as Cu_{ir}. For open curves, the curvature can not be computed at the ends, and close to the ends it also becomes quite noisy due to the smearing. Therefore, curvature near the ends is not calculated.

The following extra term added to the MDL cost function is constructed to measure the compactness of the curvature description of the set:

$$CurvatureCost = \vartheta \frac{1}{N} \frac{1}{s} \sum_{i,r} (Cu_{ir} - Cu_i^{mean})^2$$

$$\textbf{(5.7)}$$

$$Cu_i^{mean} = \frac{1}{s} \sum_r Cu_{ir}$$

Here, *s* is the number of the shapes, and ϑ is the weighing factor for this term. The curvature cost is independent of the resolution, as the other terms in the cost function.

The proposed method of adding an external term such as node penalty or curvature cost can solve the “run-away” problem by force, however this external cost-function can lead the method to an arbitrary solution. For example, different weights on external energy function can result in different correspondences and different local extremes. Nothing has been revealed on how to evaluate these weighting parameters. The ideal approach is fixing the ill posed cost function problem inherently, rather than using outer force to constrain its behaviour. In the next section, we will discuss the proposed approach, which also used an information theory to model the shape but has shown better performance than MDL approach.

5.2.2 An Entropy Based Objective Function

We seek a principled basis for choosing an objective function to describe the training shapes that will directly favour models with good correspondence performance and strong ability to fight the “run-away” problem. In order to achieve these good properties, we try to consider Entropy [39] as a basis to form an objective function.

It can be useful to think of finding correspondence as trying to maximize the amount of shared information in all images in datasets, in a group-wise manner. In a qualitative sense, we may say that if shapes with correct correspondence are correctly aligned, then the mutual information between the shapes will be maximized. Therefore, less information will be needed to describe the shape model. On the other hand, if the correspondence is poor, shapes will be out of alignment, in which case, we will have duplicated and redundant versions of information from shapes. Therefore, more information will be needed for describing the shape model. Bearing this in mind, finding correspondence can be thought of as reducing the amount of information in the combined images, which suggests the use of a measurement of information as a criterion. The most commonly used measurement of information in signal and image processing is the Shannon-Wiener entropy measure *H* [39]

$$H = \sum_{i=1}^{n} -p_i \log(p_i) \qquad \textbf{(5.8)}$$

H is the average information supplied by a set of *n* symbols whose probabilities are given by $p_1, p_2, p_3, \ldots p_n$. One of the desirable properties of Entropy is that it will have a maximum value if all symbols have equal probability of occurring, which is the case when a stack of points pile up into one location and “Pile Up” happens and MDL achieves a meaningless minimum. Although the difference between the equations of Entropy and Description looks trivial, this observation can solve the so-called “Pile Up” problem inherently. The Entropy based Energy cost function has the ability to fight the “Pile Up” problem.

In finding correspondence, we have several shapes $A_1, A_2, \ldots, A_n$ to align. We therefore have probabilities of weighting components from this training set. Joint entropy measures the amount of information we have in the several combined images [39]. The concept of joint entropy can be understood using the assumption that the probability distribution for every weighting component in the Active Shape Model (ASM) [40] is a zero centred Gaussian distribution. So for i_{th} weight b_i on j_{th} component

$$p_i = \frac{1}{\sqrt{2\pi\lambda_j}} e^{-b_i{}^2/2\lambda_j} \qquad \textbf{(5.9)}$$

Where λ_i is the j_{th} eigenvalue.

In general, there are two main differences between MDL and the proposed Minimum Entropy Model (MEM) in the way of composing the cost function. Firstly, Entropy is used as an alternative of Description Length for information quantity measurement, since, Entropy has important physical implications as the amount of “disorder” of a system. We are arguing that, the system is ordered when the points are corresponding to each other. Secondly, as we can see both MDL and MEM approaches are using PCA to extract shape variations and the probabilities we use are based on these variations. In PCA analysis, we normally use the eigenvalue to denote the degree of shape variations. Therefore, it is quite straightforward to think that shape variations should be treated differently in measurement, and shape variations with larger eigenvalue should be more appreciated. In this sense, we propose a method of composing our cost function as a combination of entropy with different assigned weights. We choose λ_i as weight for each H_i, since λ_i is the natural expression of statistical weight of each mode.

$$Costfunction = \sum_{j=1}^{t} \lambda_j H_j \qquad \textbf{(5.10)}$$

Where λ_j is the j_{th} eigenvalue, H_j is the j_{th} Entropy derived from $\{b_1^j, b_2^j, \ldots, b_i^j, \ldots, b_m^j\}$. The parameter b was defined in Equation 2.8, which are weighting parameters for Active Shape Model, t is the number of eigenvectors used, as well as the number of weighting components used in the shape model.

5.3 Optimization Strategy

Instead of the Genetic Algorithm (GA) used in Davies’s paper [35], Thodberg [38] proposed a local iterative optimization method, which has been used in many applications [41, 42, 43]. Compared with Davies's approach, this local search algorithm is easier to implement and understand. Since after personal communication, Thodberg [38] provided his code and datasets for us to evaluate our proposed algorithm. We are going to discuss more details about Thodberg's optimization method below.

For example, an Active Shape Model uses 32 landmarks on each shape, but optimization is performed up to the third level (8 landmarks). The other landmarks are set equally spaced between the existing landmarks. All the landmarks are first placed along the boundary according to ascending level. An initial step-length, which is decided by the accuracy the experiment needs, is associated with each node. The step length controls the changes of landmark position parameter K, which was defined earlier in paragraph 5.1.2. Please recall that the parameter K runs from zero to one, which controls the landmark run from left to right of the parent landmarks.

In our experiment, Thodberg's approach uses 0.01 as initial step length for MDL algorithm and it will decrease automatically by the algorithm. Below, a pseudo code is provided about the MDL optimization procedure. In this optimization scheme, the whole training set is roughly Procrustes aligned, then PCA is applied to the aligned shapes, after computing the MDL cost, each node is probed 6 times to find a cost-function value in the downhill direction. The algorithm runs sequentially for each control node, and will stop after 40 passes. We empirically found that this number of passes is enough to help the algorithm find the optimal solution.

```
A: For Passes=1…40
B:    For Node=1…8
C:        For Step=1…6
D:            For Example=1…N (N is the total number of examples)
                  Probe + and – directions
                  Re-compute mark locations of each example
                  Do Procrustes to set
                  Do PCA to set
                  Compute new MDL cost
                  If new cost is lower then accept and break loop D
                  Undo a (node) change
D:            End of example loop
C:        End of step loop
B:    End of node loop
A: End of passes loop
```

In the experiment, the step length and the step number are related. They can be roughly calculated as follows:

$$\text{Distance between nearest two nodes} = \text{step length} \times \text{step number} \qquad (\mathbf{5.11})$$

This equation guarantees that every node can run everywhere along the boundary. For a fair comparison with MDL, a master example is used in both MDL and MEM. A master example is a manual landmarked example. During optimization, landmarks on this particular example are not allowed to move. If nodes and number of examples increase, the computation time will increase dramatically.

One of the problems for finding correspondence in a learning process is slow convergence. In the next section, we will calculate the gradient of our proposed cost function, in this way a variety of optimization techniques can be considered. Ericsson in [49] has shown a promising way to derive the gradient information from cost function by using Singular Vector Decomposition (SVD) [50]. We will adopt this method, and extract a gradient from our proposed cost function.

5.3.1 Introduction of SVD

How to calculate SVD

1) Find the eigenvalues of the matrix $A^T A$ and arrange them in descending order;
2) Find the number of nonzero eigenvalues of the matrix $A^T A$;
3) Find the orthogonal eigenvectors of the matrix $A^T A$ corresponding to the eigenvalues, and arrange them in the same order of form the column-vectors of the matrix $V \in R^{n\times n}$;
4) Form a diagonal matrix $S \in R^{m\times n}$ placing on the leading diagonal of it the square roots $\sqrt{\lambda_i}$ in a descending order;

Find the column vector of matrix $U \in R^{m\times m}$: $U = \dfrac{1}{S} AV^T$

A brief introduction of how to perform SVD is given at the above table. A basic theory of linear algebra is that any real or complex $M \times N$ matrix A can be factored into $A = USV^T$, where U is a $M \times M$ orthogonal matrix, V is an $N \times N$ orthogonal matrix and S is an $M \times N$ diagonal matrix with non-negative diagonal elements (also called singular values).

Next, we are going to present the connection between PCA and SVD and their application in ASM.

5.3.2 Connection between PCA and SVD in the Application of ASM

Recall from paragraphs 3.4 and 3.5 that the idea behind PCA is to extract shape variations and statistic weight of each variation. Similar results can be achieved by using SVD, as in the following equation.

$$X = USV^T \tag{5.12}$$

Since *U* and *V* are orthogonal matrices and *S* contains the singular values of *X*, factorizing *X* can give two eigenvalue factorizations related to *X* as in Equation 5.13.

$$XX^T = US^2U^T \text{ , } X^TX = VS^2V^T \tag{5.13}$$

Now, dividing both sides of equation with $n_s - 1$, and multiplying both sides with *V*, we get Equation 5.14.

$$\frac{1}{n_s - 1} XX^T V = VS^2 \frac{1}{n_s - 1} \tag{5.14}$$

By comparing Equation 5.14 and Equation 3.9, we can see that *V* is the eigenvector matrix and S^2 is the eigenvalue.

5.3.3 Derive a Gradient from the MEM Cost Function

In this section, the gradient of the cost function is going to be derived based on the above sections. Given the n^{th} landmark on the m^{th} shape, we denote that changing the position of this landmark is C_{mn}. Therefore, the gradient of cost function is

$\frac{\partial MEM}{\partial C_{mn}}$. Since during the experiment, *H* will not change dramatically, we assume that *H* is a constant in derivation. As a result, recalling from Equation 5.10, we will have Equation 5.15.

$$\frac{\partial MEM}{\partial C_{mn}} = \sum H_i \frac{\partial \lambda_i}{\partial C_{mn}} \qquad \textbf{(5.15)}$$

According to this equation, we can see that the derivative of the cost function is directly related with the derivative of eigenvalue of shape covariance matrix. Recalling the connection between PCA and SVD, we can have Equation 5.16.

$$\frac{\partial \lambda_i}{\partial C_{mn}} = \frac{\partial s_i^2}{\partial C_{mn}} = 2s_i \frac{\partial s_i}{\partial C_{mn}} = 2s_i \sum_j \frac{\partial s_i}{\partial x_{mj}} \frac{\partial x_{mj}}{\partial C_{mn}} \qquad \textbf{(5.16)}$$

Here, *s* is the diagonal matrix product from SVD, and x_{mj} is Cartesian coordinate for the m^{th} landmark on the j^{th} shape. $\frac{\partial x_{mj}}{\partial C_{mn}}$ is the surface gradient, which can be estimated using differential approximation.

We will focus on the derivation of the $\frac{\partial s_i}{\partial x_{mj}}$ part. Given a matrix X, which was composed by the concatenated shape vectors as it was defined in paragraph 3.3. Then, this matrix *X* is decomposed by using Singular Value Decomposition analysis. We are interested in computing the derivatives of the singular values s_i with respect to shape locations. Here we have equation 5.17.

$$\frac{\partial X}{\partial x_{ij}} = \frac{\partial U}{\partial x_{ij}} SV^T + U \frac{\partial S}{\partial x_{ij}} V^T + US \frac{\partial V^T}{\partial x_{ij}} \qquad \textbf{(5.17)}$$

Then we multiply Equation 5.17 with U^T on the left and V on the right, which leads to equation 5.18.

$$U\frac{\partial X}{\partial x_{ij}}V = U^T\frac{\partial U}{\partial x_{ij}}S + \frac{\partial S}{\partial x_{ij}} + S\frac{\partial V^T}{\partial x_{ij}}V \qquad \textbf{(5.18)}$$

Since $U^T\frac{\partial U}{\partial x_{ij}}$ and $S\frac{\partial V^T}{\partial x_{ij}}V$ are anti-symmetric, all their diagonal elements vanish. Recalling that S is a diagonal matrix, it is clear to say that the diagonal elements of $U^T\frac{\partial U}{\partial x_{ij}}S$ and $S\frac{\partial V^T}{\partial x_{ij}}V$ vanish too. Therefore, we have a conclusion in Equation 5.19.

$$\frac{\partial s_i}{\partial x_{mn}} = u_{mi}v_{ni} \qquad \textbf{(5.19)}$$

If we combine the results of Equations 5.15, 5.16 and 5.19, we come to derive Equation 5.20, which will be the gradient of the MEM cost function.

$$\frac{\partial MEM}{\partial C_{mn}} = 2\sum_{i=1}^{n_s} H_i s_{ns} u_{mi} V \frac{\partial X}{\partial x_{ij}} \qquad \textbf{(5.20)}$$

5.3.4 Gradient Descent Optimization

This section introduces Gradient Descent (also known as Steepest Descent) optimization method, which was used in 2D optimization of MEM. More details and other gradient-based optimization method can be found in [74].

The method is very simple; it is based on the observation that if the real-valued cost function $F\left(x\right)$ is differentiable in a neighbourhood of a point *A*. The cost function value decreases fastest if we move the point *A* along the opposite or orthogonal direction of gradient ($-\nabla F(x)$) at point *A*. With a free parameter of length of step γ, the next point *B* will be given in Equation 5.21:

$$b = a - \gamma \nabla F(x) \qquad \textbf{(5.21)}$$

Then, it can be seen that $F(a) \geq F(b)$. Keep this in mind and recall our MEM cost function and initial landmarks; we can start from the initial positions (b_0), we will have

$$F(b_0) \geq F(b_1) \geq \ldots, \qquad \textbf{(5.22)}$$

This process can be illustrated in the following Figure 5.5:

Therefore, the sequence of (x_n) will converge to a local minimum. Note that the value of the step size is allowed to change at every iteration.

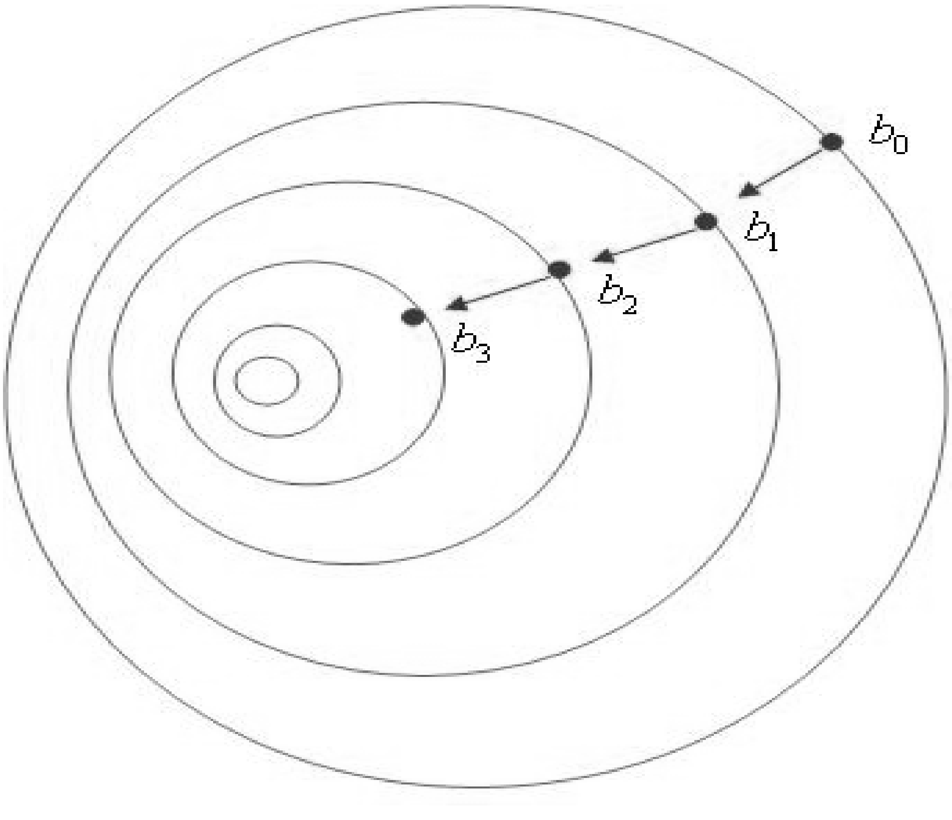


Figure 5.5 This graph shows the process of the steepest gradient optimization. It can be seen that the point is getting closer to the local minimum/maximum after each step.

In general, if the gradient of a function is know for a particular optimization problem, it generally pays off to use more sophisticated optimization techniques than Simplex or Simulated Annealing [49]. Here for comparison with MDL approach, we use the Steepest Descent optimization method in our proposed MEM approach.

5.3.5 Scheme of Optimizing MEM

With information about the gradient of cost function, we can use some more complicated optimization algorithm rather than the Genetic Algorithm [79], which is used by Davies et.al. [22]. The scheme of our optimization method is going to be shown in this section.

1) Initialization: This step will help prepare the parameters for optimization. First, each shape is labelled swith a specific number of landmarks. If we use *m* levels of landmarks to represent the shape model and optimize the first *n* levels of landmarks ($m \ge n$), then number of landmarks will be $2^m - 1$. These landmarks will be placed according to equal arc length rule.

2) Procrustes alignment: This step will involve processes of rotation, rescaling, and translation. In each iteration, corresponding affine shape error is minimized then normalized to unit shape. One important thing about this process is that in each iteration, the only change is the re-parameterization function as the curves are fixed. Landmarks on curves, rather than on the nodes, are estimated by interpolation.

3) MEM & Gradient: Based on the second step and using Equations 5.13 and 5.23, we can easily calculate the MEM cost function value and gradient from the parameterized landmarks.

4) Update Parameters: During this step, each landmark will be moved towards the MEM gradient direction. Local minima will be probed and new shape parameters will be generated at the same time. Finishing this step, if the process does not converge, it will go back to step 2.

The whole optimization scheme is shown in Figure 5.6.

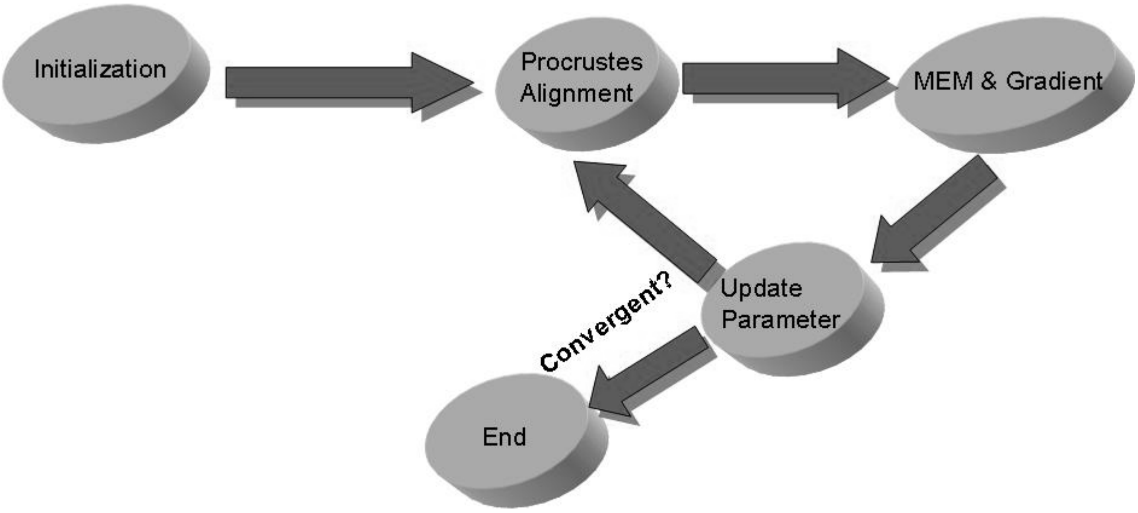


Figure 5.6 This graph is a brief scheme of optimization used in MEM algorithm. The inputs are shape surface meshes and the outputs are surfaces with landmarks.

5.4 Conclusions

In this chapter, we proposed a new framework of finding correspondence across dataset automatically. We treat the finding correspondence problem as a coding problem and measure the order degree of a statistical shape mode system by Entropy. Therefore, we argue that the minimum entropy status will hold the correct correspondence. In addition, we did not compose our MEM cost function by directly summing up energy components derived from each eigenvector as MDL does. Each Entropy component is assigned with a statistical weight concerning the different importance of each shape eigenvector. These weighting components make sense since they are based on the fact that the eigenvector with very small eigenvalues are mostly blurred by shape noise and we trust those eigenvector with large eigenvalues. Therefore, each eigenvector should be treated differently and energy component derived from each vector should also be assigned with a different weighting.

The optimization scheme of MEM algorithm in 2D is also introduced. For simplicity, we use the parameterization method introduced in MDL. This is also because we want to make a neutral comparison with MDL, and evaluate the performance difference only from the cost-function. Optimization is performed by mapping the 2D shapes, either closed or open curves, onto a curved line by using the normalized shape length. Then each node is assigned with a parameter, which is defined by the absolute distance between the two ends or one end in closed shapes scenario. During optimization, nodes are manipulated by shape re-parameterization. A linear piecewise re-parameterization method was discussed. MDL uses a local search optimization method. One problem of the MDL approach has been the slow convergence of the optimization step. In MEM, a gradient based steepest descent optimization method is used. Singular Value Decomposition was introduced in this chapter, which has a direct connection with PCA. By using SVD, we successfully derived the Jacobian of MEM cost function; therefore, a variety of optimization techniques can be considered.

In conclusion, MDL and MEM share the same scheme of shape parameterization method, however, MDL uses a local search scheme and MEM uses a gradient based optimization method. Both algorithms start optimization from equal spaced positions. In the next chapter, we will discuss the quantitative comparisons between MDL and MEM on various types of 2D datasets.

Chapter 6 Experiments and Results in 2D

The previous chapters have described and discussed the development of an automatically constructing method for finding correspondence across datasets, and evaluating schemes of different optimization methods. In this chapter, numerous experiments are performed on different kinds of 2D datasets for different validation purposes. The experiments will be conducted in several ways: closed curves, open curves with free ends, open curves with fixed ends. In each case, experiments are guided to compare performances between MDL and MEM. Datasets in this section are from Thodberg and Ericsson as used in references [49],[71], after personal communications.

By using the criteria discussed in Chapter 4, it is shown that our MEM shows better performance on Generalization Ability and Specificity and similar in Compactness. One more experiment is carried out with the purpose of testing the control of “Pile Up” problem and its comparison between MDL and MEM. The preliminary results show that the MEM algorithm can make improvements in solving the “Pile Up” problem.

6.1 Experiments on Closed Curves

In this experiment, we will try both MDL and MEM on three different datasets, which are 24 contours of metacarpals (all closed curves，see Figure 6.1), 15 flying birds contours, and 20 Mickey Mouse like cartoon contours. For a fair comparison, initial conditions and parameters are the same for both methods. For example, both methods start from equal spaced position landmarks (containing 8 nodes and 64 marks), and each node can move freely along the contours (one master example is used in MDL algorithm).

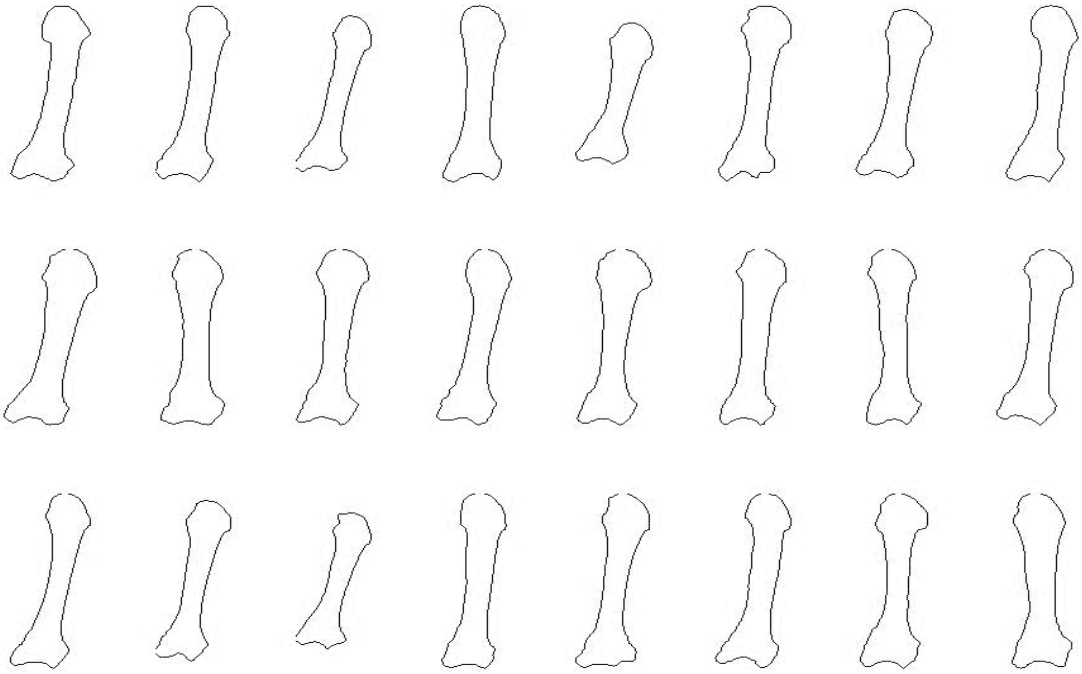


Figure 6.1 24 Contours of metacarpals, with different orientations, sizes and shapes.

All these 24 datasets, each is saved as 281 coordinates of points along the boundaries. During optimization, if a new coordinate needs to be found between existing points, a 2-Dimension interpolation will be used.

After MEM converged, it can be seen from the figure above that 8 nodes are placed at corresponding locations in the seemingly same manner in Figure 6.2. In the same time, we also show the correspondence results found by MDL in Figure 6.3.

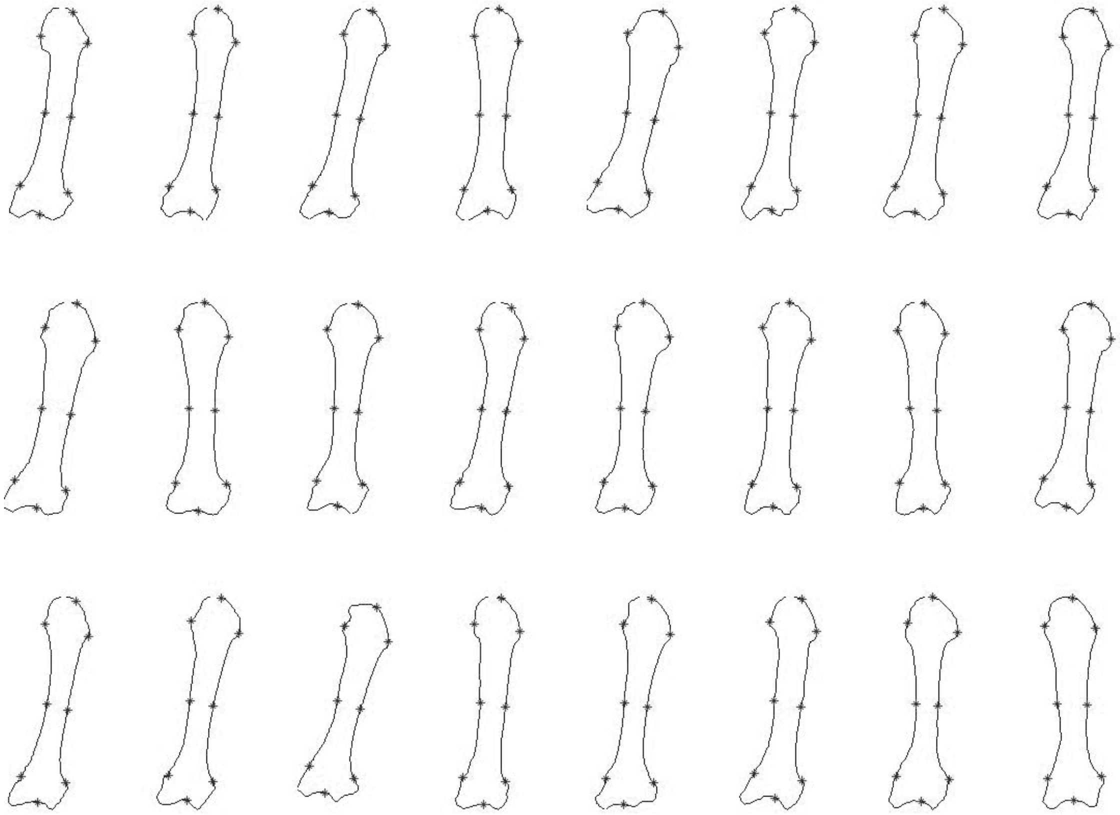


Figure 6.2 24 contours of metacarpals with 8 nodes on them, landmarks are found by MEM.

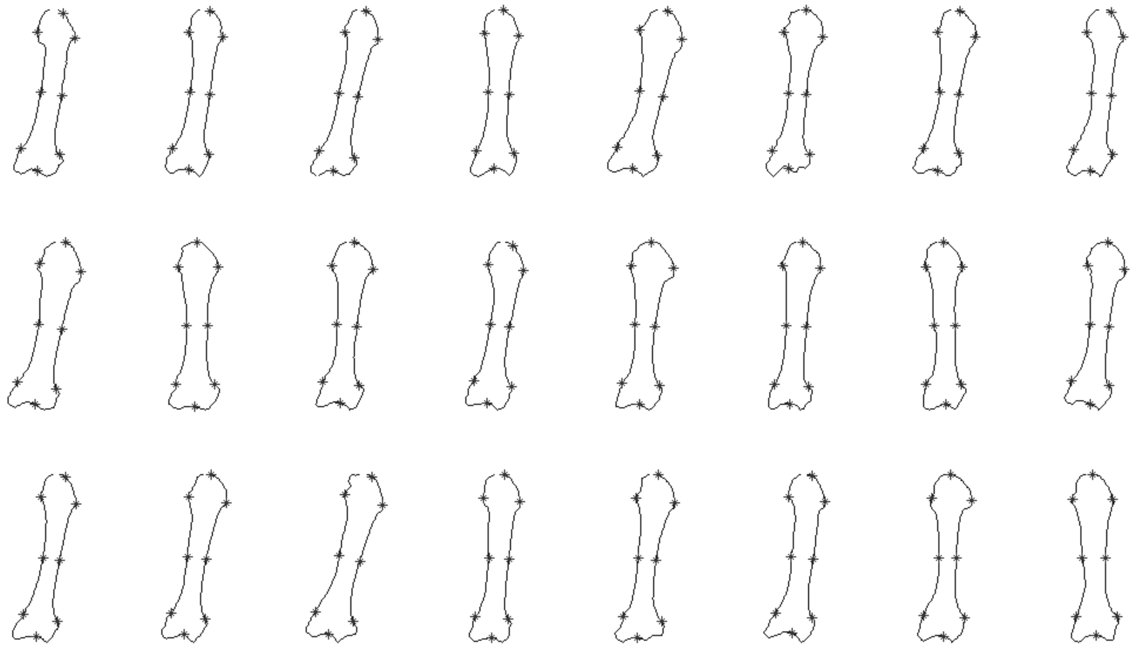


Figure 6.3 24 contours of metacarpals with 8 nodes on them, landmarks are found by MDL.

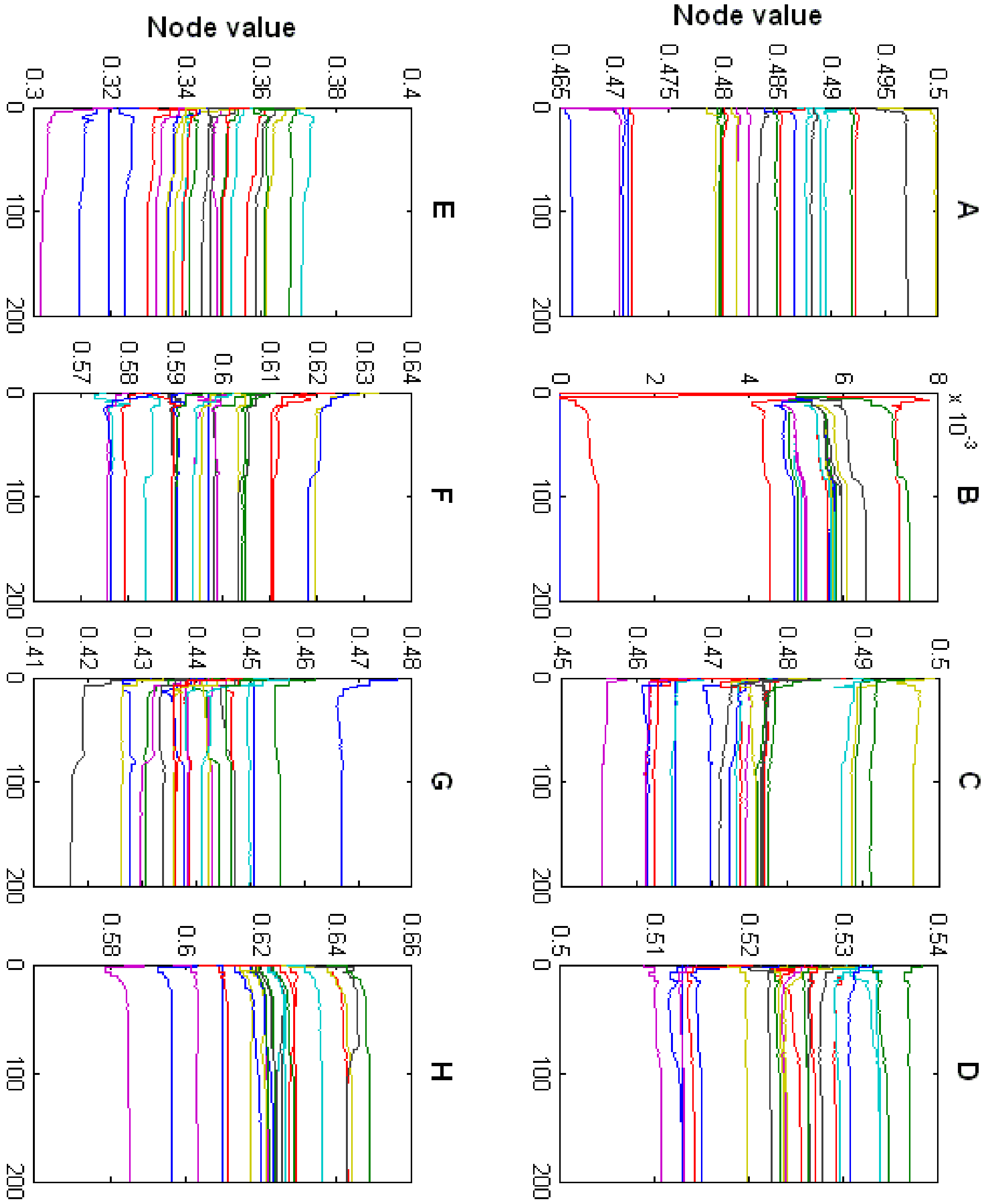


Figure 6.4 Shown is the process of node value changing during the experiment on metacarpals. There are 8 graphs representing 8 different nodes (from A to H) and in each graph 24 lines with different colours represent nodes from the 24 examples. The X-axis represents numbers of steps in optimization; Y-axis represents the node parameter value. The general idea of Node Parameter can be recalled from Figure 5.4.

In the process of MEM, it can be seen from Figure 6.4 that the node value will change in different steps until eventually stabilized. Figure 6.5 shows the output of MEM cost-function by using steepest gradient method and output of MDL by using Thodberg's approach [38].

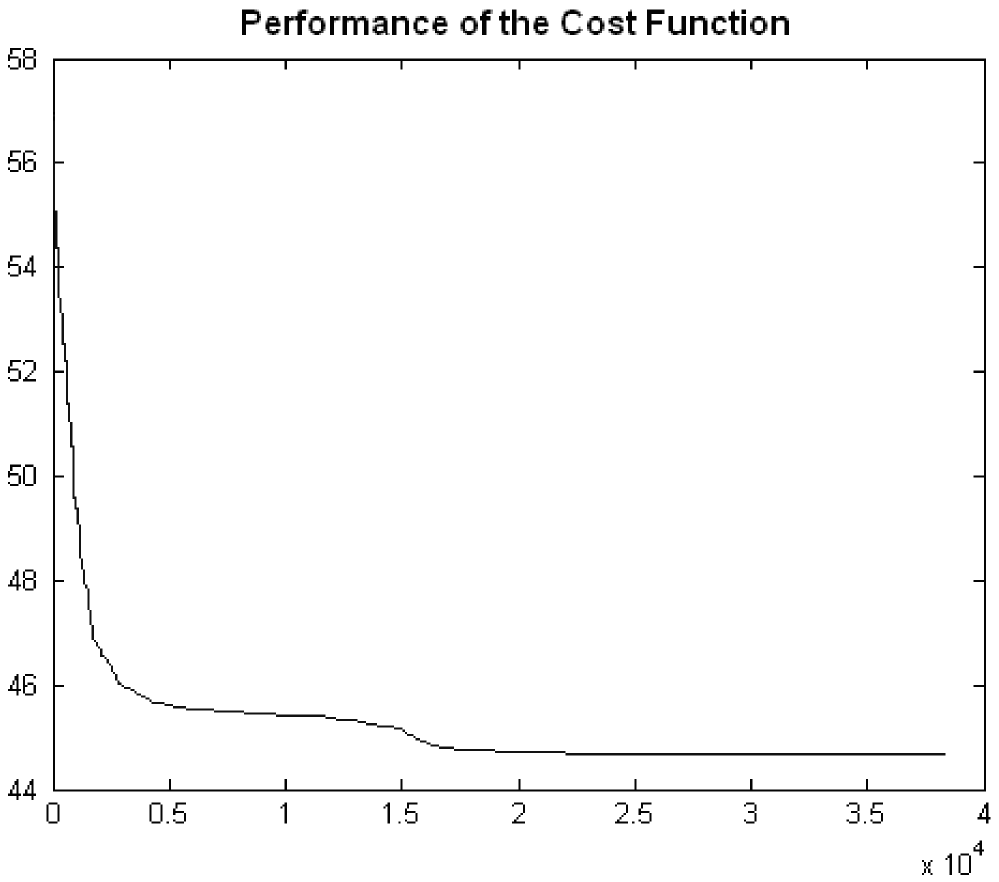


Figure 6.5 A demonstration of the output of the cost-function. X-axis represents number of steps in time and Y-axis is the corresponding value of cost function. The blue solid line shows the MEM performance by using the Steepest Descent method.

From Figure 6.5 we can see that the value of the MEM cost function drops dramatically in the first 2500 steps then stabilizes and converges afterwards. During experiments, we found the MEM approach will converge approximate 2.3 times faster than Thodberg's approach.

An ASM is performed by using the corresponding points found by MEM, in Figure 6.6 the effect of the first three principal components is shown. Then comparisons between MDL and MEM are implemented on three shape model properties: Generalization Ability, Specificity and Compactness.

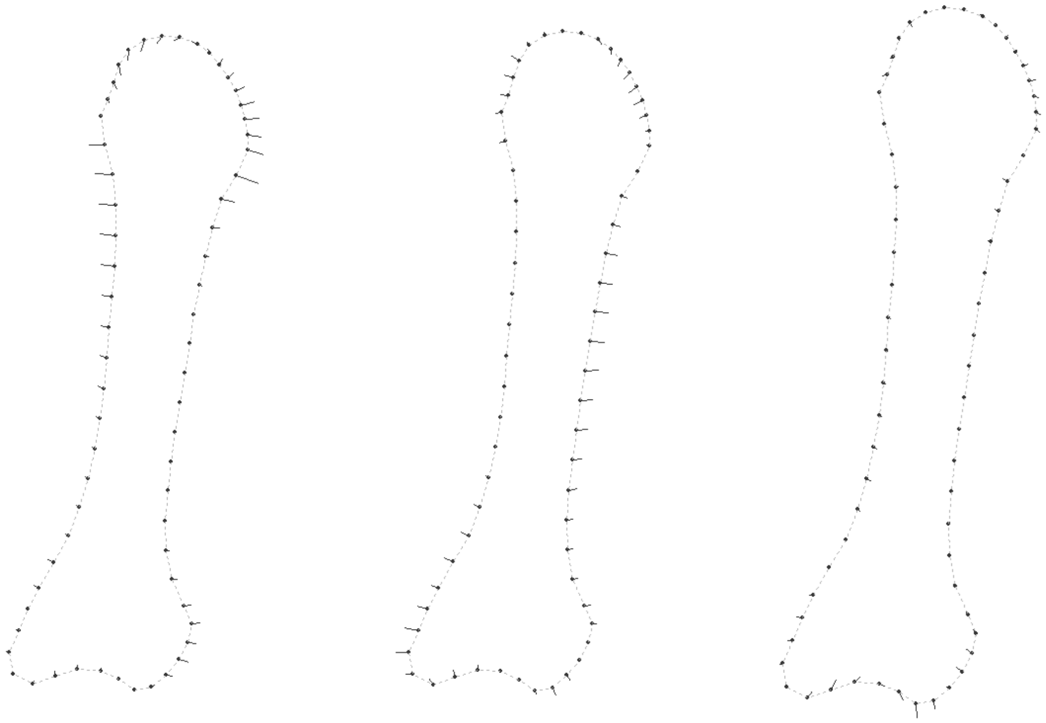


Figure 6.6 Shown is the mean shape with red marks; the whiskers starting from the marks indicate three standard deviations of the first three principal components.

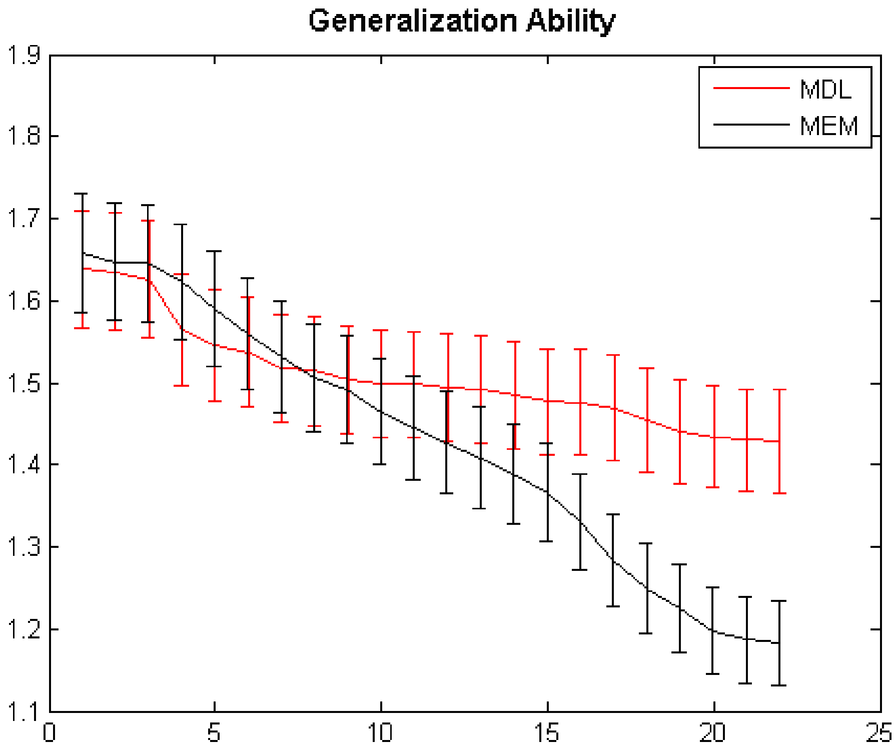


Figure 6.7 Generalization Ability comparison on closed curve. X-axis represents number of shape modes and Y-axis represents Generalization Ability.

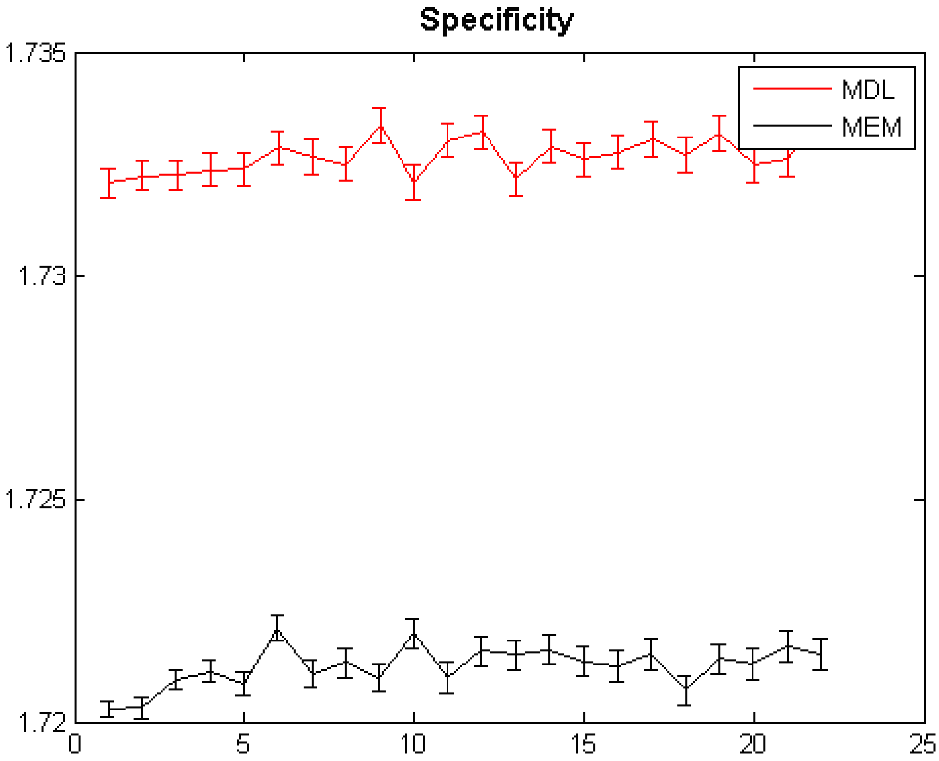


Figure 6.8 Specificity comparison on closed curve. X-axis represents number of shape modes and Y-axis represents Specificity.

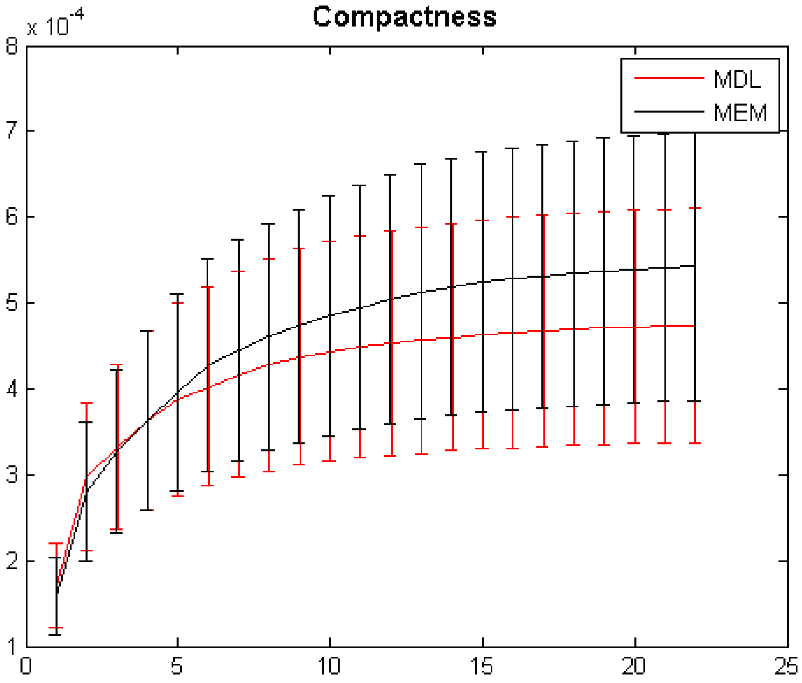


Figure 6.9 Compactness comparisons on closed curve. X-axis represents number of shape modes and Y-axis represents Compactness.

		MEM	MDL	Percentage Difference
Generalization Ability	Mean – Standard Deviation	**28.72832358**	29.6247037	3.072266049%
	Mean	**29.93930458**	31.57360417	5.313680068%
	Mean + Standard Deviation	**31.15028559**	32.85005712	5.311757577%
Specificity	Mean – Standard Deviation	**36.37891469**	37.00119245	1.696039383%
	Mean	**37.00788914**	37.25340015	0.661208582%
	Mean + Standard Deviation	**37.01458584**	37.26185859	0.665817423%
Compactness	Mean – Standard Deviation	71.64381592	**63.77564717**	11.62044002%
	Mean	100.5747843	**91.81160115**	9.109982607%
	Mean + Standard Deviation	129.5057526	**118.1826462**	9.14302525%

Table 6.1. A quantitative analysis on the three criteria comparisons based on Area Under the Curve. The smaller corresponding value is marked in bold character.

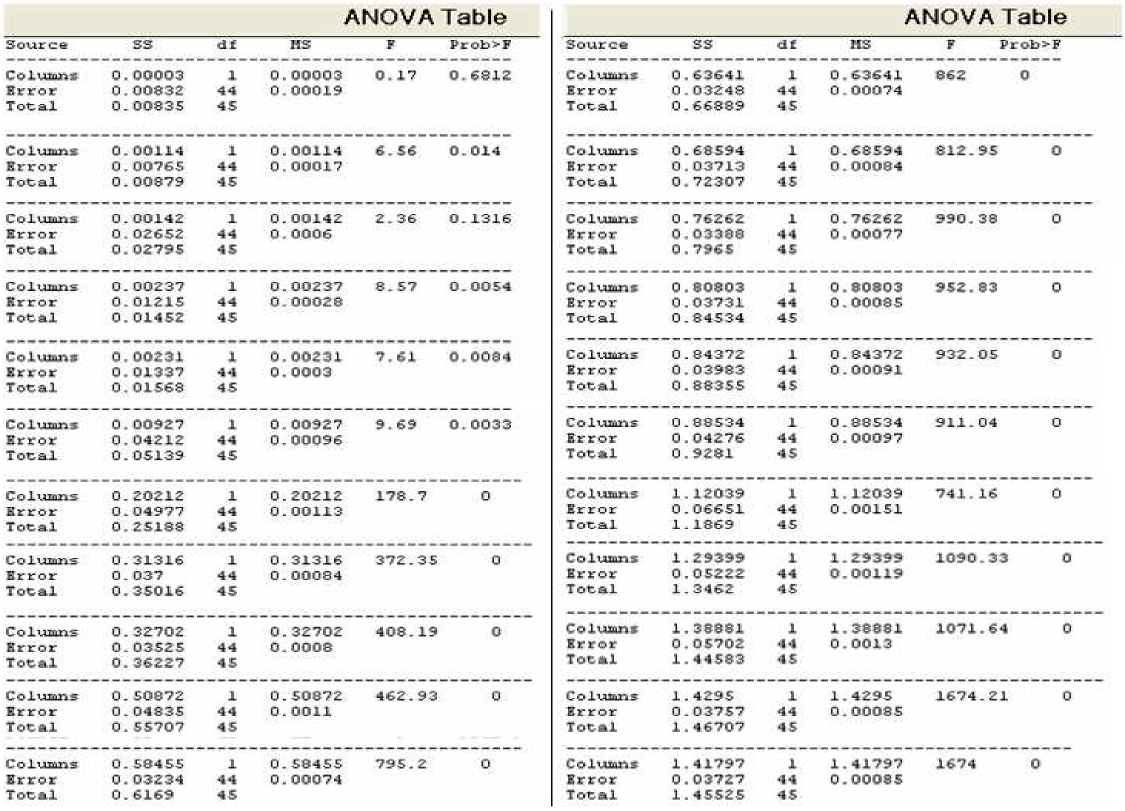
Since it is difficult to justify how many shape variations are suitable in experiments, we make a conclusion of a shape model's performance if one mode achieves better performance in most of the number of shape variations (*M*). This manner of evaluation method is also suggested by Davies [22].

As we can see from Figure 6.7, for 14 out of 22 shape variations, MEM achieves better performance in Generalization Ability ($G(MEM) < G(MDL)$). Therefore, we can conclude MEM achieved better performance in Generalization Ability. In Figure 6.8, MEM also achieved better performance in Specificity in all shape modes, since $S(MEM) < S(MDL)$. In Figure 6.9, *C(MEM)* is slightly larger than *C(MDL)*, but considering the error for each *M*, we can say that these two methods offer similar Compactness level or MEM is a bit worse than MDL.

We also estimated the Area Under the Curve (AUC) in order to quantitatively calculate the difference between the three criteria. The AUC value is calculated for each criteria of mean value, mean value minus standard deviation and mean value plus standard deviation. The results of AUC are presented in Table 6.1, where smaller values are showed in bold characters. It can be seen that for Generalization Ability, MEM is better than MDL from 3% to 5.3%, for Specificity, MEM is better than MDL from 0.6% to 1.7% and MEM is worse than MDL from 9% to 11% in Compactness.

For further analysis on the results of our quantitative comparisons, we perform ANOVA test on Generalization Ability and Specificity (Tables 6.2 and 6.3). The purpose of two-way ANOVA is to find out whether data from several groups have a common mean. In this thesis, we perform two-way ANOVA. One-way ANOVA and two-way ANOVA differ in that the groups in two-way ANOVA have two categories of defining characteristics instead of one. The standard ANOVA table has columns for the sums of squares (SS), degrees-of-freedom (df), mean squares (SS/df), *F* statistics and *p*-values. We therefore can use the *F* statistics to do hypothesis tests in order to find out if the results are from two groups or just one. For example, if the *p*-value is near to zero, it means a strong indication that the two groups are from different distributions.

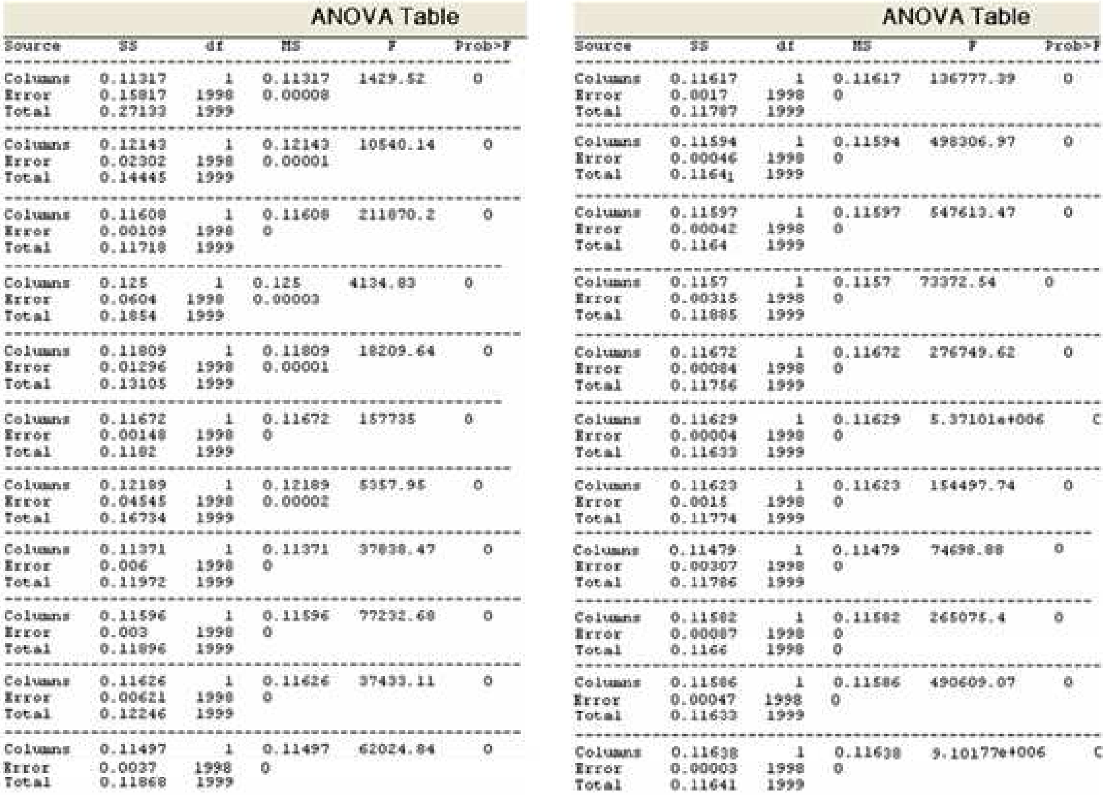
Compactness is excluded from this statistical test since Compactness is not derived from samples like the Generalization Ability and Specificity. From the results, since most of the *p*-value is zero or close to zero, we can conclude that the null hypothesis is rejected in most of the experiments and therefore the MDL and MEM results are



```
                                        ANOVA Table
Source       SS       df       MS         F      Prob>F
-------------------------------------------------------
Columns   0.00003      1    0.00003     0.17    0.6812
Error     0.00832     44    0.00019
Total     0.00835     45

-------------------------------------------------------
Columns   0.00114      1    0.00114     6.56    0.014
Error     0.00765     44    0.00017
Total     0.00879     45
-------------------------------------------------------
Columns   0.00142      1    0.00142     2.36    0.1316
Error     0.02652     44    0.0006
Total     0.02795     45
-------------------------------------------------------
Columns   0.00237      1    0.00237     8.57    0.0054
Error     0.01215     44    0.00028
Total     0.01452     45
-------------------------------------------------------
Columns   0.00231      1    0.00231     7.61    0.0084
Error     0.01337     44    0.0003
Total     0.01568     45
-------------------------------------------------------
Columns   0.00927      1    0.00927     9.69    0.0033
Error     0.04212     44    0.00096
Total     0.05139     45
-------------------------------------------------------
Columns   0.20212      1    0.20212    178.7       0
Error     0.04977     44    0.00113
Total     0.25188     45
-------------------------------------------------------
Columns   0.31316      1    0.31316    372.35       0
Error     0.037       44    0.00084
Total     0.35016     45
-------------------------------------------------------
Columns   0.32702      1    0.32702    408.19       0
Error     0.03525     44    0.0008
Total     0.36227     45
-------------------------------------------------------
Columns   0.50872      1    0.50872    462.93       0
Error     0.04835     44    0.0011
Total     0.55707     45
-------------------------------------------------------
Columns   0.58455      1    0.58455    795.2       0
Error     0.03234     44    0.00074
Total     0.6169      45
```

```
                                        ANOVA Table
Source       SS       df       MS         F      Prob>F
-------------------------------------------------------
Columns   0.63641      1    0.63641    862       0
Error     0.03248     44    0.00074
Total     0.66889     45

-------------------------------------------------------
Columns   0.68594      1    0.68594    812.95       0
Error     0.03713     44    0.00084
Total     0.72307     45
-------------------------------------------------------
Columns   0.76262      1    0.76262    990.38       0
Error     0.03388     44    0.00077
Total     0.7965      45
-------------------------------------------------------
Columns   0.80803      1    0.80803    952.83       0
Error     0.03731     44    0.00085
Total     0.84534     45
-------------------------------------------------------
Columns   0.84372      1    0.84372    932.05       0
Error     0.03983     44    0.00091
Total     0.88355     45
-------------------------------------------------------
Columns   0.88534      1    0.88534    911.04       0
Error     0.04276     44    0.00097
Total     0.9281      45
-------------------------------------------------------
Columns   1.12039      1    1.12039    741.16       0
Error     0.06651     44    0.00151
Total     1.1869      45
-------------------------------------------------------
Columns   1.29399      1    1.29399    1090.33       0
Error     0.05222     44    0.00119
Total     1.3462      45
-------------------------------------------------------
Columns   1.38881      1    1.38881    1071.64       0
Error     0.05702     44    0.0013
Total     1.44583     45
-------------------------------------------------------
Columns   1.4295       1    1.4295     1674.21       0
Error     0.03757     44    0.00085
Total     1.46707     45
-------------------------------------------------------
Columns   1.41797      1    1.41797    1674       0
Error     0.03727     44    0.00085
Total     1.45525     45
```

Table 6.2. ANOVA table of the Generalization Ability on datasets of closed curves



```
                                        ANOVA Table
Source       SS        df       MS          F        Prob>F
-----------------------------------------------------------
Columns   0.11317       1    0.11317    1429.52        0
Error     0.15817    1998    0.00008
Total     0.27133    1999
-----------------------------------------------------------
Columns   0.12143       1    0.12143    10540.14        0
Error     0.02302    1998    0.00001
Total     0.14445    1999
-----------------------------------------------------------
Columns   0.11608       1    0.11608    211870.2        0
Error     0.00109    1998    0
Total     0.11718    1999
-----------------------------------------------------------
Columns   0.125         1   0.125      4134.83       0
Error     0.0604     1998   0.00003
Total     0.1854     1999
-----------------------------------------------------------
Columns   0.11809       1    0.11809    18209.64        0
Error     0.01296    1998    0.00001
Total     0.13105    1999
-----------------------------------------------------------
Columns   0.11672       1    0.11672    157735       0
Error     0.00148    1998    0
Total     0.1182     1999
-----------------------------------------------------------
Columns   0.12189       1    0.12189    5357.95       0
Error     0.04545    1998    0.00002
Total     0.16734    1999
-----------------------------------------------------------
Columns   0.11371       1    0.11371    37838.47        0
Error     0.006      1998    0
Total     0.11972    1999
-----------------------------------------------------------
Columns   0.11596       1    0.11596    77232.68        0
Error     0.003      1998    0
Total     0.11896    1999
-----------------------------------------------------------
Columns   0.11626       1    0.11626    37433.11        0
Error     0.00621    1998    0
Total     0.12246    1999
-----------------------------------------------------------
Columns   0.11497       1    0.11497    62024.84        0
Error     0.0037     1998    0
Total     0.11868    1999
```

```
                                        ANOVA Table
Source       SS        df       MS          F           Prob>F
--------------------------------------------------------------
Columns   0.11617       1    0.11617    136777.39        0
Error     0.0017     1998    0
Total     0.11787    1999
--------------------------------------------------------------
Columns   0.11594       1    0.11594    498306.97        0
Error     0.00046    1998    0
Total     0.11641    1999
--------------------------------------------------------------
Columns   0.11597       1    0.11597    547613.47        0
Error     0.00042    1998    0
Total     0.1164     1999
--------------------------------------------------------------
Columns   0.1157        1    0.1157    73372.54       0
Error     0.00315    1998    0
Total     0.11885    1999
--------------------------------------------------------------
Columns   0.11672       1    0.11672    276749.62        0
Error     0.00084    1998    0
Total     0.11756    1999
--------------------------------------------------------------
Columns   0.11629       1    0.11629    5.37101e+006        C
Error     0.00004    1998    0
Total     0.11633    1999
--------------------------------------------------------------
Columns   0.11623       1    0.11623    154497.74        0
Error     0.0015     1998    0
Total     0.11774    1999
--------------------------------------------------------------
Columns   0.11479       1    0.11479    74698.88       0
Error     0.00307    1998    0
Total     0.11786    1999
--------------------------------------------------------------
Columns   0.11582       1    0.11582    265075.4       0
Error     0.00087    1998    0
Total     0.1166     1998    0
--------------------------------------------------------------
Columns   0.11586       1    0.11586    490609.07        0
Error     0.00047    1998    0
Total     0.11633    1999
--------------------------------------------------------------
Columns   0.11638       1    0.11638    9.10177e+006        C
Error     0.00003    1998    0
Total     0.11641    1999
```

Table 6.3. ANOVA table of the Specificity on datasets of closed curves

statistically different (Tables 6.2 and 6.3). Results on the other datasets are also presented in this section, beginning with the datasets of flying birds with landmarks found by MEM, see Figure 6.10.

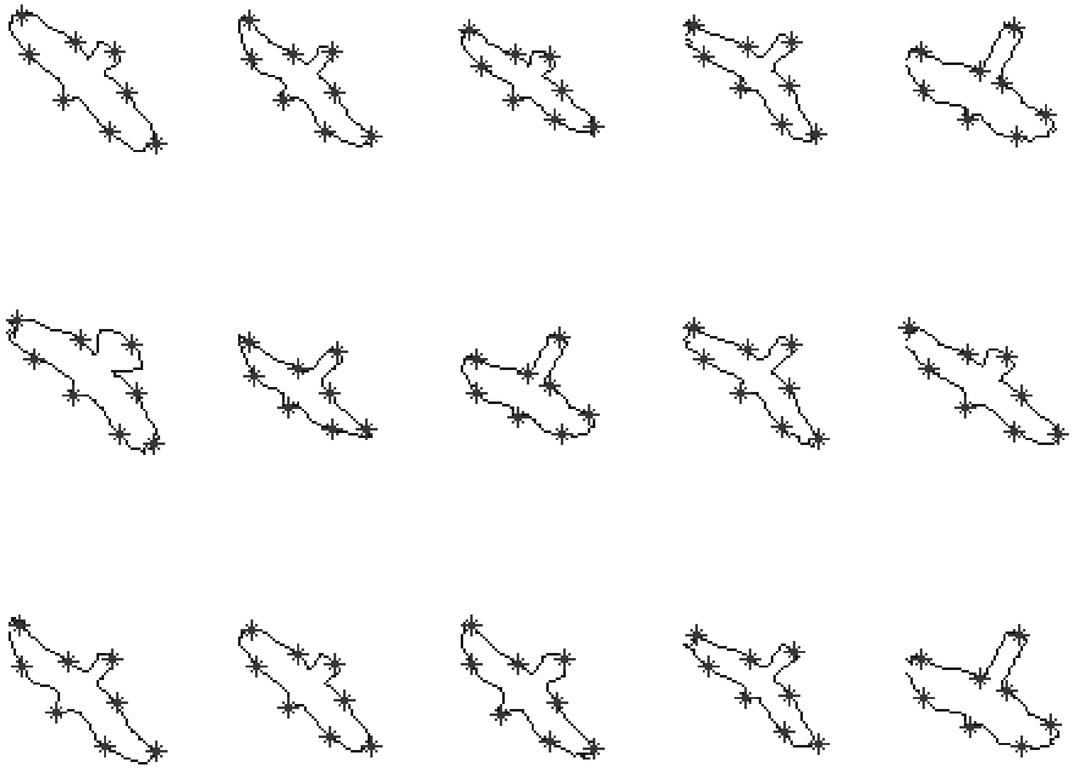


Figure 6.10 Graph shows the 8 landmarks found by our MEM algorithm on the dataset of 15 flying birds.

Figure 6.10 shows the correspondence results by MEM on datasets of 15 flying bird examples. From the results, we can observe that the correspondence points are marked in the same manner. For example, the points on birds head, wings and tails are marked in the seemingly same locations. In the example showed in Figure 6.10, we used only 8 control points in the optimization. However, we used more points in the reconstruction of Active Shape Model. In this experiment, we used a total of 64 points, the rest 56 (64-8) points are set equally spaced in between existing points. This strategy can effectively save computation time and only allows the algorithm to be optimized to the accuracy we need. Figure 6.11 shows the output of the MEM cost-function. It can be observed that cost-function will stabilize eventually. During experiments, we observe that the MEM approach can be optimized approximately 1.9 times faster than the MDL approach. This is due to the usage of the steepest descent optimization algorithm.

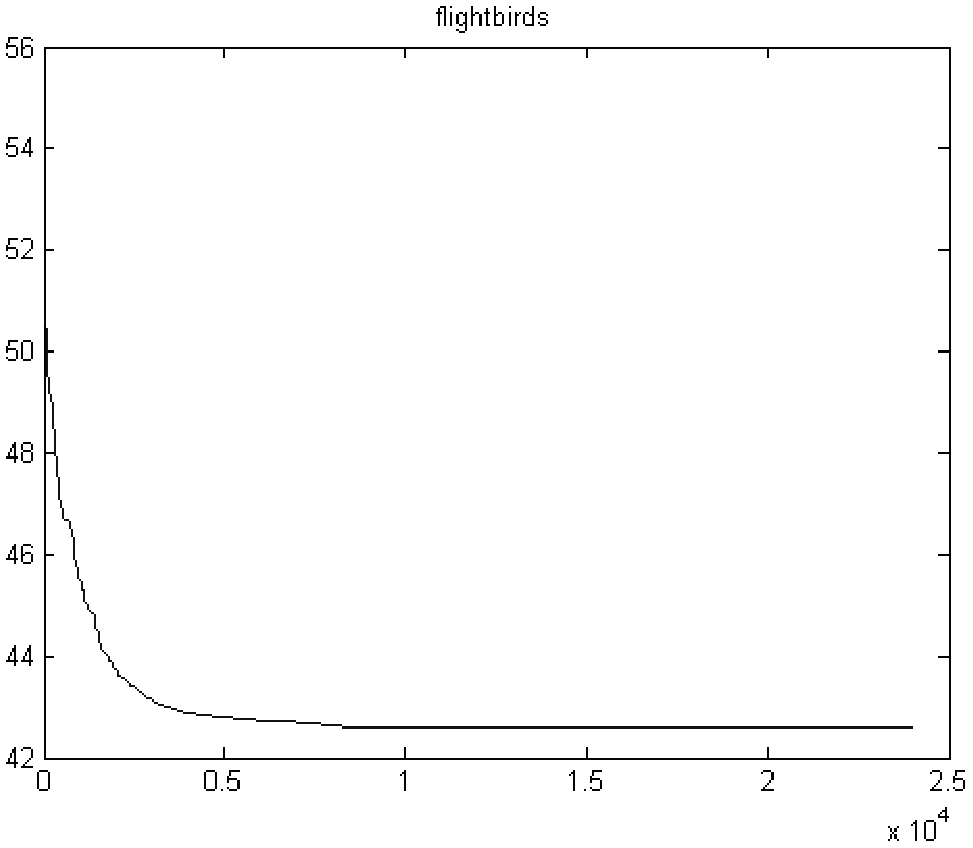


Figure 6.11 A demonstration of output of cost-function during optimization. X-axis represents number of steps time in optimization and Y-axis represents value of the cost function. The blue solid line shows the MEM performance by using steepest descent method.

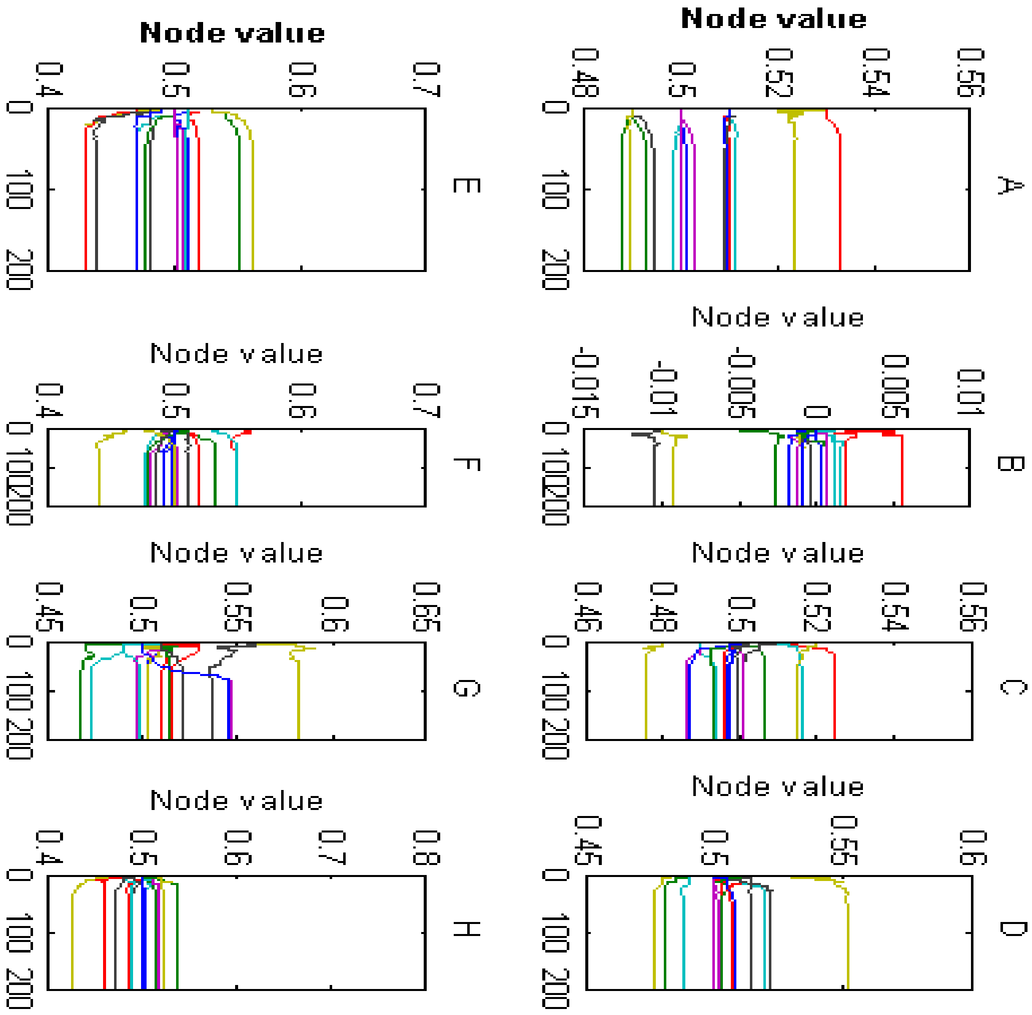


Figure 6.12 Shown is the process of node value changing during the experiment on flying birds. There are 8 graphs representing 8 different nodes (from A to H) and in each graph 15 lines with different colours representing nodes from the 15 examples. The X-axis represents numbers of steps in optimization, Y-axis represents the node parameter value.

Figure 6.12 demonstrates the location changes of the 8 control points during optimization. From this figure, it can be seen that the all 8 control points start from location parameter 0.5, since we are using absolute curve length as a parameterization method, 0.5 means that the points are equally spaced. During the optimization, each control point's location parameters first change dramatically, then after a few steps, achieve stabilization.

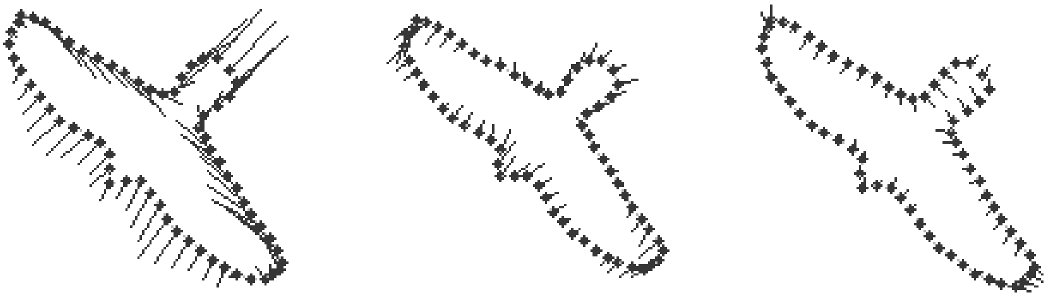


Figure 6.13 Graph shows the effect of first 3 shape variations.

Figure 6.13 shows the first three main shape variations captured by Active Shape Model. In this figure, the gray round dot shows the mean position of the flying birds and each node is assigned with an arrow pointing along the eigenvector captured by the shape model. It can be observed from Figure 6.13 that the first eigenvector captures most of the shape variations and the variations exist on most of the shape contours except both ends of the wings.

Figure 6.14 shows another example of using MEM to capture the correspondence points across the dataset of Mickey Mouse like cartoons. The datasets change in different size and different shapes. From the results, we can see that the 8 control points are placed in the same manner. The correspondence points lies on ears, feet, mouth, hand and tail.

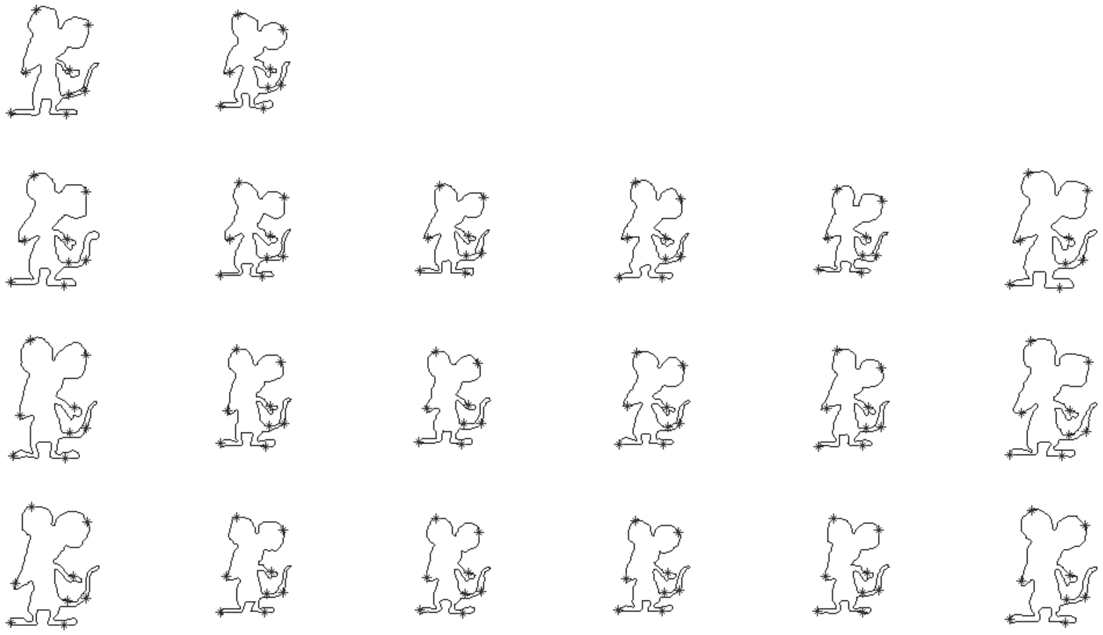


Figure 6.14 Graph shows the 8 landmarks found by our MEM algorithm on the dataset of 20 Mickey Mouse like cartoons.

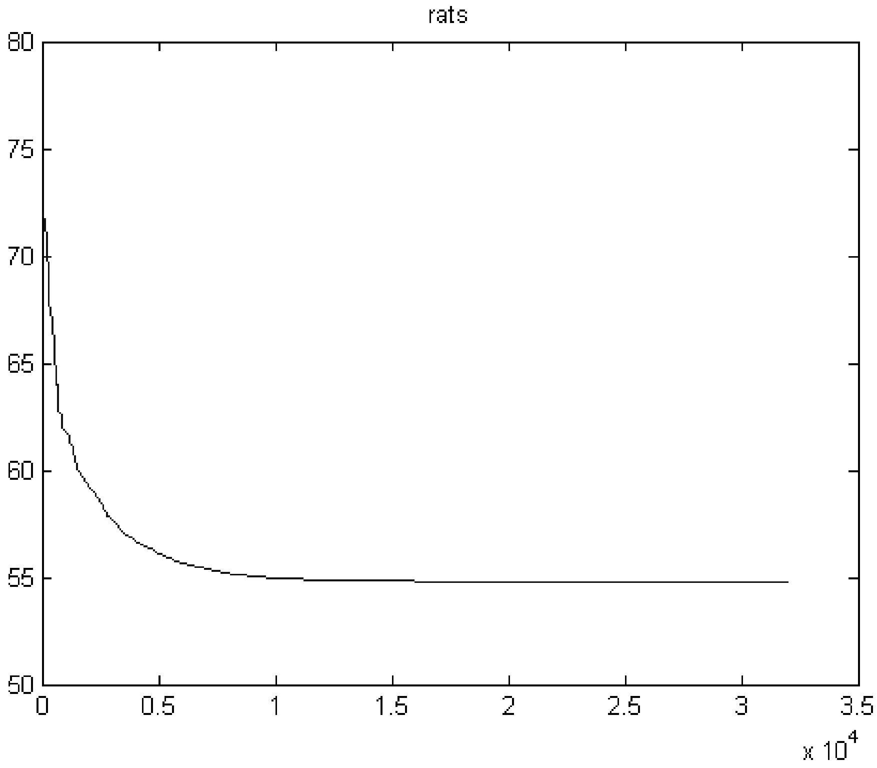


Figure 6.15 Graph shows the performance of cost function. The X-axis represents the number of steps and Y-axis represents the value of cost function. The blue solid line shows the output of MEM cost-function performance by using the Steepest Descent method.

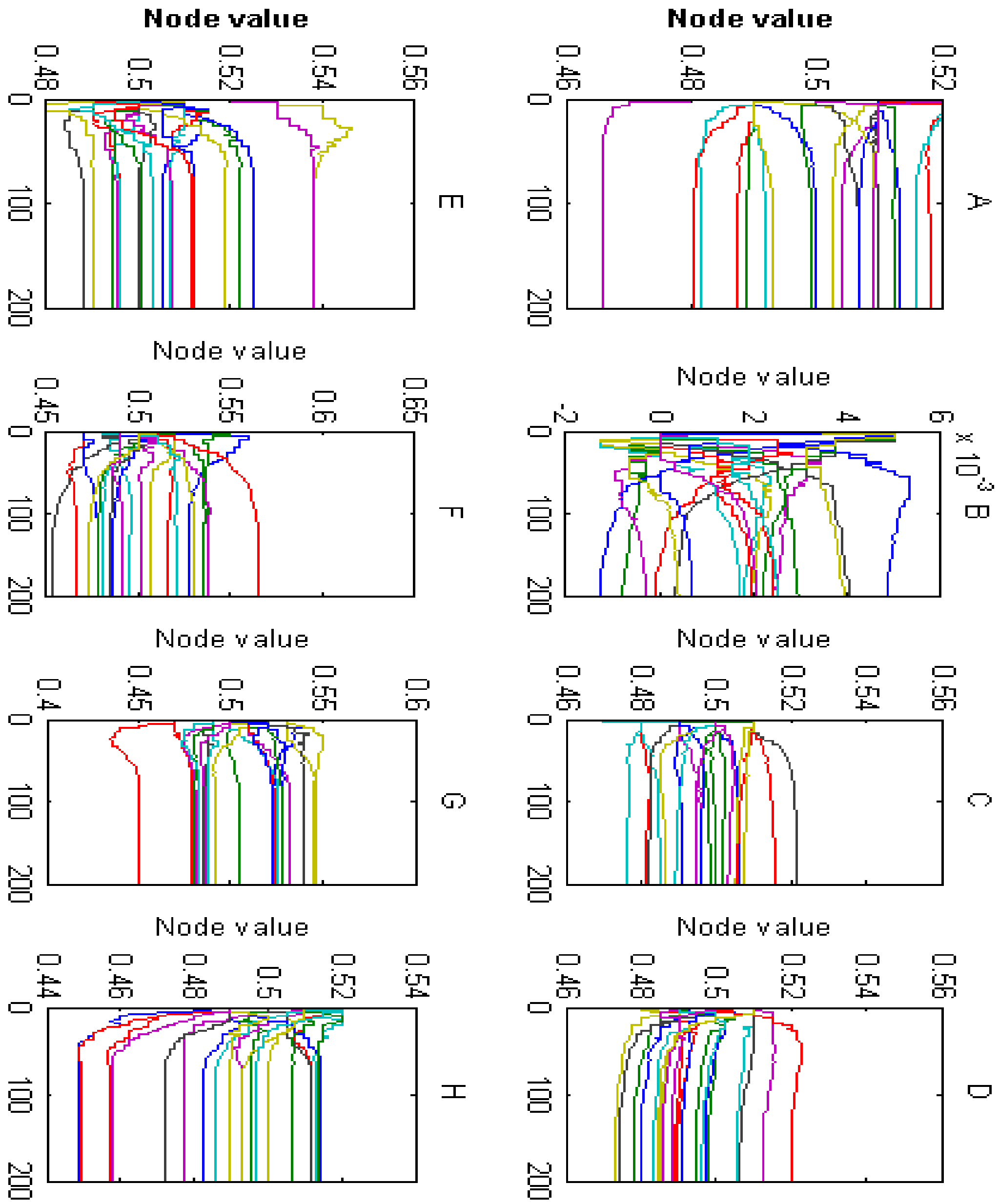


Figure 6.16 Graph shows the process of node value changing during the experiment on Mickey Mouse like cartoon. There are 8 graphs representing 8 different nodes (from A to H) and in each graph 20 lines with different colours representing nodes from the 15 examples. The X-axis represents numbers of steps in optimization, Y-axis represents the node parameter value.

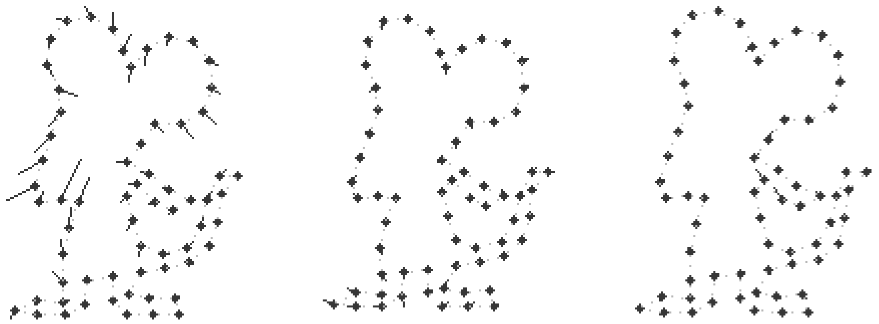


Figure 6.17 Graph shows the effect of first three shape modes.

Figure 6.15 shows the output of the MEM cost-function. From experiments, we observe that MEM can be optimized approximately 2.3 times faster than the MDL, which is due to the usage of Steepest Descent optimization method. Figure 6.16 shows the parameters change during optimization. As the previous experiment on birds, after a few steps of optimization, MEM finally achieves stabilization. Figure 6.17, shows the Mickey Mouse like shape variations captured by Active Shape Model. From this figure, it can be seen that the first shape variations is more prominent than the rest two variations and the shape variations mainly concentrated on the head of Mickey.

From the experiments performed above, it can be seen that the new MEM algorithm can find correspondence points across closed curve datasets in a reasonable same manner. MEM by using steepest descent method can converge faster than Thodberg's MDL approach [38]. Furthermore, in comparison tests, MEM outweighed MDL in both Specificity and Generalization Ability evaluations and achieved similar performance in Compactness.

6.2 Experiments on Open Curve with Fixed Ends

In this experiment, we will perform both MDL and MEM on a dataset composed with 32 contours of femurs (all open curves, see Figure 6.18). We will fix explicit ends to each example, which are the 1^{st} point and 32^{nd} point.

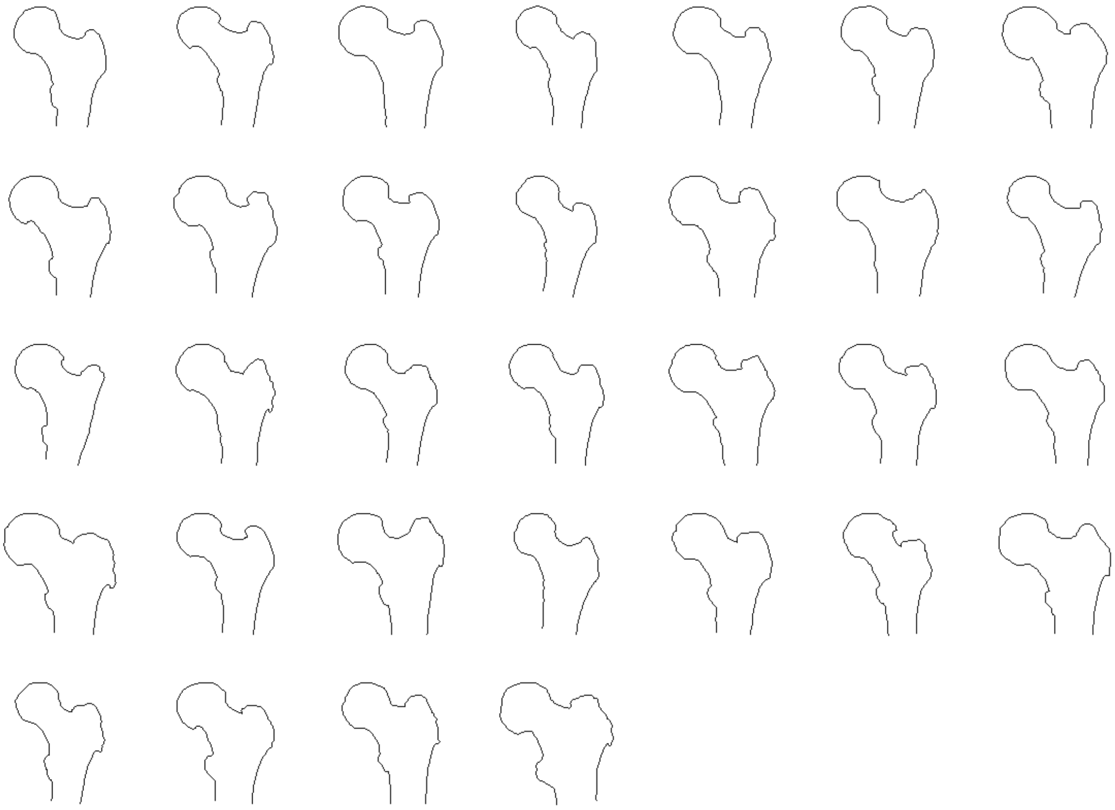


Figure 6.18 Examples of 32 contours of femurs taken from different patients.

For a fair comparison, we use one master example (marked by an expert), 9 nodes and 65 marks for both MDL and MEM (two fixed points that can not be moved in optimization have been pre-placed). All the points in each example start with node parameter 0.5 except the fixed points in the end of the shapes. During optimization, points will move along the shape contours to find the final optimal result. The process of node movement is shown in Figure 6.21. After each step, the value of the cost function will decline until stabilization. After optimization by MEM algorithm, we can see from Figure 6.19 that four level, 9 nodes have been placed across 32 datasets in a reasonably same manner. The first level is blue that has been fixed, the second level is green that lies in between blue, the third level is black that lie between blue and red and the forth level is red that lie either between blue and black or between black and green.

The correspondence points found by MDL is also shown in Figure 6.20.

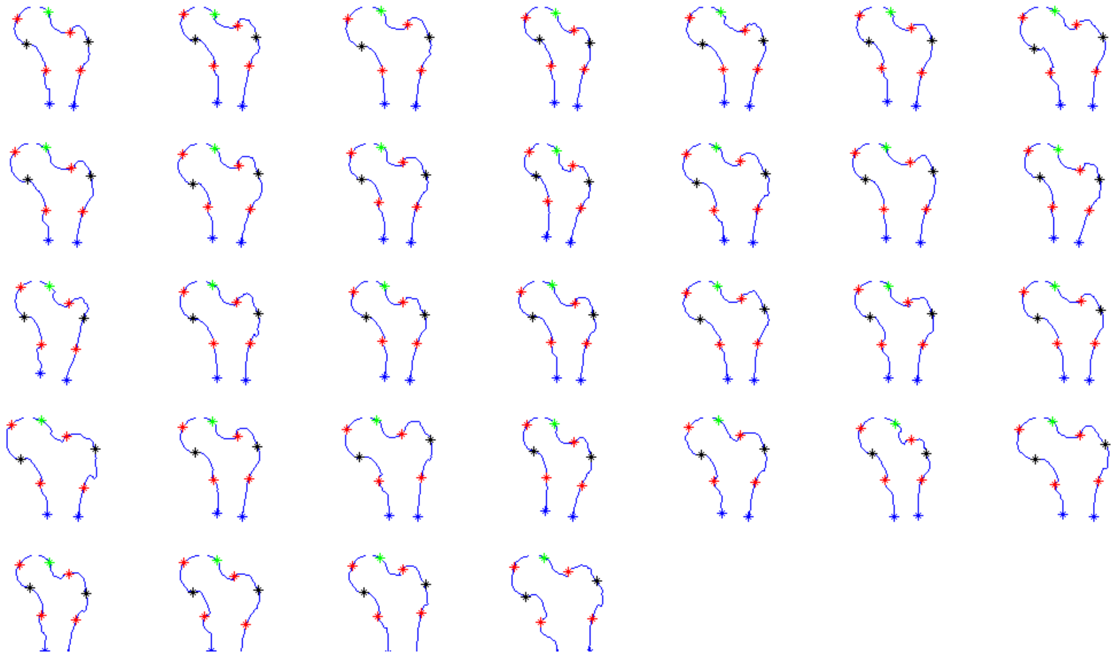


Figure 6.19 Results after optimization by MEM. It can be seen that four levels of nodes are placed along the boundary curves with the blue colour representing the fixed points. The first one is the master example that has been pre-processed by an expert.

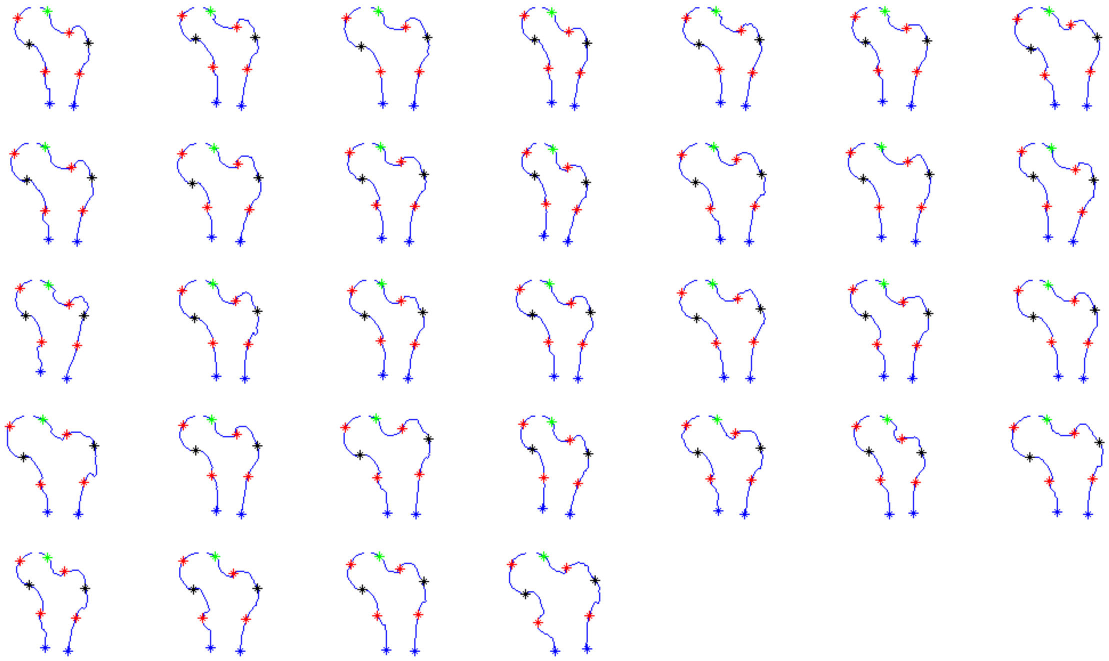


Figure 6.20 Results after optimization by MDL. It can be seen that four levels of nodes are placed along the boundary curves with the blue colour representing the fixed points. The first one is the master example that has been pre-processed by an expert.

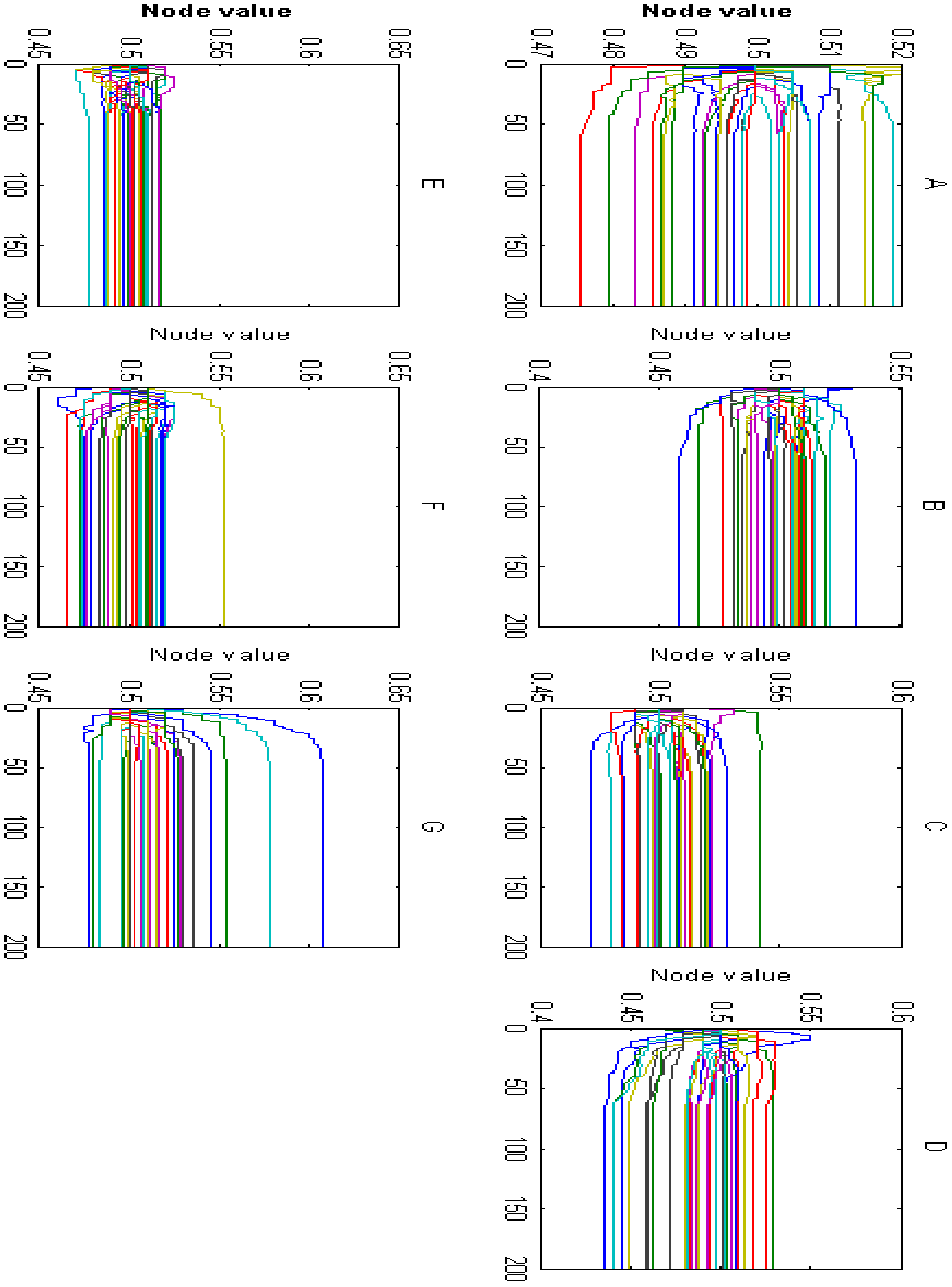


Figure 6.21 A demonstration of node movement during optimization. Seven out of nine nodes are shown here. Each graph gives the movement of the node in 32 examples represented by different colour. X coordinate is step number and Y coordinate is node value. All nodes start to move from parameter 0.5 and stabilize around the 50th step.

As previously, the outputs from MEM cost-function are presented in the Figure 6.22. During experiments, we can observe that our proposed MEM approach can converge approximately 2.2 times faster than the MDL approach.

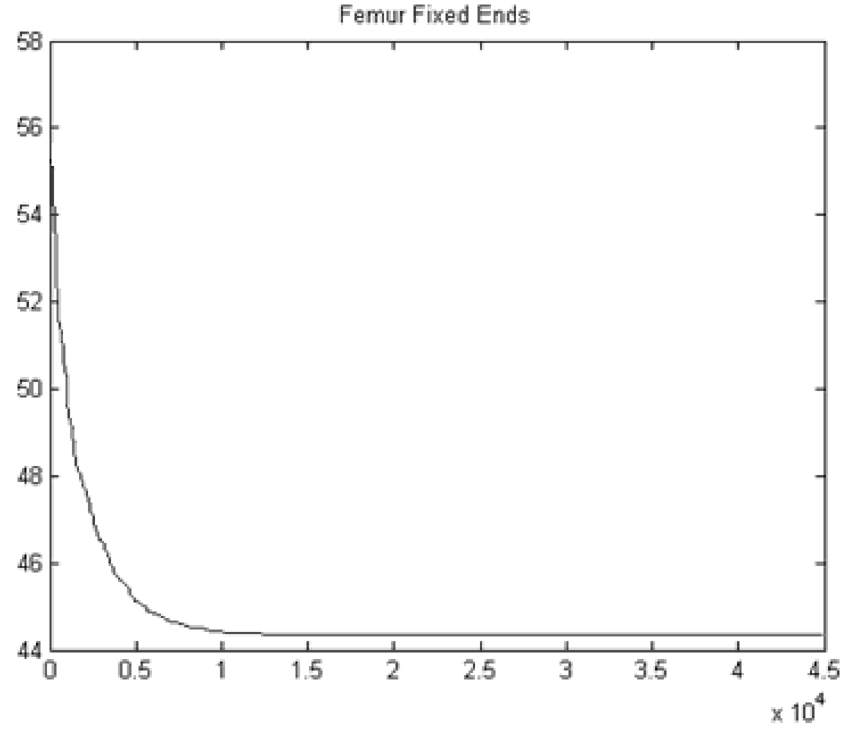


Figure 6.22 A demonstration of output of cost-function. X coordinate represents steps in time and Y coordinate represents value of the cost function. The blue solid line shows the output of MEM cost-function by using our proposed MEM approach.

After applying MEM to datasets, all corresponding points are allocated. At the same time, an ASM will be ready for performing further tests. An example of the effects of first three principal components are shown below in Figure 6.23.

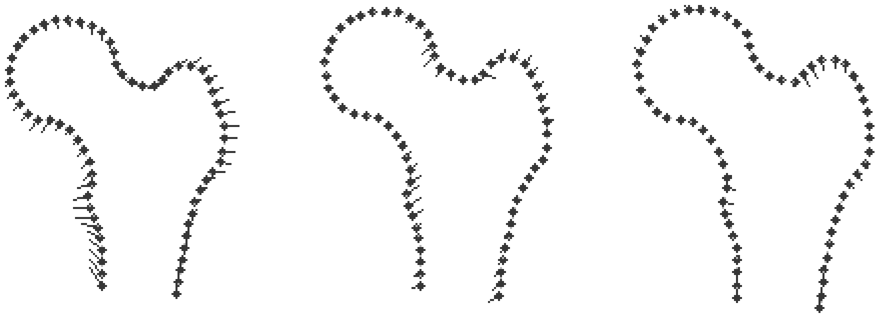


Figure 6.23 Shown is the mean shape with red marks, the whiskers emanating from the marks indicate three standard deviations of the first three principal components.

Figure 6.24, shows the Generalization Ability comparison on open curve with fixed ends, which provides $G(MEM) < G(MDL)$ in all M but the first one, so we can conclude that MEM achieves better performance in Generalization Ability test.

In Figure 6.25, it can be seen that MEM only has slightly better performance in most of M (number of shape modes). Compared with results from the experiments performed on closed curves, MEM's advantage over MDL has been reduced due to manual interference in the form of fixed end points. Still we can conclude that MEM is more specific than MDL when applying to open curve with fixed ends.

Similar to what happened in experiments with closed curves MEM achieved similar Compactness level as MDL did, results are presented in Figure 6.26.

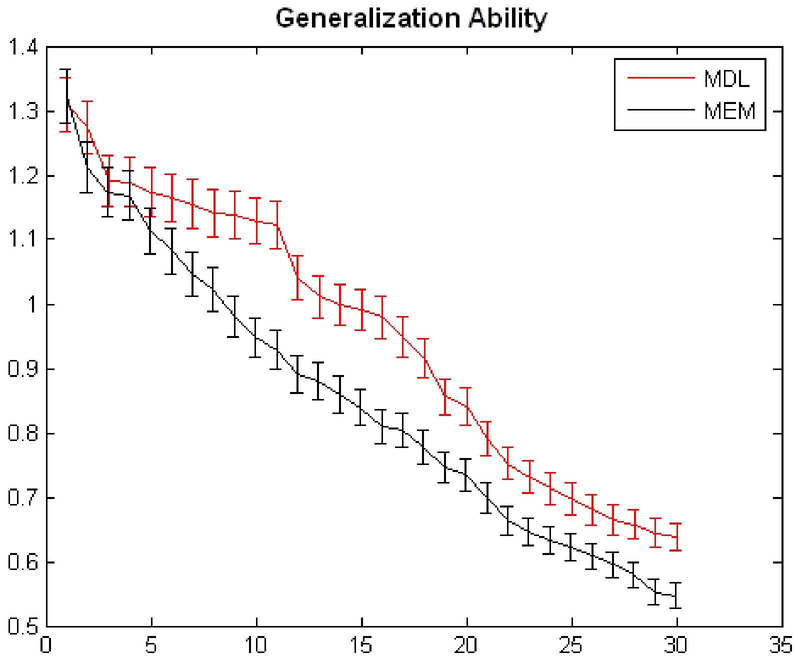


Figure 6.24 Generalization Ability comparisons on open curve with fixed ends. X-axis represents number of modes and Y-axis represents value of Generalization Ability.

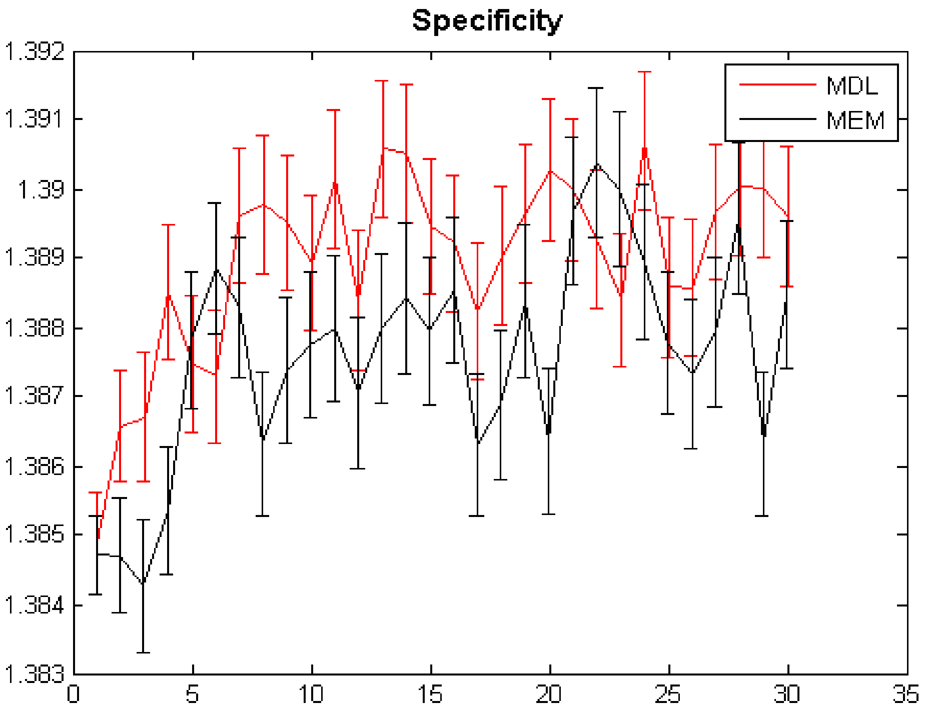


Figure 6.25 Specificity comparisons on open curve with fixed ends. X-axis represents number of modes and Y-axis represents value of Specificity.

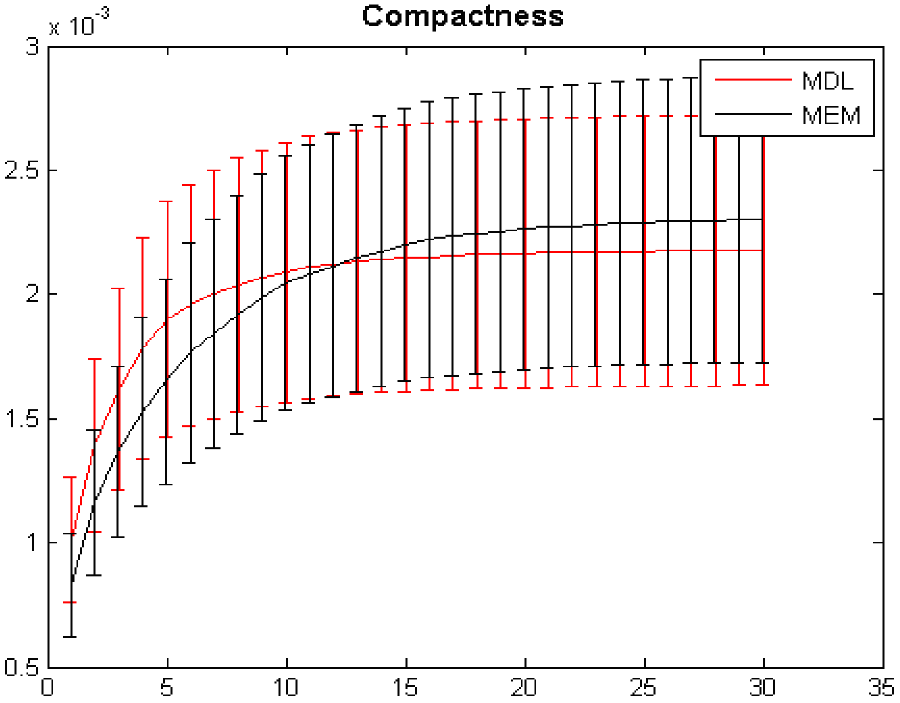
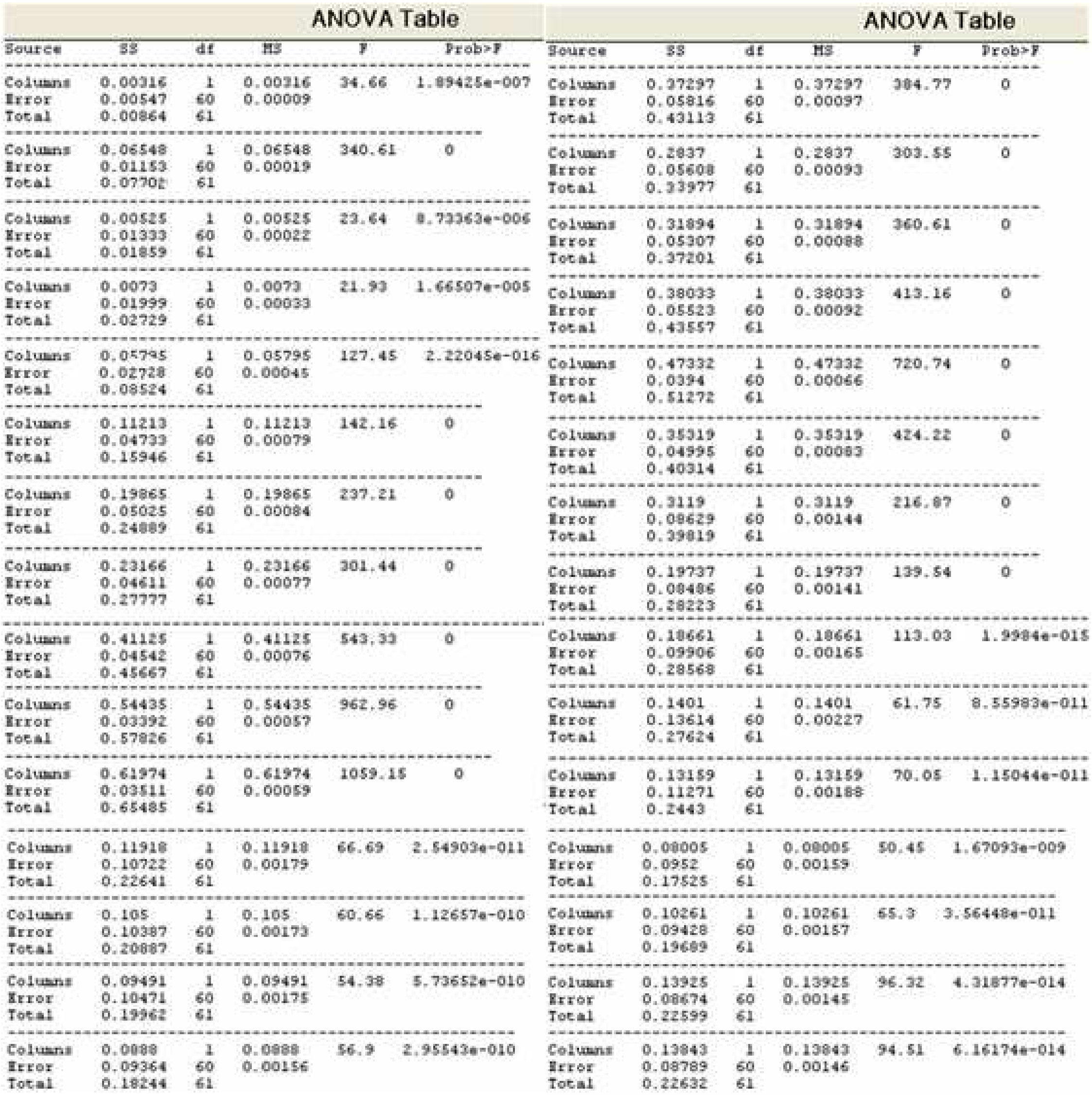


Figure 6.26 Compactness comparisons on open curve with fixed ends. X-axis represents number of modes and Y-axis represents value of Compactness.

		MEM	MDL	Percentage Difference
Generalization Ability	Mean – Standard Deviation	**19.6739668**	21.34493112	8.147290193%
	Mean	**20.36815613**	22.84819852	11.47733264%
	Mean + Standard Deviation	**21.06234546**	23.59616743	11.34754297%
Specificity	Mean – Standard Deviation	**29.14427775**	30.50049923	4.54766216%
	Mean	**30.52226656**	30.55334555	0.101772171%
	Mean + Standard Deviation	**30.5440339**	30.57412511	0.098468955%
Compactness	Mean – Standard Deviation	32.52903414	**31.6409396**	2.767944232%
	Mean	44.11569886	**43.18801673**	2.125183621%
	Mean + Standard Deviation	54.97383523	**53.84699932**	2.070992959%

Table 6.4. A quantitative analysis on the three criteria comparisons based on Area Under The Curve. The smaller corresponding value is marked in bold character.

We estimated the Area Under the Curve (AUC) to quantitatively present the differences between the three criteria. The AUC value is calculated for each criteria of mean value, mean value minus standard deviation and mean value plus standard deviation. The results of AUC are presented in Table 6.4, where smaller values are made in bold characters. It can be seen that for Generalization Ability, MEM is better than MDL from 8.1% to 11.5%, for Specificity, MEM is better than MDL from 0.1% to 4.5% and MEM is worse than MDL from 2% to 2.7% in Compactness.

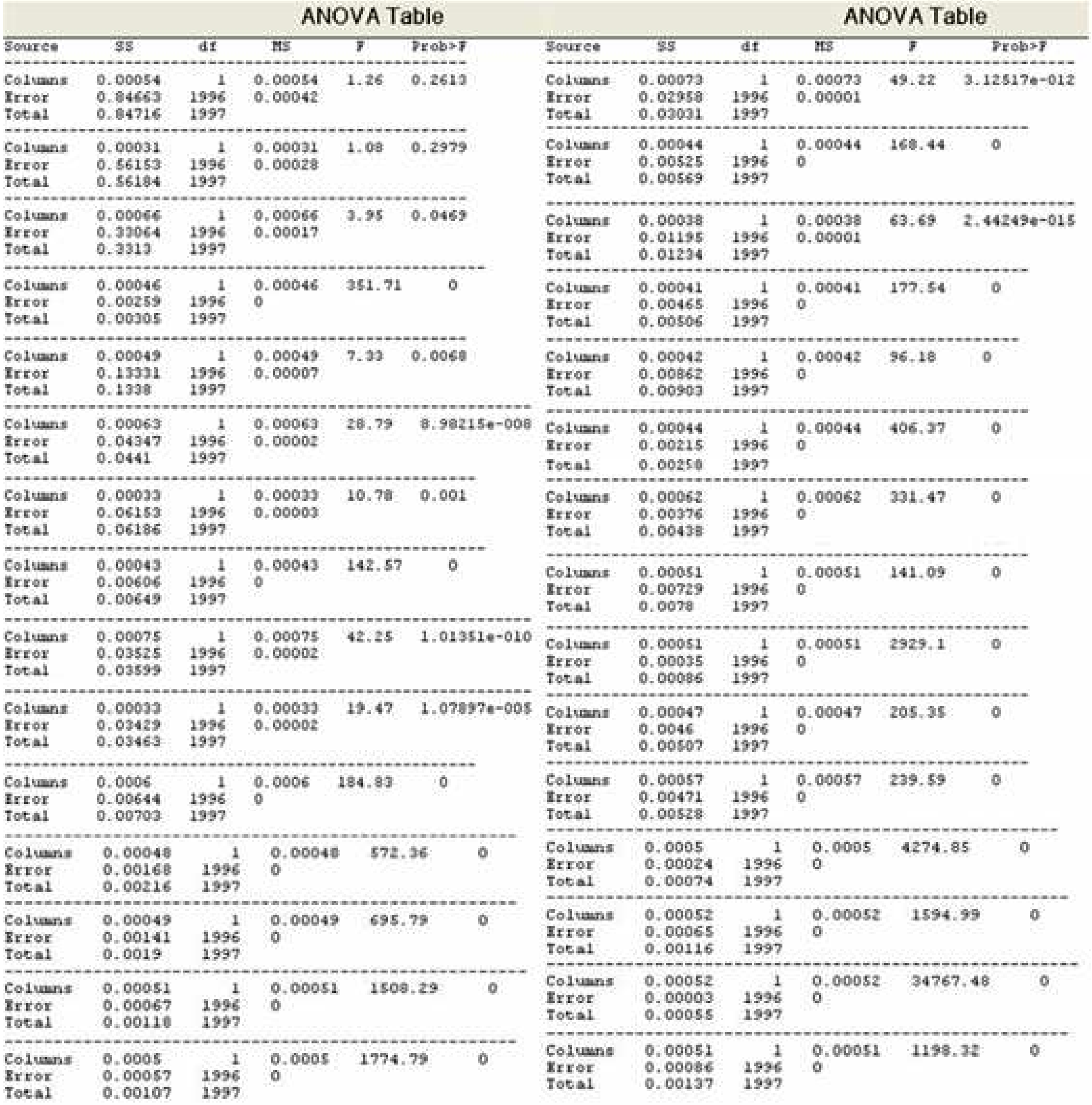


```
                          ANOVA Table
Source        SS       df      MS          F         Prob>F
---------------------------------------------------------------
Columns    0.00316      1    0.00316     34.66    1.89425e-007
Error      0.00547     60    0.00009
Total      0.00864     61
-----------------------------------------------------------
Columns    0.06548      1    0.06548    340.61         0
Error      0.01153     60    0.00019
Total      0.07702     61
---------------------------------------------------------------
Columns    0.00525      1    0.00525     23.64    8.73363e-006
Error      0.01333     60    0.00022
Total      0.01859     61
---------------------------------------------------------------
Columns    0.0073       1    0.0073      21.93    1.66507e-005
Error      0.01999     60    0.00033
Total      0.02729     61
----------------------------------------------------------------
Columns    0.05795      1    0.05795    127.45    2.22045e-016
Error      0.02728     60    0.00045
Total      0.08524     61
-----------------------------------------------------------
Columns    0.11213      1    0.11213    142.16         0
Error      0.04733     60    0.00079
Total      0.15946     61
-----------------------------------------------------------
Columns    0.19865      1    0.19865    237.21         0
Error      0.05025     60    0.00084
Total      0.24889     61
-----------------------------------------------------------
Columns    0.23166      1    0.23166    301.44         0
Error      0.04611     60    0.00077
Total      0.27777     61
----------------------------------------------------------------
Columns    0.41125      1    0.41125    543.33         0
Error      0.04542     60    0.00076
Total      0.45667     61
-----------------------------------------------------------
Columns    0.54435      1    0.54435    962.96         0
Error      0.03392     60    0.00057
Total      0.57826     61
------------------------------------------------------------
Columns    0.61974      1    0.61974   1059.15          0
Error      0.03511     60    0.00059
Total      0.65485     61
---------------------------------------------------------------
Columns    0.11918      1    0.11918     66.69    2.54903e-011
Error      0.10722     60    0.00179
Total      0.22641     61
---------------------------------------------------------------
Columns    0.105        1    0.105       60.66    1.12657e-010
Error      0.10387     60    0.00173
Total      0.20887     61
---------------------------------------------------------------
Columns    0.09491      1    0.09491     54.38    5.73652e-010
Error      0.10471     60    0.00175
Total      0.19962     61
--------------------------------------------------------------
Columns    0.0888       1    0.0888      56.9     2.95543e-010
Error      0.09364     60    0.00156
Total      0.18244     61
```

```
                          ANOVA Table
Source        SS       df      MS          F         Prob>F
-----------------------------------------------------------
Columns    0.37297      1    0.37297    384.77         0
Error      0.05816     60    0.00097
Total      0.43113     61
-----------------------------------------------------------
Columns    0.2837       1    0.2837     303.55         0
Error      0.05608     60    0.00093
Total      0.33977     61
-----------------------------------------------------------
Columns    0.31894      1    0.31894    360.61         0
Error      0.05307     60    0.00088
Total      0.37201     61
-----------------------------------------------------------
Columns    0.38033      1    0.38033    413.16         0
Error      0.05523     60    0.00092
Total      0.43557     61
-----------------------------------------------------------
Columns    0.47332      1    0.47332    720.74         0
Error      0.0394      60    0.00066
Total      0.51272     61
-----------------------------------------------------------
Columns    0.35319      1    0.35319    424.22         0
Error      0.04995     60    0.00083
Total      0.40314     61
-----------------------------------------------------------
Columns    0.3119       1    0.3119     216.87         0
Error      0.08629     60    0.00144
Total      0.39819     61
-----------------------------------------------------------
Columns    0.19737      1    0.19737    139.54         0
Error      0.08486     60    0.00141
Total      0.28223     61
---------------------------------------------------------------
Columns    0.18661      1    0.18661    113.03    1.9984e-015
Error      0.09906     60    0.00165
Total      0.28568     61
---------------------------------------------------------------
Columns    0.1401       1    0.1401      61.75    8.55983e-011
Error      0.13614     60    0.00227
Total      0.27624     61
---------------------------------------------------------------
Columns    0.13159      1    0.13159     70.05    1.15044e-011
Error      0.11271     60    0.00188
Total      0.2443      61
---------------------------------------------------------------
Columns    0.08005      1    0.08005     50.45    1.67093e-009
Error      0.0952      60    0.00159
Total      0.17525     61
-------------------------------------------------------------
Columns    0.10261      1    0.10261     65.3     3.56448e-011
Error      0.09428     60    0.00157
Total      0.19689     61
---------------------------------------------------------------
Columns    0.13925      1    0.13925     96.32    4.31877e-014
Error      0.08674     60    0.00145
Total      0.22599     61
---------------------------------------------------------------
Columns    0.13843      1    0.13843     94.51    6.16174e-014
Error      0.08789     60    0.00146
Total      0.22632     61
```

Table 6.5 ANOVA table of the Generalization Ability on datasets of open curves with fixed ends.

From Tables 6.5 and 6.6, it can be seen that most of the *p*-value is zero or close to zero therefore the ANOVA results suggests that the two datasets are statistically different for all the cases in Generalizaion ability and different for most of the cases in Specificitiy.



ANOVA Table

```
Source       SS        df       MS          F       Prob>F
-----------------------------------------------------------
Columns    0.00054       1    0.00054     1.26     0.2613
Error      0.84663    1996    0.00042
Total      0.84716    1997
-----------------------------------------------------------
Columns    0.00031       1    0.00031     1.08     0.2979
Error      0.56153    1996    0.00028
Total      0.56184    1997
-----------------------------------------------------------
Columns    0.00066       1    0.00066     3.95     0.0469
Error      0.33064    1996    0.00017
Total      0.3313     1997
-------------------------------------------------------------
Columns    0.00046       1    0.00046   351.71        0
Error      0.00259    1996    0
Total      0.00305    1997
-----------------------------------------------------------
Columns    0.00049       1    0.00049     7.33     0.0068
Error      0.13331    1996    0.00007
Total      0.1338     1997
-------------------------------------------------------------------
Columns    0.00063       1    0.00063    28.79    8.98215e-008
Error      0.04347    1996    0.00002
Total      0.0441     1997
------------------------------------------------------------
Columns    0.00033       1    0.00033    10.78    0.001
Error      0.06153    1996    0.00003
Total      0.06186    1997
-------------------------------------------------------------
Columns    0.00043       1    0.00043   142.57        0
Error      0.00606    1996    0
Total      0.00649    1997
-------------------------------------------------------------------
Columns    0.00075       1    0.00075    42.25    1.01351e-010
Error      0.03525    1996    0.00002
Total      0.03599    1997
-------------------------------------------------------------------
Columns    0.00033       1    0.00033    19.47    1.07897e-005
Error      0.03429    1996    0.00002
Total      0.03463    1997
------------------------------------------------------------
Columns    0.0006        1    0.0006    184.83        0
Error      0.00644    1996    0
Total      0.00703    1997
-----------------------------------------------------------------
Columns     0.00048        1     0.00048     572.36       0
Error       0.00168     1996     0
Total       0.00216     1997
-----------------------------------------------------------------
Columns     0.00049        1     0.00049     695.79       0
Error       0.00141     1996     0
Total       0.0019      1997
-------------------------------------------------------------------
Columns     0.00051        1     0.00051    1508.29       0
Error       0.00067     1996     0
Total       0.00118     1997
-----------------------------------------------------------------
Columns     0.0005         1     0.0005     1774.79       0
Error       0.00057     1996     0
Total       0.00107     1997
```

ANOVA Table

```
Source       SS        df       MS          F         Prob>F
-------------------------------------------------------------------
Columns    0.00073       1    0.00073    49.22    3.12517e-012
Error      0.02958    1996    0.00001
Total      0.03031    1997
-------------------------------------------------------------
Columns    0.00044       1    0.00044   168.44        0
Error      0.00525    1996    0
Total      0.00569    1997

-------------------------------------------------------------------
Columns    0.00038       1    0.00038    63.69    2.44249e-015
Error      0.01195    1996    0.00001
Total      0.01234    1997
-------------------------------------------------------------
Columns    0.00041       1    0.00041   177.54        0
Error      0.00465    1996    0
Total      0.00506    1997
------------------------------------------------------------
Columns    0.00042       1    0.00042    96.18       0
Error      0.00862    1996    0
Total      0.00903    1997
-------------------------------------------------------------
Columns    0.00044       1    0.00044   406.37        0
Error      0.00215    1996    0
Total      0.00258    1997
-------------------------------------------------------------
Columns    0.00062       1    0.00062   331.47        0
Error      0.00376    1996    0
Total      0.00438    1997

-------------------------------------------------------------
Columns    0.00051       1    0.00051   141.09        0
Error      0.00729    1996    0
Total      0.0078     1997
-------------------------------------------------------------
Columns    0.00051       1    0.00051  2929.1         0
Error      0.00035    1996    0
Total      0.00086    1997
-------------------------------------------------------------
Columns    0.00047       1    0.00047   205.35        0
Error      0.0046     1996    0
Total      0.00507    1997
-------------------------------------------------------------
Columns    0.00057       1    0.00057   239.59        0
Error      0.00471    1996    0
Total      0.00528    1997
-----------------------------------------------------------------
Columns     0.0005         1     0.0005     4274.85       0
Error       0.00024     1996     0
Total       0.00074     1997
------------------------------------------------------------------
Columns     0.00052        1     0.00052    1594.99       0
Error       0.00065     1996     0
Total       0.00116     1997
-------------------------------------------------------------------
Columns     0.00052        1     0.00052   34767.48       0
Error       0.00003     1996     0
Total       0.00055     1997
------------------------------------------------------------------
Columns     0.00051        1     0.00051    1198.32       0
Error       0.00086     1996     0
Total       0.00137     1997
```

Table 6.6 ANOVA table of the Specificity on datasets of open curves with fixed ends.

6.3 Experiments on Open Curves with Free Ends

Furthermore, experiments are performed on the same datasets of 32 contours of femurs, but they will have free end points during optimization. It means every point on each shape is able to move freely following the decline direction of the cost function. Again, MDL with one single master example and MEM without master example were tested.

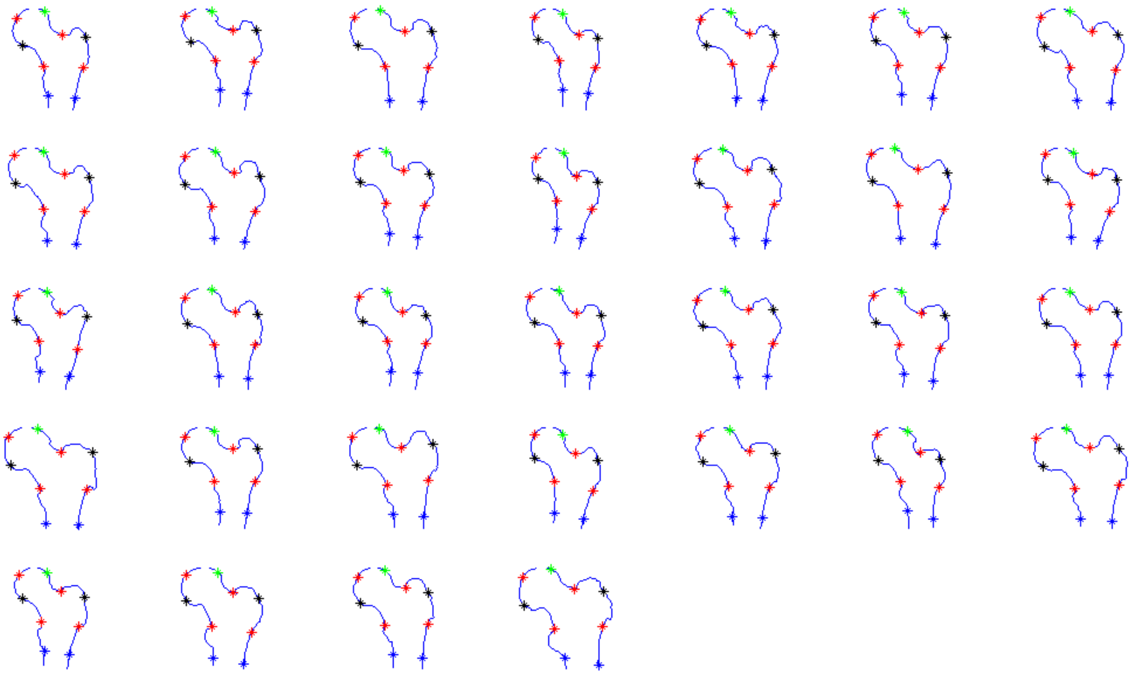


Figure 6.27 Results after applying MEM on open curve with free ends. Annotation is the same as the previous experiment.

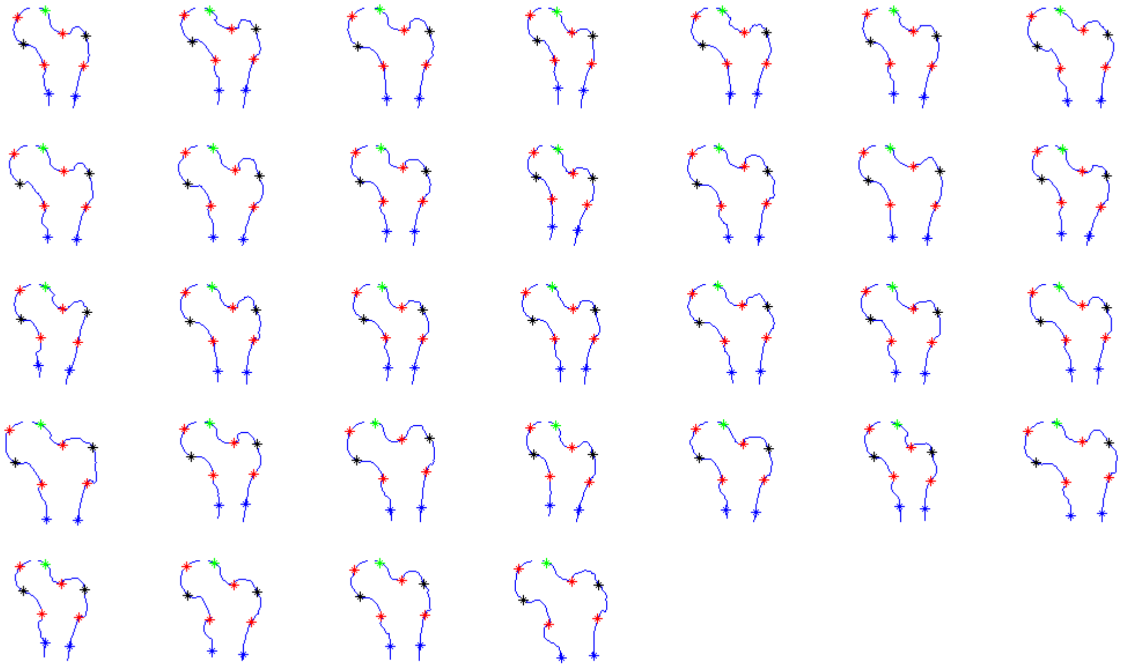


Figure 6.28 Results after applying MDL on open curve with free ends.

As can be seen from Figure 6.27, the blue level points are not fixed to the end of the curves. Compared with results of open curves with fixed end, the correspondence results are different. Figure 6.28 shows the correspondence results found by MDL algorithm.

Figure 6.30 shows the output of cost-function of MEM during optimization. From the figure, we can observe that the cost-function achieve stabilization after a few steps of optimization. During experiments, we can observe that our MEM converge about 2.2 times faster than the MDL approach. Comparison between Figure 6.22 and Figure 6.30 shows that MEM achieves stabilization faster if the ends are free. The movement of each node during optimization and three standard deviations of the first three principal components are also shown in Figures 6.29, and 6.31, respectively.

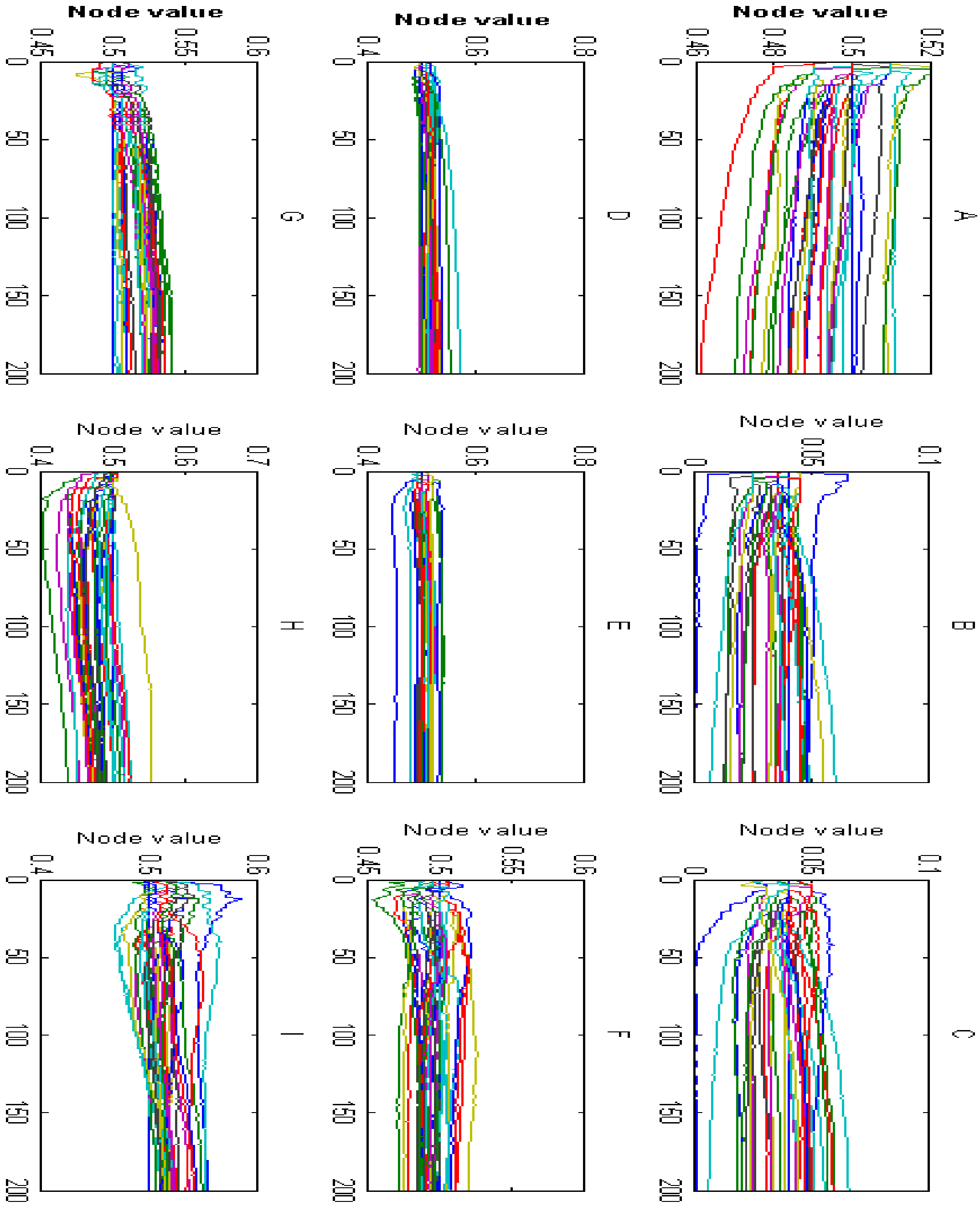


Figure 6.29 This figure shows nine nodes' movement during optimization. Each graph gives the movement of the node in 32 examples represented by different colour. X coordinate is step number and Y coordinate is node value. All nodes start to move from parameter 0.5 and stabilize around the 25th step.

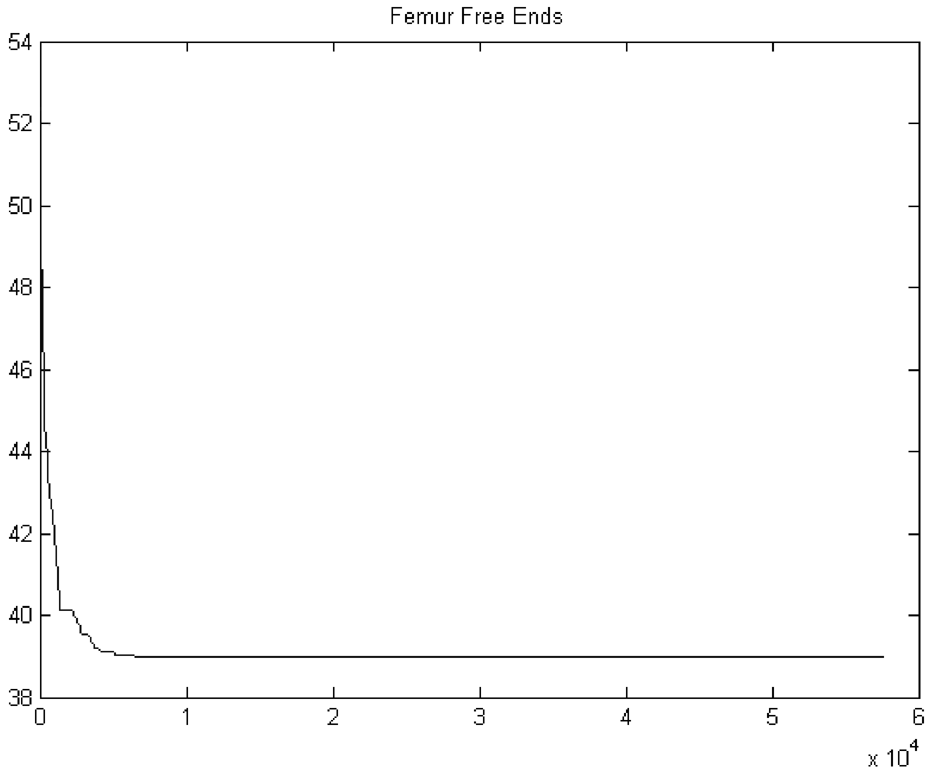


Figure 6.30 Cost function performance during optimization with x-axis representing steps in time, y-axis representing output of the cost function. The blue solid line shows the performance by our proposed MEM approach.

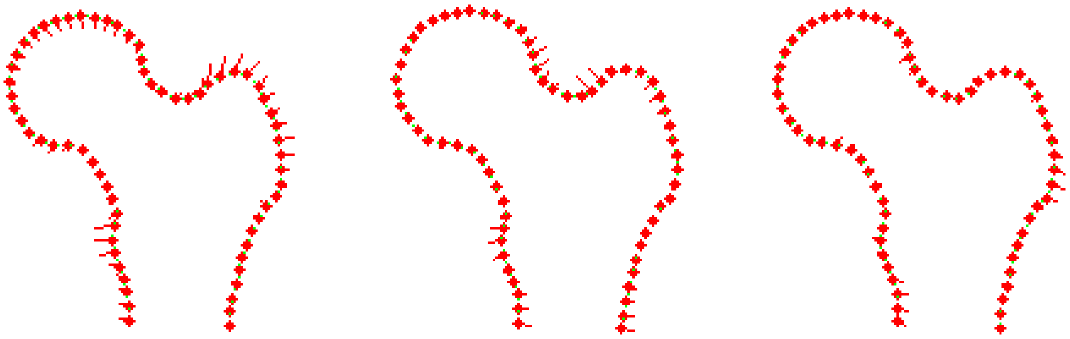


Figure 6.31 Shows the mean shape with red marks, the whiskers emanating from the marks indicate three standard deviations of the first three principal components.

Details of the MEM performance versus MDL are shown in Figures 6.32, 6.33 and 6.34. The conclusion is that MEM has better Generalization Ability, Specificity and similar Compactness compared with MDL. We also noticed that compared between experiments on open curve with free ends and experiments on curve with fixed ends, the former one performs better in more modes and the difference in each mode is larger considering the error bar.

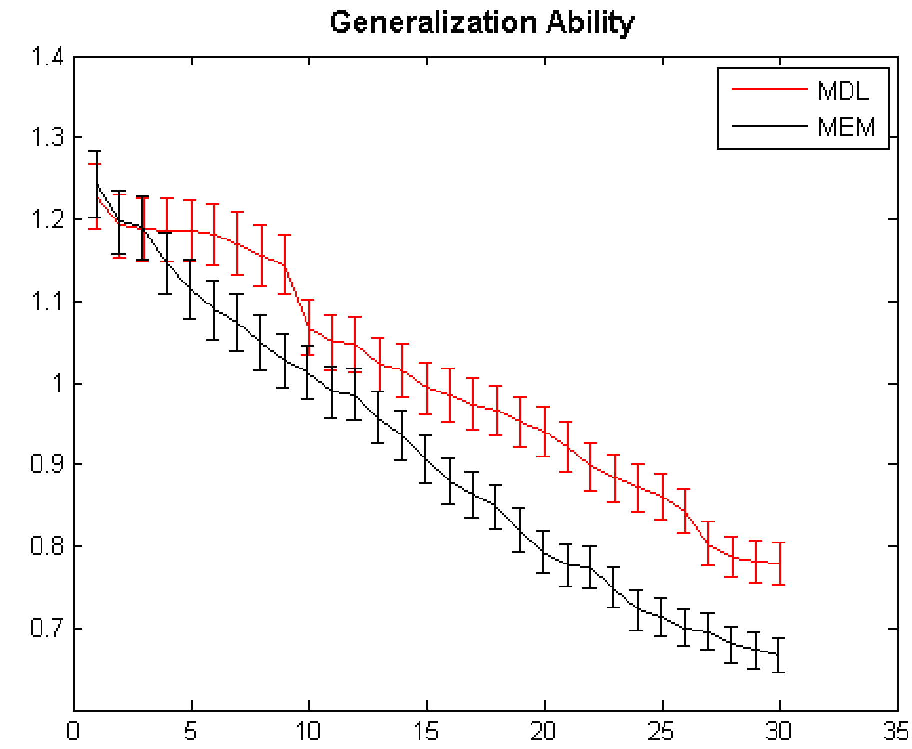


Figure 6.32 Generalization Ability comparison on open curve with free ends. X-axis represents number of modes used in optimization Y-axis represents Generalization Ability.

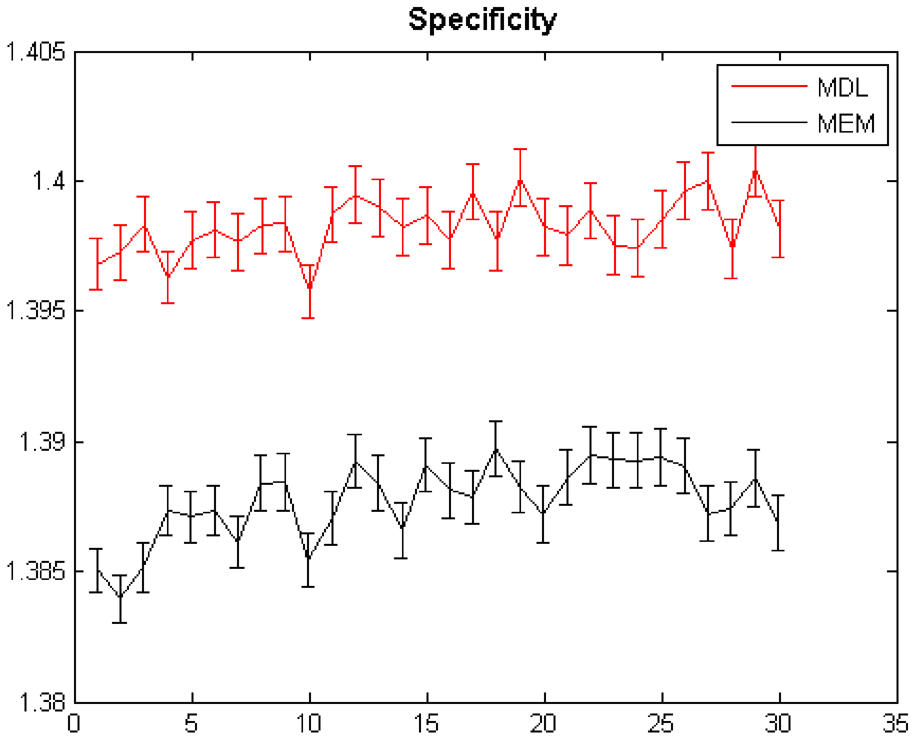


Figure 6.33 Specificity comparison on open curve with free ends. X-axis represents number of modes used in optimization Y-axis represents Specificity.

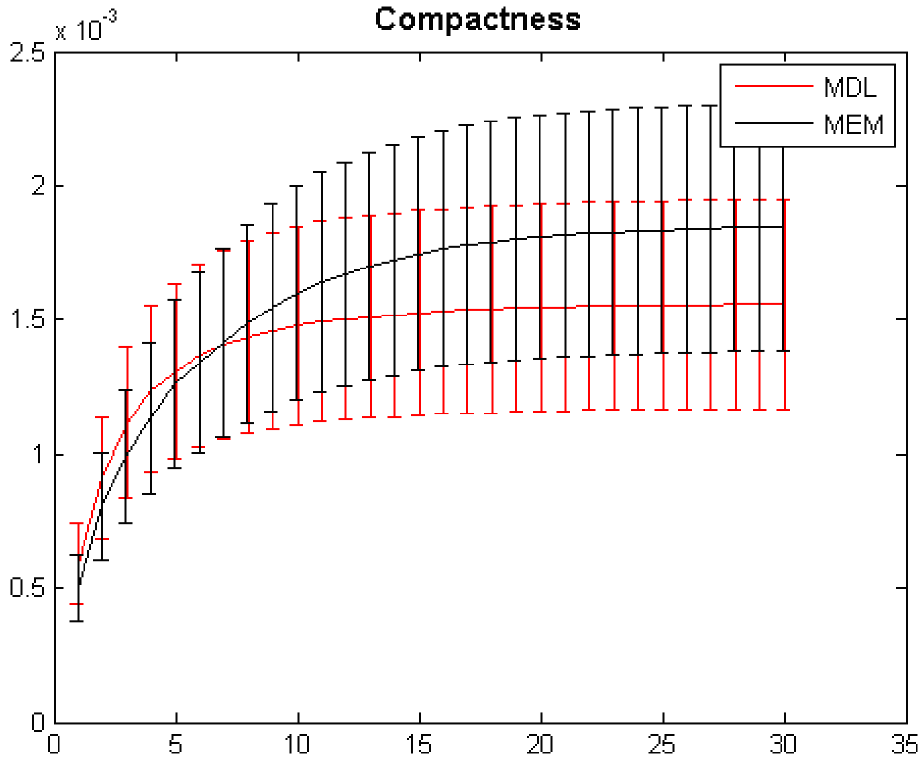


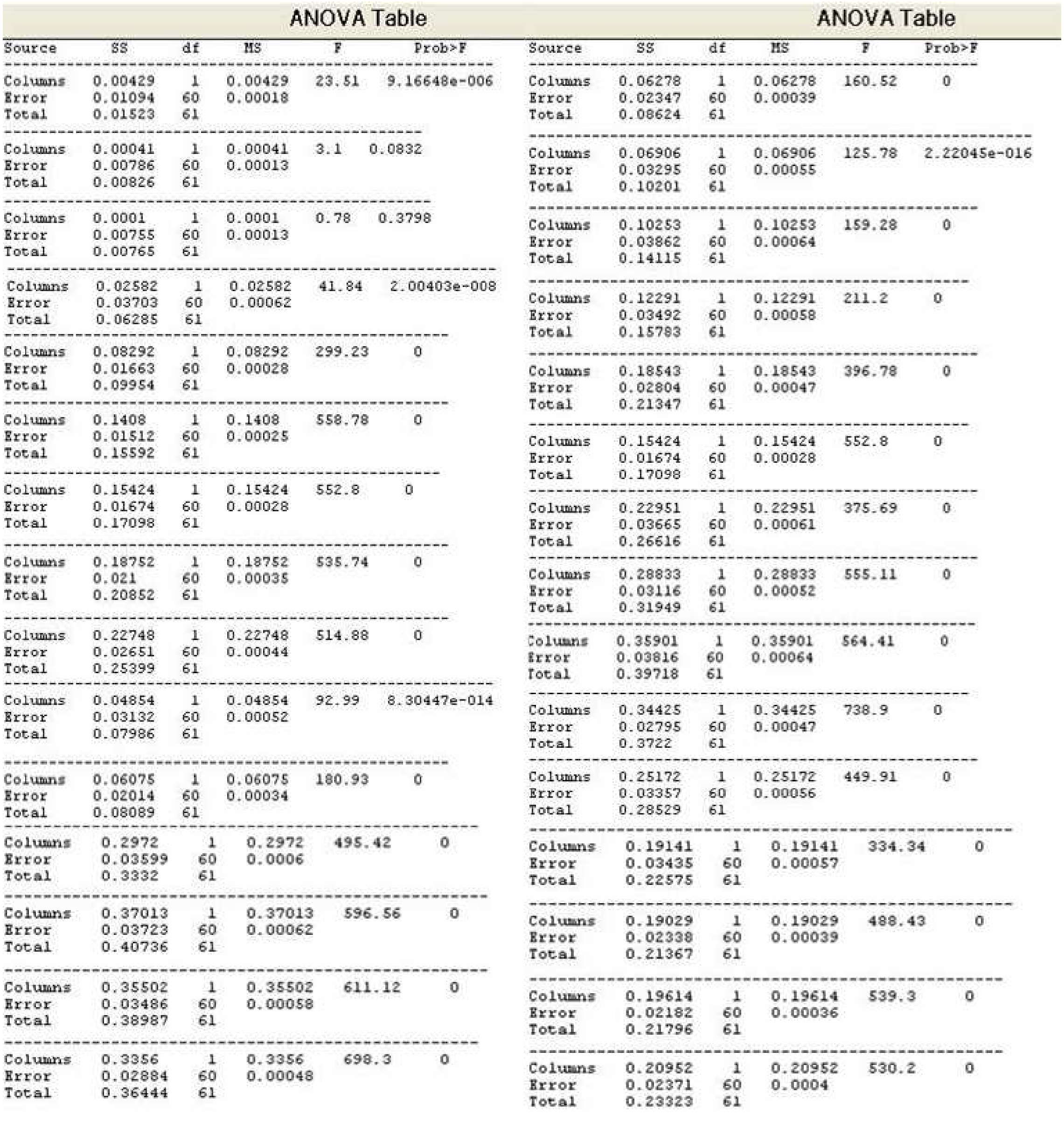
Figure 6.34 Compactness comparisons on open curve with free ends. X-axis represents number of modes used in optimization Y-axis represents Compactness.

		MEM	MDL	Percentage Difference
Generalization Ability	Mean – Standard Deviation	**20.8549217**	21.84049217	4.616750983%
	Mean	**21.56465324**	23.4852349	8.526465812%
	Mean + Standard Deviation	**22.27438479**	24.24138702	8.45735612%
Specificity	Mean – Standard Deviation	**29.34152043**	30.5049467	3.888036601%
	Mean	**30.52784939**	30.76233419	0.765162664%
	Mean + Standard Deviation	**30.52784939**	30.55075207	0.767901327%
Compactness	Mean – Standard Deviation	25.75947493	**22.52097315**	13.41537581%
	Mean	34.12603632	**31.44714765**	8.170683524%
	Mean + Standard Deviation	42.49259771	**39.1970983**	8.068335589%

Table 6.7. A quantitative analysis on the three criteria comparisons based on Area Under the Curve. The smaller corresponding value is marked in bold character.

We perform the Area Under the Curve (AUC) to quantitatively calculate the difference between MEM and MDL using the three criteria. The AUC value is calculated for each criteria of mean value, mean value minus standard deviation and mean value plus standard deviation. The results of AUC are presented in Table 6.7, where smaller values are made in bold characters. It can be seen that for Generalization Ability, MEM is better than MDL from 4.6% to 8.5%, for Specificity, MEM is better than MDL from 0.8% to 3.9% and MEM is worse than MDL from 8% to 13.4% in terms of Compactness.

The ANOVA tests in Tables 6.8 and 6.9 shows that most *p*-value are zero or close to zero, therefore it suggests that the MDL and MEM are different in all the cases for Specificity test and are different for most of the cases for Generalization Ability except two.



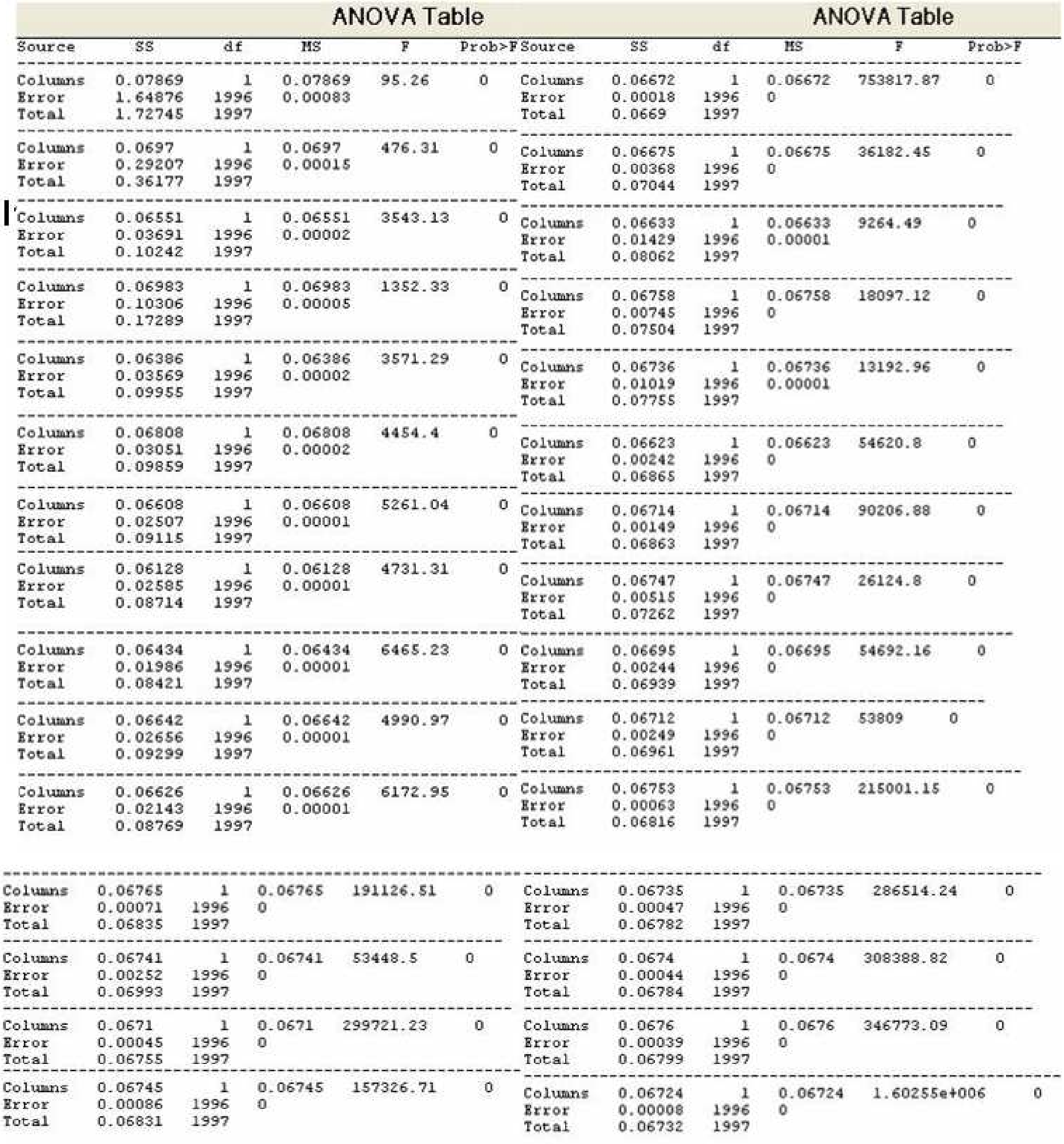
ANOVA Table

```
Source      SS       df     MS         F        Prob>F
--------------------------------------------------------
Columns    0.00429    1    0.00429   23.51    9.16648e-006
Error      0.01094   60    0.00018
Total      0.01523   61
---------------------------------------------------
Columns    0.00041    1    0.00041   3.1    0.0832
Error      0.00786   60    0.00013
Total      0.00826   61
----------------------------------------------------
Columns    0.0001     1    0.0001    0.78   0.3798
Error      0.00755   60    0.00013
Total      0.00765   61
--------------------------------------------------------
 Columns    0.02582    1    0.02582   41.84    2.00403e-008
 Error      0.03703   60    0.00062
 Total      0.06285   61
-----------------------------------------------------
Columns    0.08292    1    0.08292   299.23      0
Error      0.01663   60    0.00028
Total      0.09954   61
-----------------------------------------------------
Columns    0.1408     1    0.1408    558.78      0
Error      0.01512   60    0.00025
Total      0.15592   61
----------------------------------------------------
Columns    0.15424    1    0.15424   552.8      0
Error      0.01674   60    0.00028
Total      0.17098   61

-----------------------------------------------------
Columns    0.18752    1    0.18752   535.74      0
Error      0.021     60    0.00035
Total      0.20852   61

-----------------------------------------------------
Columns    0.22748    1    0.22748   514.88      0
Error      0.02651   60    0.00044
Total      0.25399   61
--------------------------------------------------------
Columns    0.04854    1    0.04854   92.99    8.30447e-014
Error      0.03132   60    0.00052
Total      0.07986   61

-----------------------------------------------------
Columns    0.06075    1    0.06075   180.93      0
Error      0.02014   60    0.00034
Total      0.08089   61
-------------------------------------------------------
Columns     0.2972      1    0.2972    495.42       0
Error       0.03599    60    0.0006
Total       0.3332     61
--------------------------------------------------------
Columns     0.37013     1    0.37013    596.56      0
Error       0.03723    60    0.00062
Total       0.40736    61

--------------------------------------------------------
Columns     0.35502     1    0.35502    611.12      0
Error       0.03486    60    0.00058
Total       0.38987    61
-------------------------------------------------------
Columns     0.3356      1    0.3356     698.3       0
Error       0.02884    60    0.00048
Total       0.36444    61
```

ANOVA Table

```
Source      SS       df     MS         F       Prob>F
----------------------------------------------------
Columns    0.06278    1    0.06278   160.52     0
Error      0.02347   60    0.00039
Total      0.08624   61

-----------------------------------------------------------
Columns    0.06906    1    0.06906   125.78   2.22045e-016
Error      0.03295   60    0.00055
Total      0.10201   61

----------------------------------------------------
Columns    0.10253    1    0.10253   159.28     0
Error      0.03862   60    0.00064
Total      0.14115   61

---------------------------------------------------
Columns    0.12291    1    0.12291   211.2     0
Error      0.03492   60    0.00058
Total      0.15783   61

----------------------------------------------------
Columns    0.18543    1    0.18543   396.78     0
Error      0.02804   60    0.00047
Total      0.21347   61

---------------------------------------------------
Columns    0.15424    1    0.15424   552.8     0
Error      0.01674   60    0.00028
Total      0.17098   61
----------------------------------------------------
Columns    0.22951    1    0.22951   375.69     0
Error      0.03665   60    0.00061
Total      0.26616   61
----------------------------------------------------
Columns    0.28833    1    0.28833   555.11     0
Error      0.03116   60    0.00052
Total      0.31949   61
----------------------------------------------------
Columns   0.35901    1   0.35901   564.41     0
Error     0.03816   60   0.00064
Total     0.39718   61
---------------------------------------------------
Columns    0.34425    1    0.34425   738.9     0
Error      0.02795   60    0.00047
Total      0.3722    61
----------------------------------------------------
Columns    0.25172    1    0.25172   449.91     0
Error      0.03357   60    0.00056
Total      0.28529   61
--------------------------------------------------------
Columns     0.19141     1    0.19141    334.34      0
Error       0.03435    60    0.00057
Total       0.22575    61

--------------------------------------------------------
Columns     0.19029     1    0.19029    488.43      0
Error       0.02338    60    0.00039
Total       0.21367    61

-------------------------------------------------------
Columns     0.19614     1    0.19614    539.3      0
Error       0.02182    60    0.00036
Total       0.21796    61

-------------------------------------------------------
Columns     0.20952     1    0.20952    530.2      0
Error       0.02371    60    0.0004
Total       0.23323    61
```

Table 6.8 ANOVA table of the Generalization Ability on datasets of open curves with free ends.



```
                              ANOVA Table
Source        SS        df       MS          F      Prob>F
----------------------------------------------------------
Columns     0.07869        1   0.07869     95.26       0
Error       1.64876     1996   0.00083
Total       1.72745     1997
----------------------------------------------------------
Columns     0.0697         1   0.0697     476.31       0
Error       0.29207     1996   0.00015
Total       0.36177     1997
----------------------------------------------------------
Columns     0.06551        1   0.06551   3543.13       0
Error       0.03691     1996   0.00002
Total       0.10242     1997
----------------------------------------------------------
Columns     0.06983        1   0.06983   1352.33       0
Error       0.10306     1996   0.00005
Total       0.17289     1997
----------------------------------------------------------
Columns     0.06386        1   0.06386   3571.29       0
Error       0.03569     1996   0.00002
Total       0.09955     1997
----------------------------------------------------------
Columns     0.06808        1   0.06808   4454.4        0
Error       0.03051     1996   0.00002
Total       0.09859     1997
----------------------------------------------------------
Columns     0.06608        1   0.06608   5261.04       0
Error       0.02507     1996   0.00001
Total       0.09115     1997
----------------------------------------------------------
Columns     0.06128        1   0.06128   4731.31       0
Error       0.02585     1996   0.00001
Total       0.08714     1997
----------------------------------------------------------
Columns     0.06434        1   0.06434   6465.23       0
Error       0.01986     1996   0.00001
Total       0.08421     1997
----------------------------------------------------------
Columns     0.06642        1   0.06642   4990.97       0
Error       0.02656     1996   0.00001
Total       0.09299     1997
----------------------------------------------------------
Columns     0.06626        1   0.06626   6172.95       0
Error       0.02143     1996   0.00001
Total       0.08769     1997

----------------------------------------------------------
Columns     0.06765        1   0.06765   191126.51     0
Error       0.00071     1996   0
Total       0.06835     1997
----------------------------------------------------------
Columns     0.06741        1   0.06741   53448.5     0
Error       0.00252     1996   0
Total       0.06993     1997
----------------------------------------------------------
Columns     0.0671         1   0.0671   299721.23     0
Error       0.00045     1996   0
Total       0.06755     1997
----------------------------------------------------------
Columns     0.06745        1   0.06745   157326.71     0
Error       0.00086     1996   0
Total       0.06831     1997
```

```
                              ANOVA Table
Source        SS        df       MS          F          Prob>F
--------------------------------------------------------------
Columns     0.06672        1   0.06672   753817.87        0
Error       0.00018     1996   0
Total       0.0669      1997
--------------------------------------------------------------
Columns     0.06675        1   0.06675   36182.45        0
Error       0.00368     1996   0
Total       0.07044     1997
--------------------------------------------------------------
Columns     0.06633        1   0.06633   9264.49       0
Error       0.01429     1996   0.00001
Total       0.08062     1997
--------------------------------------------------------------
Columns     0.06758        1   0.06758   18097.12        0
Error       0.00745     1996   0
Total       0.07504     1997
--------------------------------------------------------------
Columns     0.06736        1   0.06736   13192.96        0
Error       0.01019     1996   0.00001
Total       0.07755     1997
--------------------------------------------------------------
Columns     0.06623        1   0.06623   54620.8       0
Error       0.00242     1996   0
Total       0.06865     1997
--------------------------------------------------------------
Columns     0.06714        1   0.06714   90206.88        0
Error       0.00149     1996   0
Total       0.06863     1997
--------------------------------------------------------------
Columns     0.06747        1   0.06747   26124.8       0
Error       0.00515     1996   0
Total       0.07262     1997
--------------------------------------------------------------
Columns     0.06695        1   0.06695   54692.16        0
Error       0.00244     1996   0
Total       0.06939     1997
--------------------------------------------------------------
Columns     0.06712        1   0.06712   53809      0
Error       0.00249     1996   0
Total       0.06961     1997
--------------------------------------------------------------
Columns     0.06753        1   0.06753   215001.15       0
Error       0.00063     1996   0
Total       0.06816     1997

--------------------------------------------------------------
Columns     0.06735        1   0.06735   286514.24       0
Error       0.00047     1996   0
Total       0.06782     1997
--------------------------------------------------------------
Columns     0.0674         1   0.0674    308388.82       0
Error       0.00044     1996   0
Total       0.06784     1997
--------------------------------------------------------------
Columns     0.0676         1   0.0676    346773.09       0
Error       0.00039     1996   0
Total       0.06799     1997
--------------------------------------------------------------
Columns     0.06724        1   0.06724   1.60255e+006       0
Error       0.00008     1996   0
Total       0.06732     1997
```

Table 6.9 ANOVA table of the Specificity on datasets of open curves with free ends.

6.4 Experiments on Improved Control of "Pile Up"

When applying the MDL technique, one may encounter the so-called "Pile Up" problem. The problem happens during optimization, when points can pile up into one location. In this case, the cost function will attain a global minimum or meaningless local minimum and fail in describing the rest of the shapes. As we know, the points we move in the experiments are control points, defined by the accuracy we need. There are more low levels in between existing control points. Therefore, it is unacceptable to have two or more control points overlapping each other. As have been stated in paragraph 4.7.2, different researchers have tried different methods to solve this problem. They either do not work efficient or lead the algorithm into an arbitrary manner.

In Figure 6.35, 22 datasets of silhouettes contours are shown. During the experiments on these datasets, we encountered the “Pile Up” problem. Figure 6.36 shows the results by applying MDL and Figure 6.37 shows the results by applying MEM respectively.

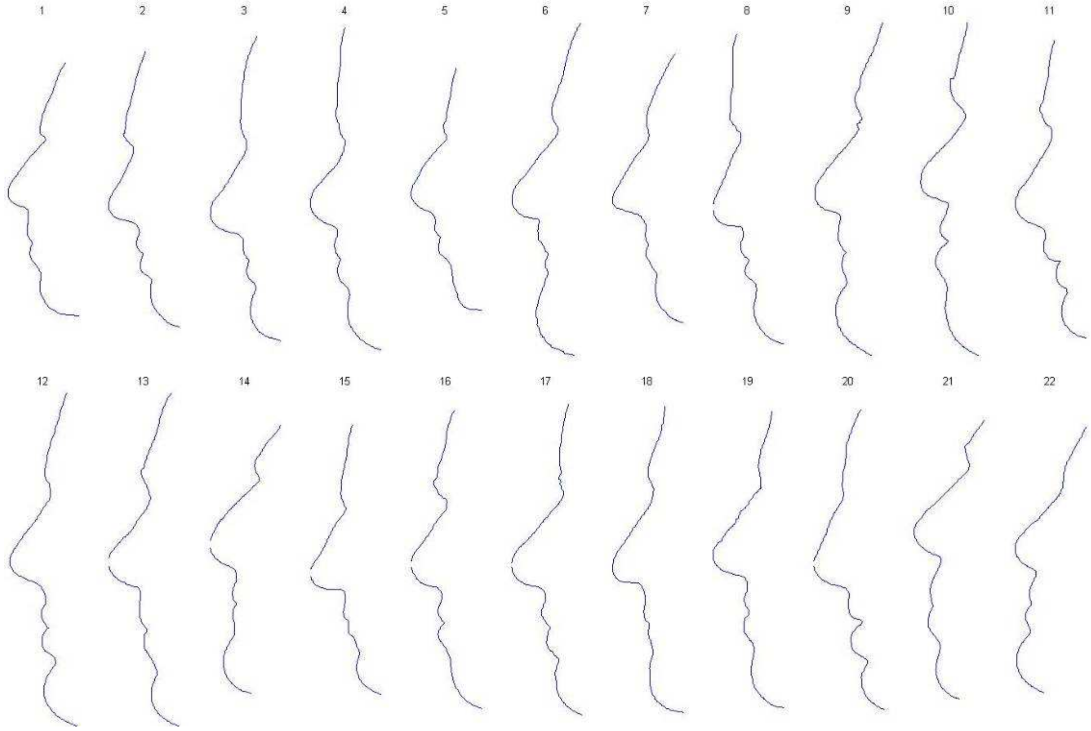


Figure 6.35 22 datasets of silhouettes contours are shown here

An example of the “Pile Up” effect is shown in Figure 6.36 when MDL was applied to datasets of silhouettes contours (open curves with free ends). Figure 6.36, shows the final converged results found by MDL algorithm. For making more clear the visual effect of the “Pile Up”, we put the bottom two points one pixel away from each other. In fact, the two bottom points are overlapping each other. It can be seen from the figure that the two points (a level one point and a level four point) at the bottom will collide or overlap. This “Pile Up” happened even when the first subject is used as a master example and this example has a well distributed control points on it. This “Pile Up” happened in MDL is because that the MDL will have lower cost function value if two points at the bottom are actually overlapping. In this case, however, since MEM has a larger cost-function value when points are overlapped and have equal probability, MEM approach prevents the “Pile Up” problem from happening.

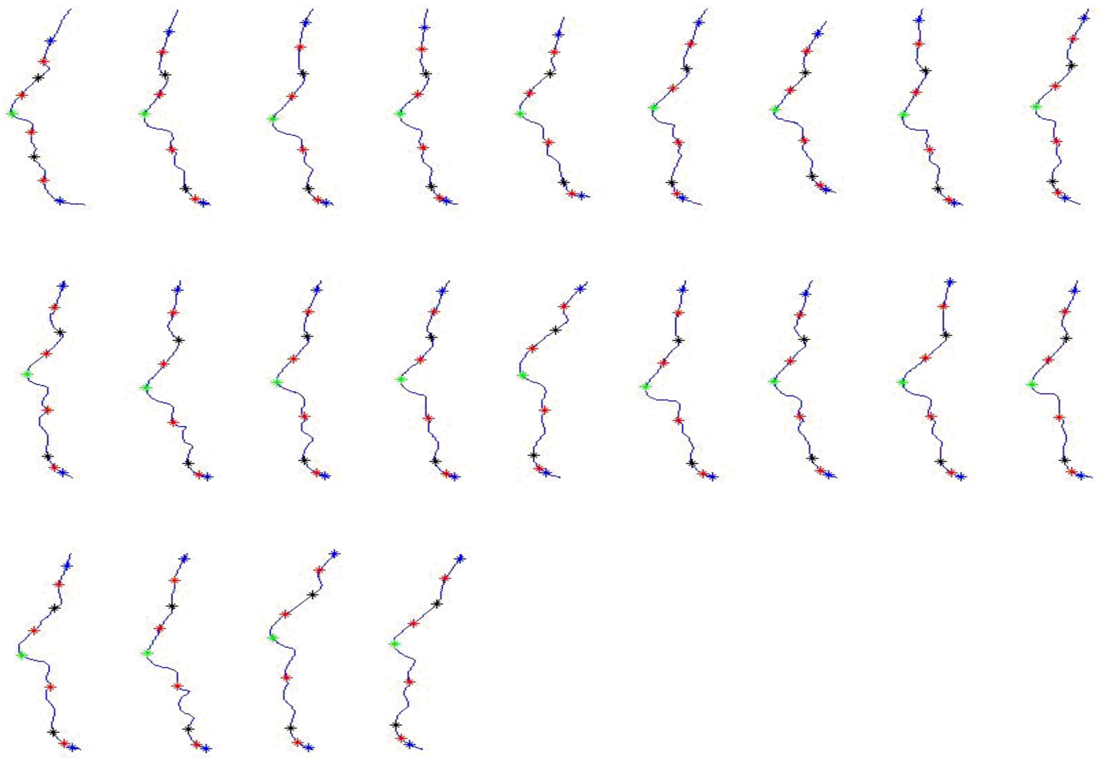


Figure 6.36 Results of MDL analysis of silhouettes contours. Here all 22 examples are shown, they are one step before MDL finally converged (blue is level one, green is level two, black is level three and red is level four). It can be seen that the two points at the bottom (red and blue) tried to pile up although one fixed master example has been used (first one).

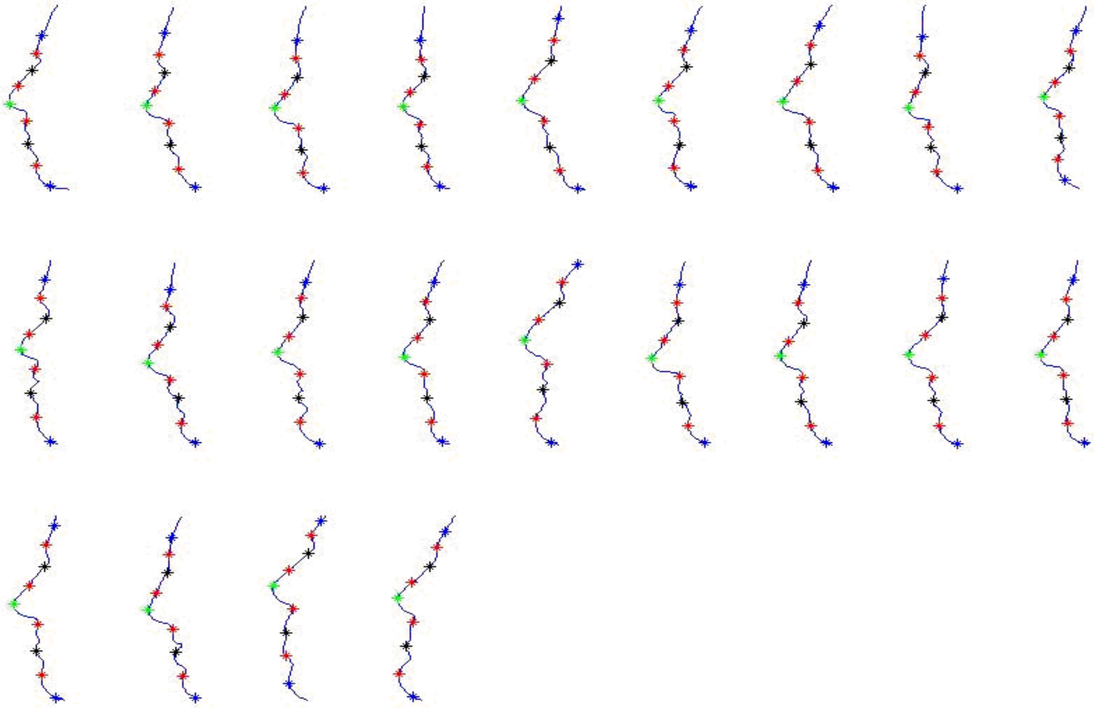


Figure 6.37 Results of MEM analysis of silhouettes contours.

In Figure 6.37, MEM results are shown. Compared with Figure 6.36, we can see that the MEM results do not suffer from the “Pile Up” problem, and the results are reasonably more accurate, all points are placed in the same manner across the datasets.

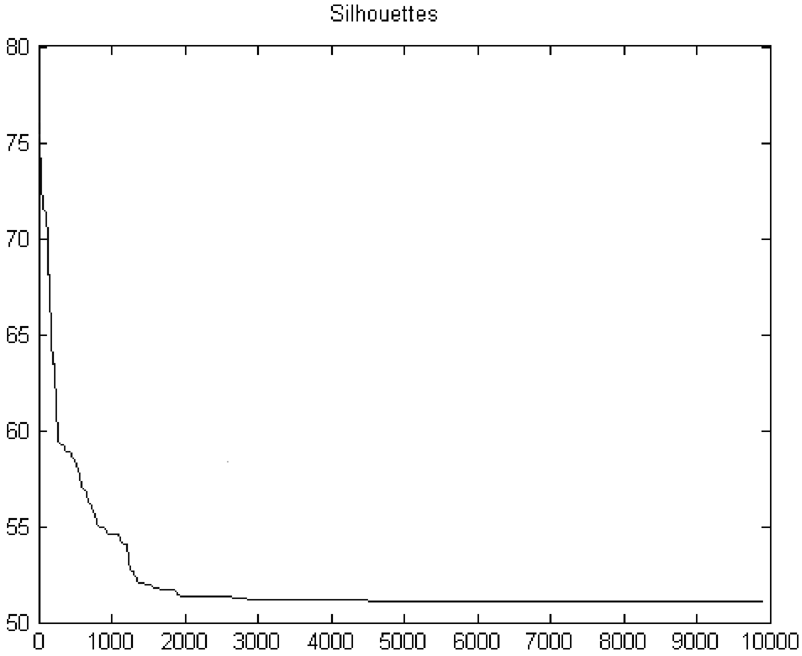


Figure 6.38 Performance of the MEM cost-function is shown here, X-axis is steps in time and Y-axis is output of cost-function. The blue solid line shows the performance by using our proposed MEM approach.

Figure 6.38 shows the performance of the cost-function. From this graph, we can observe the cost-function value decline during optimization. Once again, during the experiment we observe that the MEM can be optimized about 2.2 times faster than the MDL approach due to the usage of gradient information in MEM approach. Figure 6.39 shows the parameter value for each control nodes during optimization. The figure shows that all nodes start from value 0.5 (equal spaced situation) and gained a stabilization after a few steps. After correspondence points were found by the MEM algorithm, an Active Shape Model can be built to capture shape variations. The first three largest variations are shown in Figure 6.40.

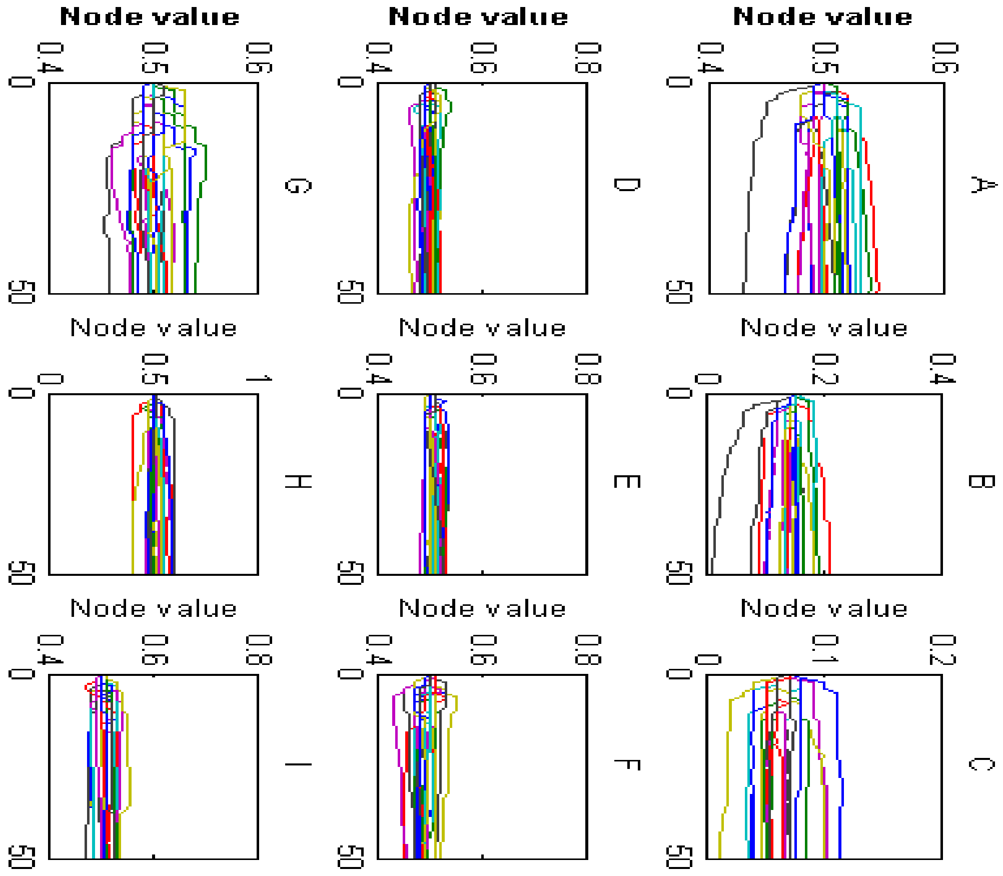


Figure 6.39 The changes of node value during optimization is shown in the graph. Each graph gives the movement of the node in 22 examples represented by different color. X coordinate is step number and Y coordinate is node value. All nodes start to move from parameter 0.5.

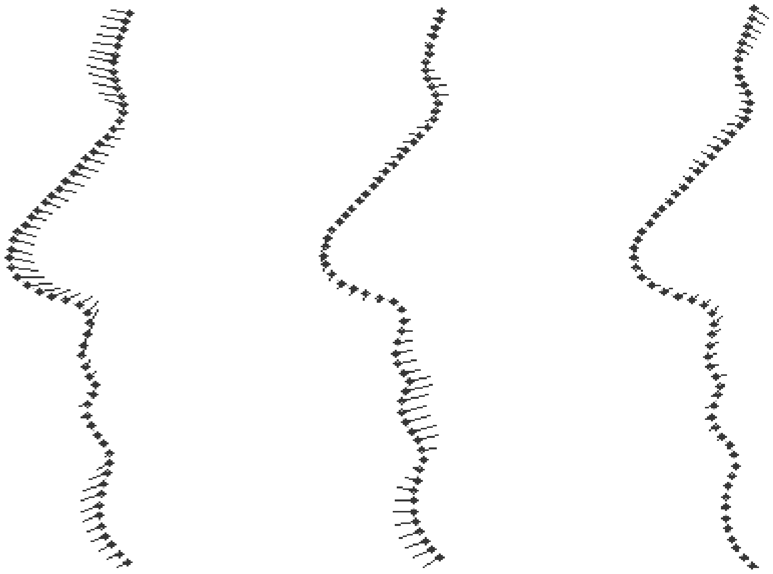


Figure 6.40 The changes of node value during optimization is shown in the graph.

6.5 Conclusions of the Experiments

In this chapter, we validated our proposed MEM algorithm by performing both MEM and MDL on several datasets for different 2D scenarios. They are closed curves, open curves with fixed ends, and open curves with free ends. Due to lack of ground truth, we adopted three generally accepted criteria to compare our proposed MEM with MDL. These criteria are Generalization Ability, Specificity and Compactness. Based on the results from the above paragraphs, several conclusions can be drawn. From the performance of the MEM cost function, we can see that the cost function converged during optimization and the converged results produced reasonable correspondence results. Based on the cost function performance from both MDL and MEM, it can be observed that our proposed MEM approach can converge faster than the MDL approach. From our quantitative comparison results between MDL and MEM, it can be observed that MEM usually achieved better scores on Generalization Ability, and Specificity, and worse results on Compactness. In the case of open curves, it takes longer for the cost function to find correspondence under the scenario of fixed ends. Therefore, we suggest using free ends during optimization of datasets of open curves. The final experiment on facial profiles showed the advantage of using Entropy, rather than Description Length, as the component of cost function. In the experiment, MDL suffered the “Pile Up” problem, where bottom points on chin areas overlapped. However, MEM did not have this problem although the experiment was performed without using a master example and external cost function. We argue that MEM favours a distributed correspondence and MDL favours a congested correspondence.

Chapter 7 A 3D Minimum Entropy Approach and Experiment Results

7.1 Discussion on 2D Work

The MEM method introduced in the previous chapter provides a principled framework for automatic statistical shape model building. As can be seen from Chapter 6, MEM has been successfully applied to numerous 2D datasets. Different datasets, for example, closed curves, open curves with free ends, open curves with fixed ends, and open curves with complicated landforms were tested. According to the experimental results based on the comparisons between MEM and MDL (the current state of art), we can see that MEM provides better performance on Generalization Ability and Specificity and similar Compactness. Some good properties of MEM are also revealed, since it can keep the objective function away from the local minimum that often traps MDL without changing the function into an arbitrary manner. MEM favours a distributed correspondence, which makes balance between equal spaced results and congested results. On the contrary, MDL will pile up occasionally. The gradient of MEM is also proposed by using some useful results derived by SVD. With the help of cost function gradient, various optimization methods can be performed rather than the Genetic Algorithm used in the MDL approach. However, most of the medical image datasets are in 3D format [47], which requires an extension of our current 2D method. In this chapter, we will discuss some preliminary results on 3D MEM method.

7.2 Limitations of 2D MEM and MDL

From the previous chapter of our 2D results, we can see that the proposed MEM algorithm has provided a reasonably more robust (in terms of solving the “Pile Up” problem) and accurate (in terms of better Generalization Ability and Specificity) approach to solve the “Correspondence Problem”. It can find corresponding points across 2D datasets automatically. However, the algorithm is not perfect enough to solve all the correspondence problems yet. Since, there are still some issues in MEM to solve. For example, both MDL and MEM inherently do not consider images with missing information. Figures 7.1 and 7.2 are shown as an example of limitations of MEM and MDL approaches. When images with missing information are used, correspondence points found by both algorithms can be very wrong. In Figure 7.1, since the dataset is in 2D, parts of the bird’s shape is blocked by itself, the final results are heavily influenced by this. Therefore, if 3D shape information can be used, the optimal correspondence results can be improved. Most of the medical datasets are in the form of three Dimensions. Therefore, to make our proposed method more applicable to real medical image processing task, we will have to extent our current 2D scheme into 3D.

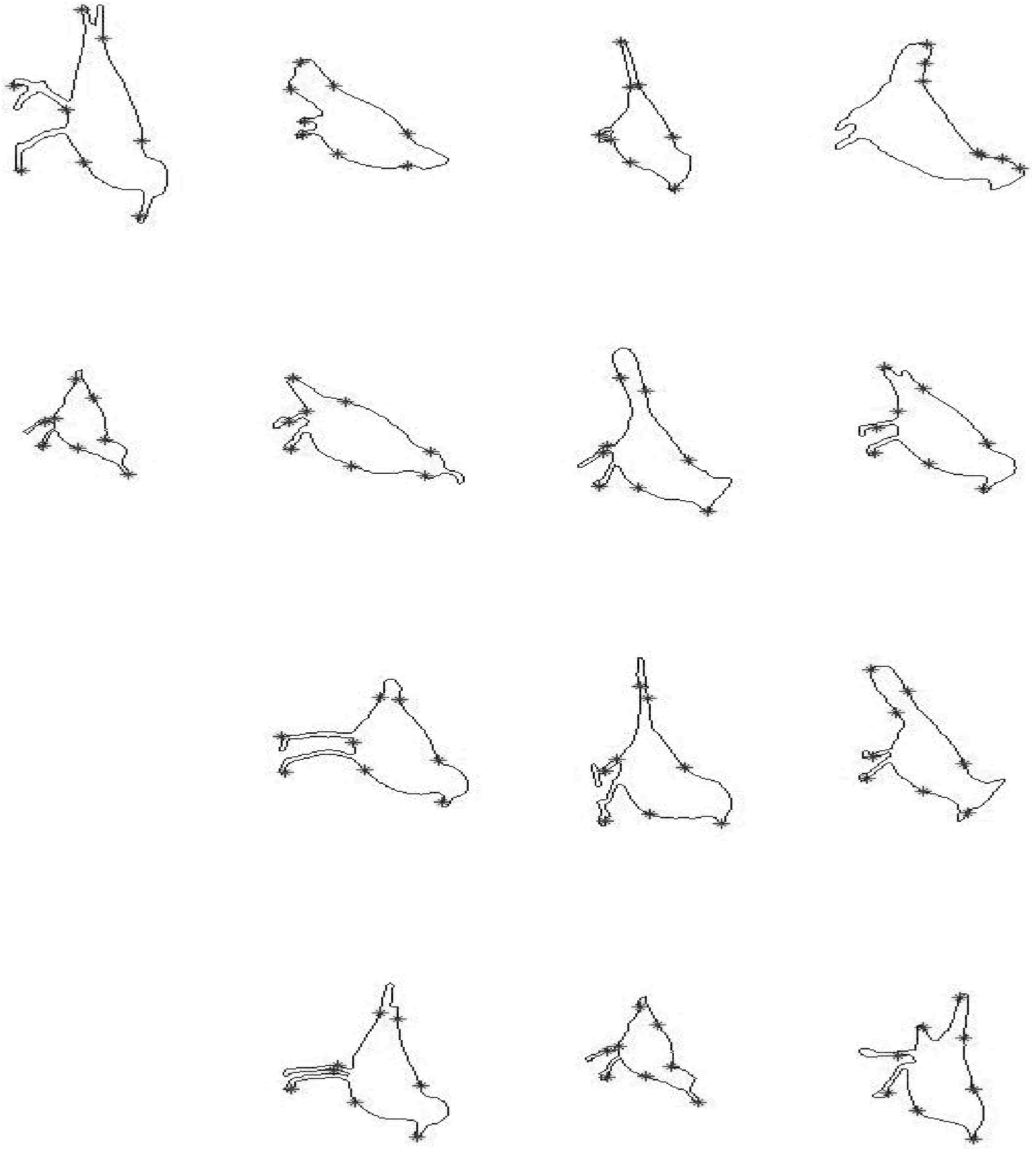


Figure 7.1 Correspondence points found by MEM are shown in this graph.

Figure 7.1 above, shows the correspondence points found by our proposed MEM algorithm. In this the experiment, we used 8 control points during optimization and as in the previous 2D experiments, the rest of the landmarks in between control points are set equally spaced. In Figure 7.2, the MDL algorithm was used to find correspondence points across shapes automatically. As in MEM, 8 control points are used and landmarks in between control points are also set equally spaced.

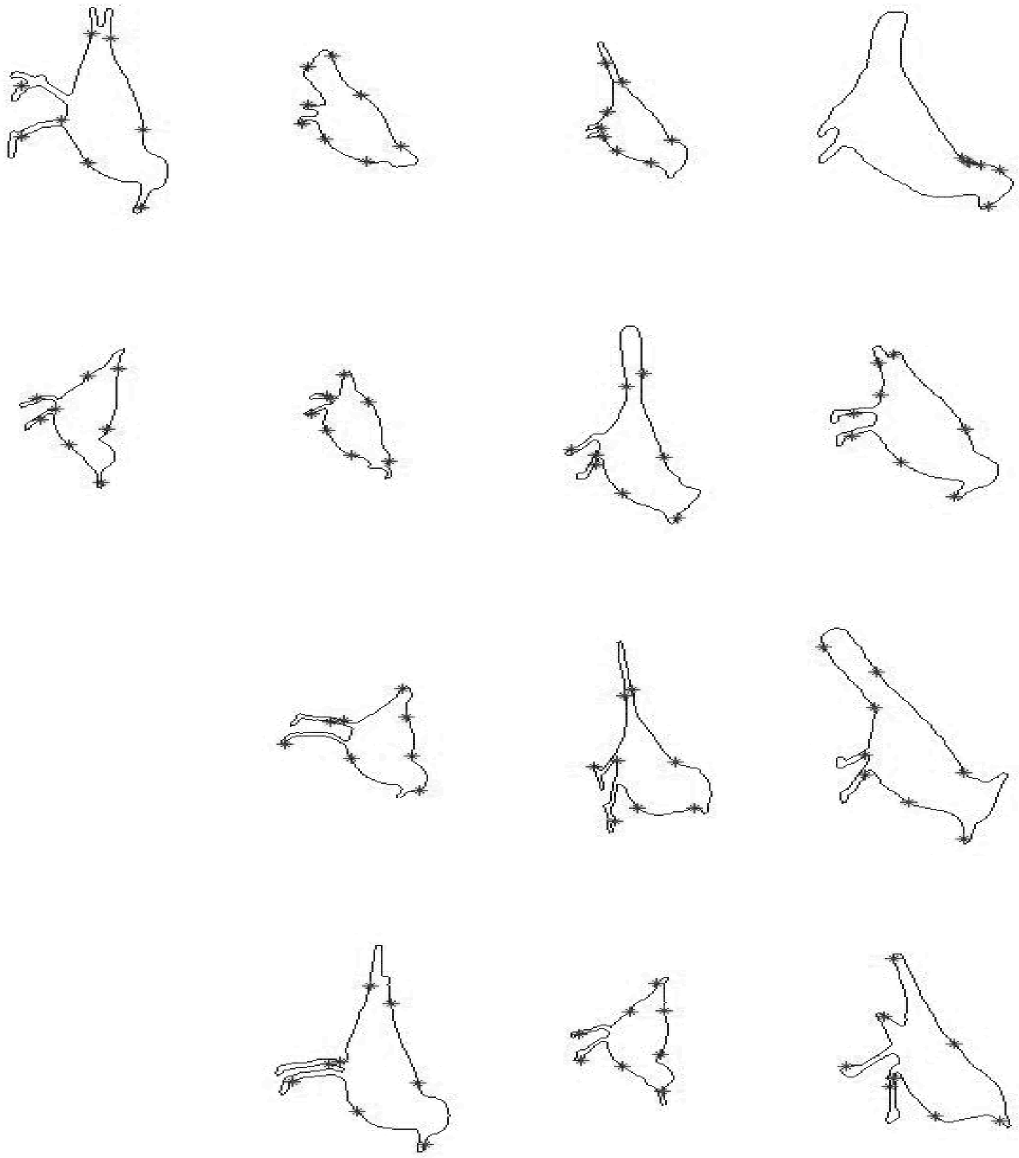


Figure 7.2 Correspondence points found by MDL

It can be seen from Figure 7.1 and 7.2 that both algorithms found correspondence points in a wrong fashion. The cause of these wrong results can be interpreted as in this particular dataset has some structure information which is not shared by each example. For example, they have different type of tail (some has one tail, some has two tails), feet and some of the shape variations are hidden due to the changes of 3D view angle. These differences are marked in Figure 7.3.

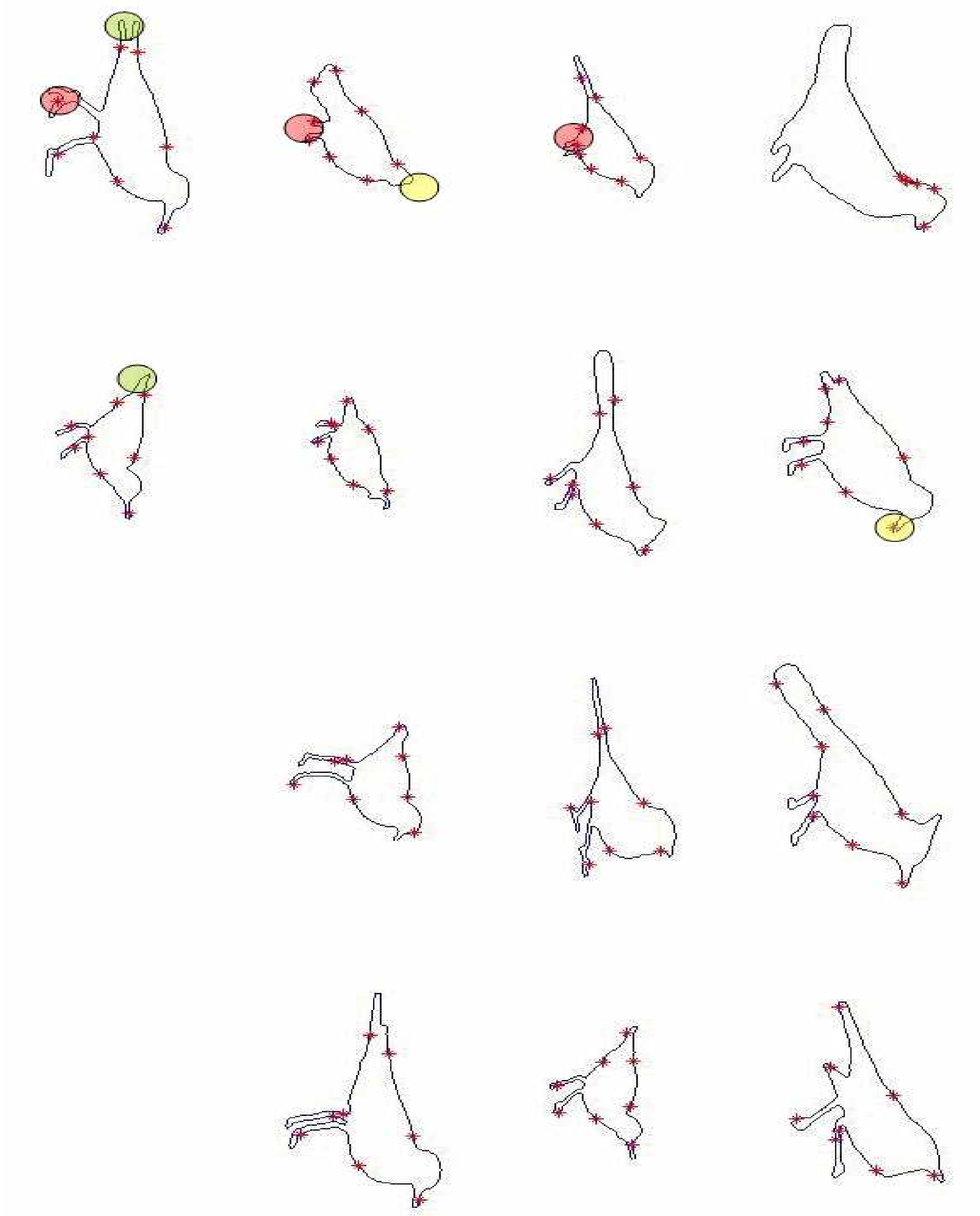


Figure 7.3 Coloured round masks are used to emphasize the differences between corresponding structures. Green is marked on tails, some tail has one branch and the other has two. Yellow marks the bird beak, because some pictures are shot from behind, therefore the beak is not shown. Red marks feat, for some of the birds, only one branch can be observed due to camera angle and 3D rotation.

The problems, reported above, are not considered in algorithms MEM and MDL. MEM and MDL are applicable to datasets $\{A_1, A_2, \dots A_n\}$ under the condition that data have corresponding information to each other (for any structure in A_i, there is a corresponding structure in all other datasets).

In this birds examples, some of the missing information is due to changes of view angles and different poses, which obscure the information in images. In 2D, it will become quite difficult to solve this problem, which was introduced by 3D shape variations. Therefore, in this chapter we will provide our natural extension of MEM in 3D. In the next few sections, we will discuss a new 3D scheme of MEM algorithm and some experiments on 3D datasets are also presented.

7.3 MEM on 3D

This chapter shows how our previously stated 2D methods can be extended to 3D. Though intuitive, something straightforward in 2D is not that simple in 3D. In 3D MEM algorithm, several problems have to be solved. They are: (i) refining the surface parameterization and (ii) re-parameterization method (correspondence method). In this section, we are going to discuss the details of 3D scheme. We use an existing technique to tackle the shape parameterization and re-parameterization, and combine them with our proposed MEM cost-function to form a completely new scheme.

The parameterization method has to be refined. For example, a 2D shape is parameterized by absolute length and mapped to a curved line. In 3D cases, shape surfaces are mapped to a unit sphere, where shape is defined by only two parameters (longitude and latitude). This parameterization has to be unique and invertible to make sure of a valid mapping. Re-parameterization also needs to be refined to fit the 3D approach. Even, the initial parameterization method has to be reformed accordingly.

We use conformal shape mapping [52] as a bridge between datasets and unit sphere to solve the 3D shape parameterization problem. Gaussian envelope function and random rotation matrix method are fitted into our approach to tackle the 3D shape re-parameterization problem. The initialization position is realized by dividing the unit sphere along longitude and altitude equally. These points are then mapped to original shape space to locate the initialization positions. Finally, the gradient of 3D cost function and optimization scheme are established. The conclusions are based on our comparison results on 3D artificial datasets and hippocampus datasets.

7.3.1 Surface Extraction

In the scope of this thesis, we are working on segmented datasets. In 2D, they are shape contours, which are saved as points along the boundary. In 3D, we are mostly working with binary data, under the assumption that 1 represents the object, and 0 the background. Therefore, surface extraction techniques have to be utilized to cope with our MEM algorithm. In reference [64], the Marching Cubes algorithm, also called iso-surface extraction, is described. This algorithm is chosen in our 3D scheme to generate the mesh from binary datasets. The basic rationale behind this idea is that we can define a voxel (cube) by the pixel values at the eight corners of the cube (Figure 7.4).

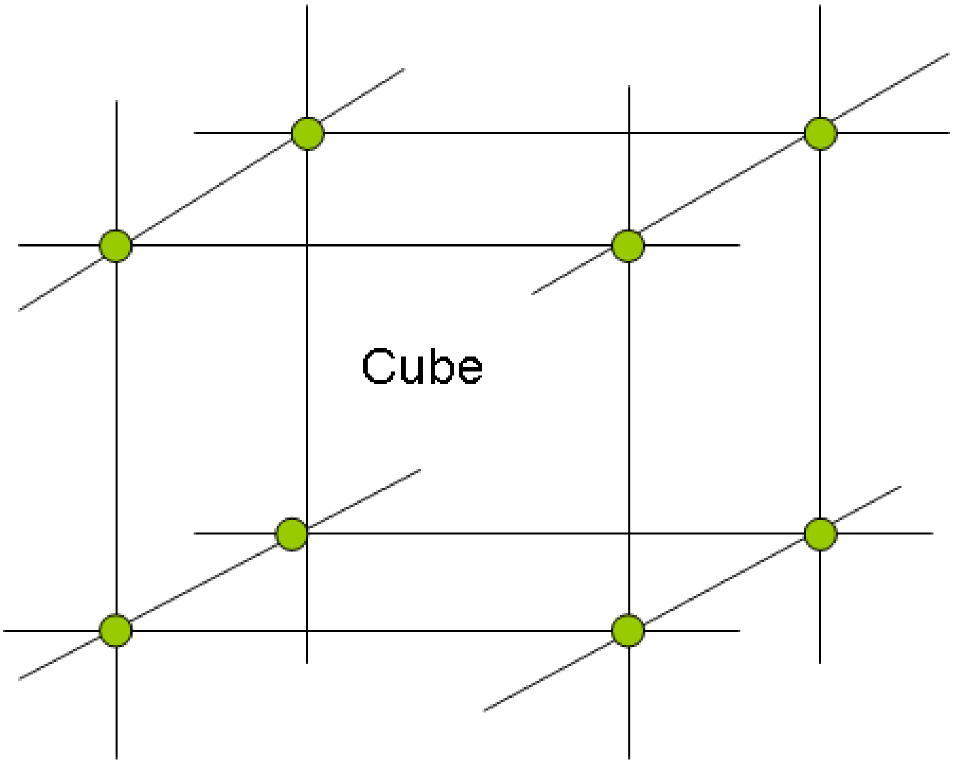


Figure 7.4 The definition of voxel/cube is shown in this graph.

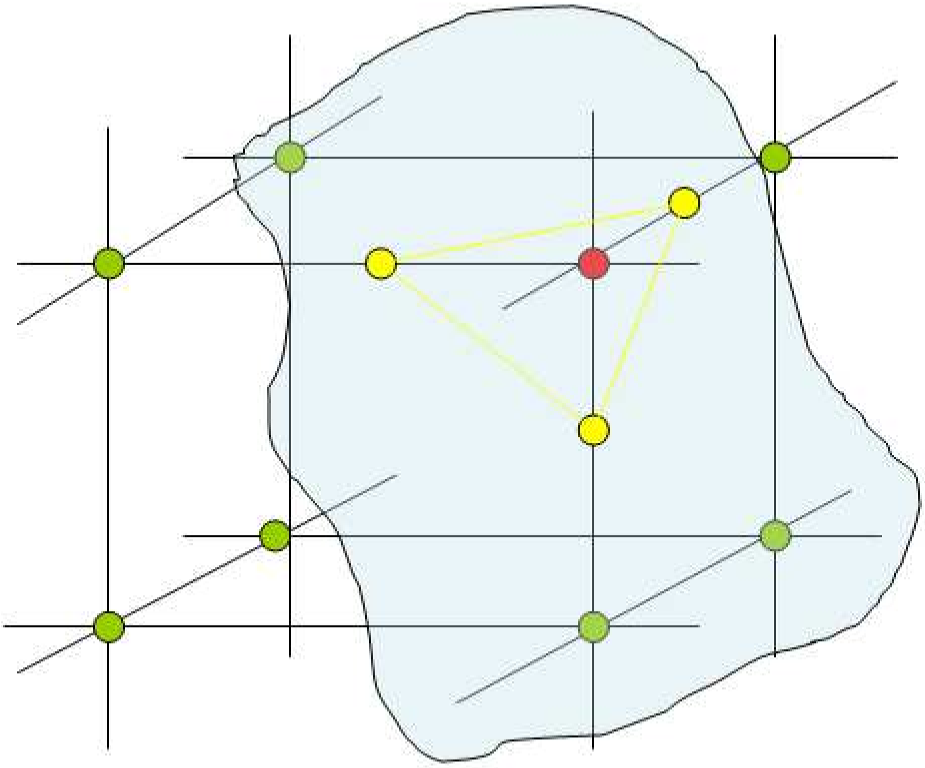


Figure 7.5 The cube is cut by object surface on yellow points, red point is in the background.

If one or more pixels of a cube have values less than the user-specified iso-value, and one or more have values greater than this value, then we know that the voxel must contribute to some component of the iso-surface. By determining which edges of the cube are intersected by the iso-surface, we can create triangular patches, which divide the cube between regions within the iso-surface and regions outside, see Figure 7.5.

In 3D space, we enumerate 256 (i.e. 2^8) different situations for marching cubes. In Figure 7.5, only one simple situation is shown. By connecting the patches from all cubes on the iso-surface boundary, we get a surface representation. The problem with the marching cubes method is that it can generate large numbers of surfaces, which are more than we need. Thus, down sampling of the surface is usually needed to keep the same number of vertices and faces. After this process, the datasets are saved into two separate files, one with extension *pts*, another one with *fce*. File with extension *pts* contains the vertex coordinates, which is a $N \times 3$ matrix. Each row represents the coordinates of one vertex and these vertices are indexed the number of row from 0 to $N-1$. File with extension *fce* keeps the vertex relations, e.g. which 3 vertices form a face. Therefore, with these two files, we can recover the original data surface information. Moreover, these two files are quite easy to use for parameterization.

7.3.2 Shape Parameterization

In order to minimize the complexity of the parameterization of 3D shapes, we will limit our discussion to the closed two-manifolds of genus zero only (which means that the surface can not be allowed to fold, tear and self-intersect). Objects of this kind are topologically equivalent to a unit sphere and there exist medical image data, which belong to this class, such as the liver, kidney, lungs, and brain. An example of brain data and mesh found by Marching Cubes algorithms are shown in Figures 7.6 and 7.7 respectively.

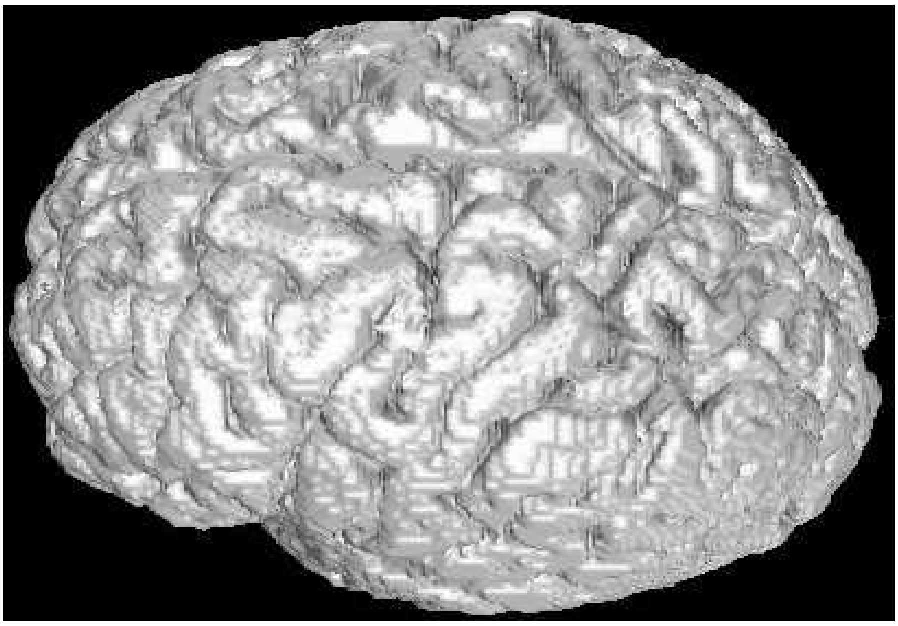


Figure 7.6 This is an example of 3D human brain data from reference [64]

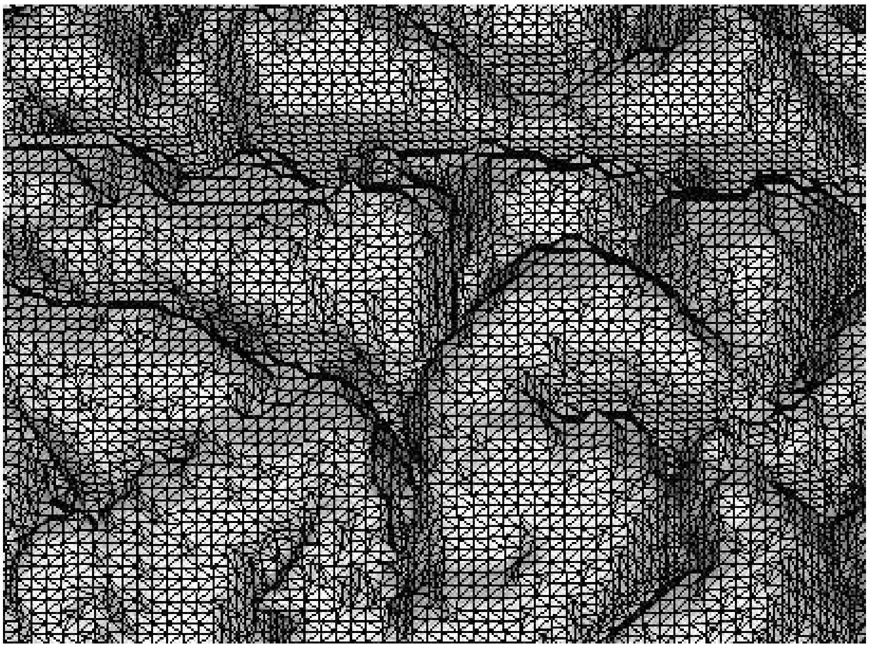


Figure 7.7 This shows a magnified display of a brain surface constructed by using marching cubes. This picture is from reference [64]

Therefore, our task is to find the means (i.e. a parameterization function Φ) to map our data on a unit sphere such that this process should be easily reversed (Figure 7.8 and 7.9).

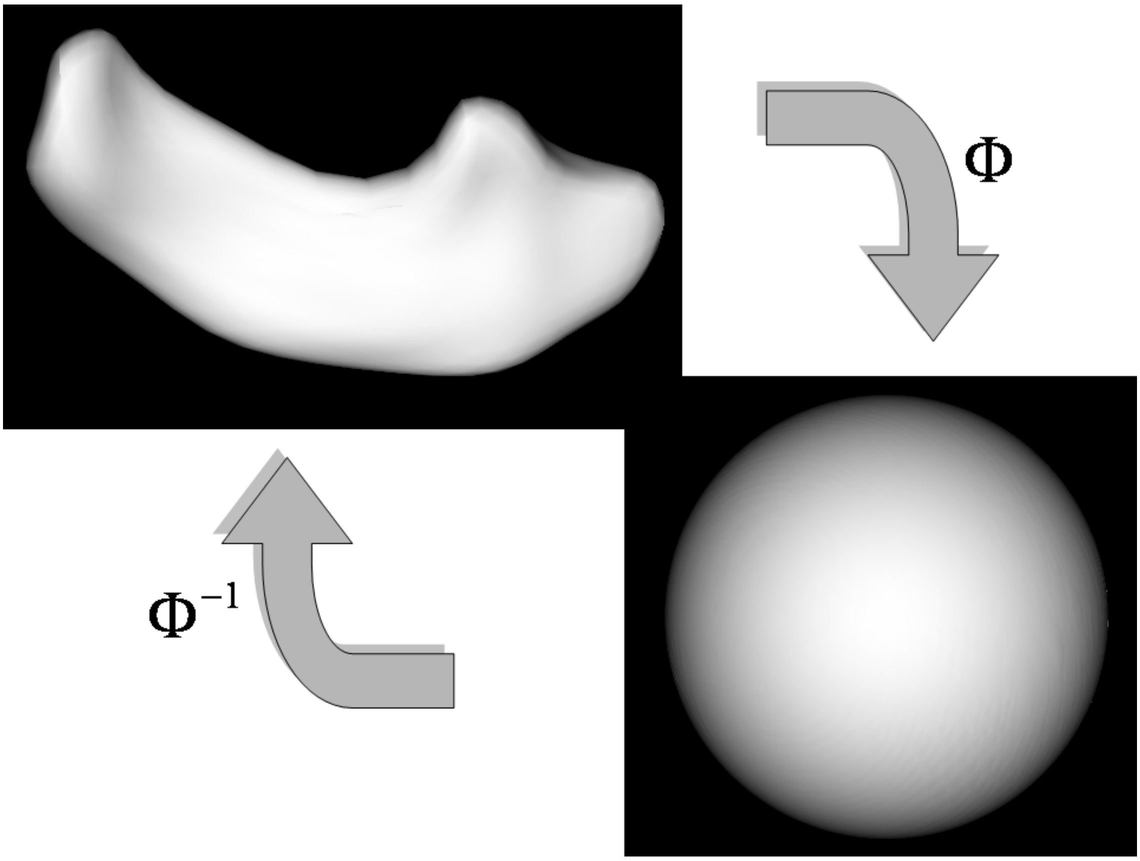


Figure 7.8 A demonstration of mapping between shape and sphere. Φ **is the bridge between original dataset and parameterized sphere.**

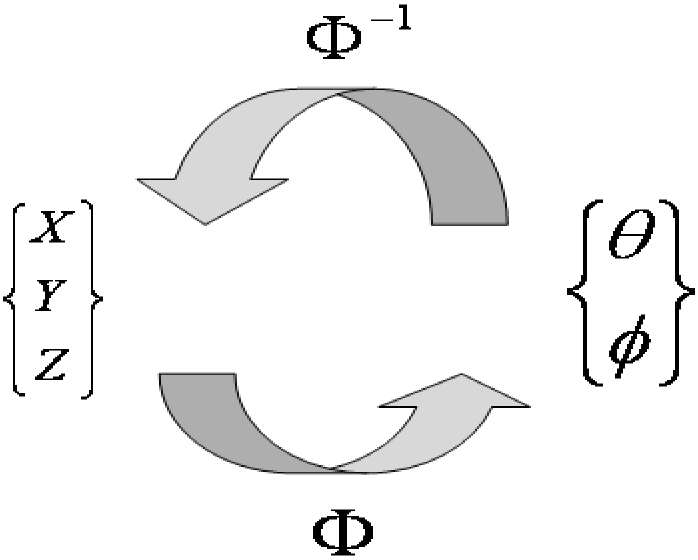

Φ^{-1}
$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}$
$\begin{Bmatrix} \theta \\ \phi \end{Bmatrix}$
Φ

Figure 7.9 This graph shows mapping and inverse mapping between shape space and spherical space.

If successful, we can downgrade the complexity of our algorithm to two dimensions of longitude θ and latitude ϕ. Mapping one arbitrary shape to unit sphere will obviously introduce some error. Therefore, metrics are designed to minimize this type of error. For example, metrics are created to preserve either local angles or triangle face relations while trying to minimize the distortions of other features. Davies et al [44] have proposed using a simplified version of the spherical harmonics method, which is described in [51]. In our proposed 3D algorithm, we use an angle preserving method, which is the Conformal Mapping (CM) from Gu et al [52]. We used a public available CM from the Insight Journal (www.insight-journal.org) and ITK (www.itk.org). This shape parameterization method introduces minimum angle distortion. Compared with Davies et al approach, Conformal Mapping offers minimum distortion on angles, which means moving clouds of points on the parameterized shape in a specific direction will cause the corresponding landmarks on the training shape to move in a coherent direction as well. This kind of shape parameterization offers a convenient method to retain local geometric information, when mapping data between surfaces.

We will begin to discuss this method by first presenting Algorithm 1, based on reference [51].

Algorithm 1 Each dataset is represented as a triangulated mesh $K = (V, E)$, with *V* denoting Vertices, and *E* Edges. Vertex locations are specified by function $f : V \rightarrow R^3$, which is an embedding function defined on the original vertices of *K*. A second function $\Psi : V \rightarrow R^3$ specifies the verted location as mapped to the unit sphere.

Definition 1 $\omega(v)$ represents the normal vector of *v*.

Then, conformal mapping energy is given by the equation below.

$$E(K,\Psi) = \sum_{[u,v]\varepsilon E} K_{u,v} \left\| \omega(u) - \omega(v) \right\|^2 \qquad \mathbf{(7.1)}$$

This form is string energy with all edge weights $K_{u,v}$. Therefore, minimizing this energy is with condition that edge weights $K_{u,v}$ are controlled by Equation 7.2.

$$K_{u,v} = \frac{1}{2}(\cot\alpha + \cot\beta) \qquad \mathbf{(7.2)}$$

Where α and β are opposing angles on edge (u,v). On the other hand, setting $K_{u,v}$ to 1 will yield barycentre mapping, where each vertex is positioned at the centre of its neighbourhood. For more details, it is recommended to read reference [53]. A demonstration of this conformal mapping is shown in Figure 7.10 below. More results about conformal mapping are shown in Figure 7.19.

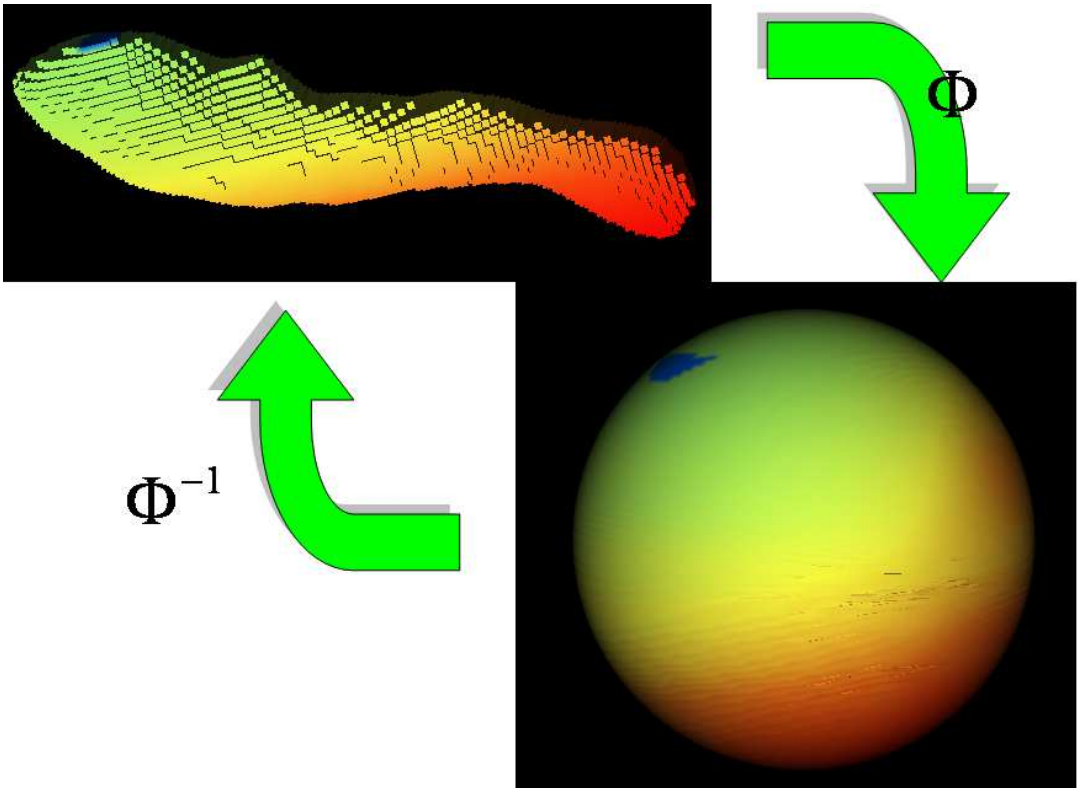


Figure 7.10 The colour coded correspondence is shown in this figure. Original dataset is colour coded, therefore corresponding points between original data and sphere can be found by identical colour.

Until now, we only discussed the shapes with spherical topology. Although, this is appropriate for most of the medical image data we are investigating, it is interesting to consider shapes, with other kind of topologies. According to my limited knowledge about finding correspondence in a learning process, there is no robust way for using either MDL or MEM on other topological shapes. However, in reference [22], Davies proposed some approaches. Since some methods have been used to parameterize 2D shapes with open (R^1) or closed (S^1) end, it can be helpful to think about shapes as in the space of $R^1 \times S^1$. In this way, some 2D techniques can be utilized. However, this is only a computation trick; a more general method should be developed in future research.

7.3.3 A Continuous Parameterization

Sometimes, in order to obtain a 3D position for an arbitrary landmark/node, which is not a vertex on the dataset, we have to find a position in between landmark nodes. Since mapping landmarks between shape and sphere is computationally quite expensive in our algorithm (it takes about half of our whole computation time), we came to the conclusion that the advanced intersection strategy called likelihood search has to be employed, this intersection approach was introduced in reference [54]. The theory is that intersected triangles are cached and in case of missing cache, neighbour landmarks will be given higher priority to be probed. For simplicity, we adopt the intersection based on barycentric co-ordinates. Equation 7.3 defines (ξ_1, ξ_2, ξ_3) as the bar-centric co-ordinate of new vertices v with respect to triangle t_k, see Figure 7.11.

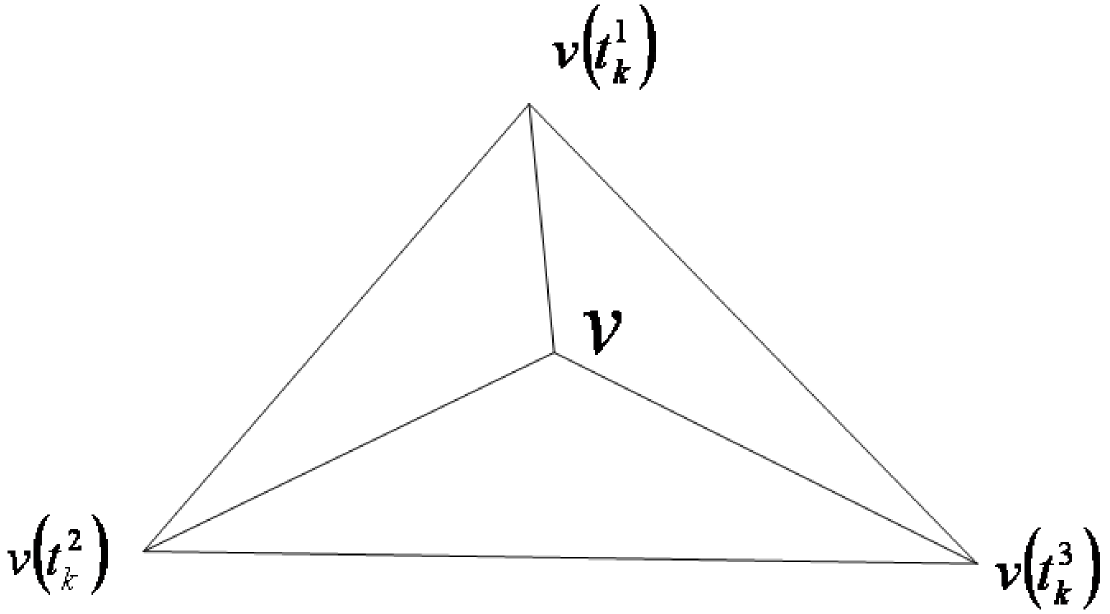


Figure 7.11 How a new point is inserted into an existing triangle, where a new point is inserted into a triangle.

$$\xi_1 = \frac{Area(v, v(t_k^1), v(t_k^2))}{Area(v(t_k^1), v(t_k^2), v(t_k^3))}$$
$$\xi_2 = \frac{Area(v, v(t_k^3), v(t_k^2))}{Area(v(t_k^1), v(t_k^2), v(t_k^3))} \qquad \textbf{(7.3)}$$
$$\xi_3 = \frac{Area(v, v(t_k^1), v(t_k^3))}{Area(v(t_k^1), v(t_k^2), v(t_k^3))}$$

Where Area(ABC) is the area of the triangle *ABC*. Triangle t_k contains *v* with the conditions that $\xi_1 > 0, \xi_2 > 0, \xi_3 > 0$ and $\xi_1 + \xi_2 + \xi_3 = 1$. Therefore, with a vertex *v*, on the original data, its barycentric co-ordinate can be used to define the corresponding point on the sphere.

$$u(\theta_v, \phi_v) = \xi_1 \Phi_1\left(v\left(t_k^1\right)\right) + \xi_2 \Phi_2\left(v\left(t_k^2\right)\right) + \xi_3 \Phi_3\left(v\left(t_k^3\right)\right) \qquad \textbf{(7.4)}$$

The reverse mapping can be found in the same way.

7.3.4 Shape Re-Parameterization

To cope with 3D cases, we need a transformation function, which can manipulate points around the shapes. In Davies *et.al.* [44], they use a symmetric theta transformation, which employs a Cauchy Kernel representation. The Cauchy Kernel function is addressed in [46], is uni-model, symmetric function of the form:

$$f(x) = \frac{1}{\pi}\frac{\eta}{\eta^2 + (x-a)^2}, \quad -\infty < x < \infty, \quad \eta \geq 0 \qquad \textbf{(7.5)}$$

Where, η is the width of the Cauchy, and *a* is the position of the centre. By using this kernel, the normalized $\phi(u)$ that lies in the range [0, 1].

$$\Phi(x) = c\left[u + b - \sum_k \frac{A_k}{\pi}\arctan\left(\frac{a_k - u}{\omega_k}\right)\right] \qquad \textbf{(7.6)}$$

Where

$$b = \sum_k \frac{A_k}{\pi}\arctan\left(\frac{a_k}{\omega_k}\right)$$

$$c^{-1} = \left[1 + b - \sum_k \frac{A_k}{\pi}\arctan\left(\frac{a_k - 1}{\omega_k}\right)\right] + b \qquad \textbf{(7.7)}$$

Where A_k is the magnitude of the k^{th} kernel. The constant term ensures that $\phi(u) = u$ when all A_k are set to zero. The basic idea is to draw a great circle (which is the circle with the biggest diameter on the sphere) between any point *v* and a “fixed” point *m* on the sphere. The re-parameterization of the sphere can then be achieved by applying the same re-parameterization to each great circle using the kernel stated above. A demonstration of the re-parameterization is shown in Figure 7.12.

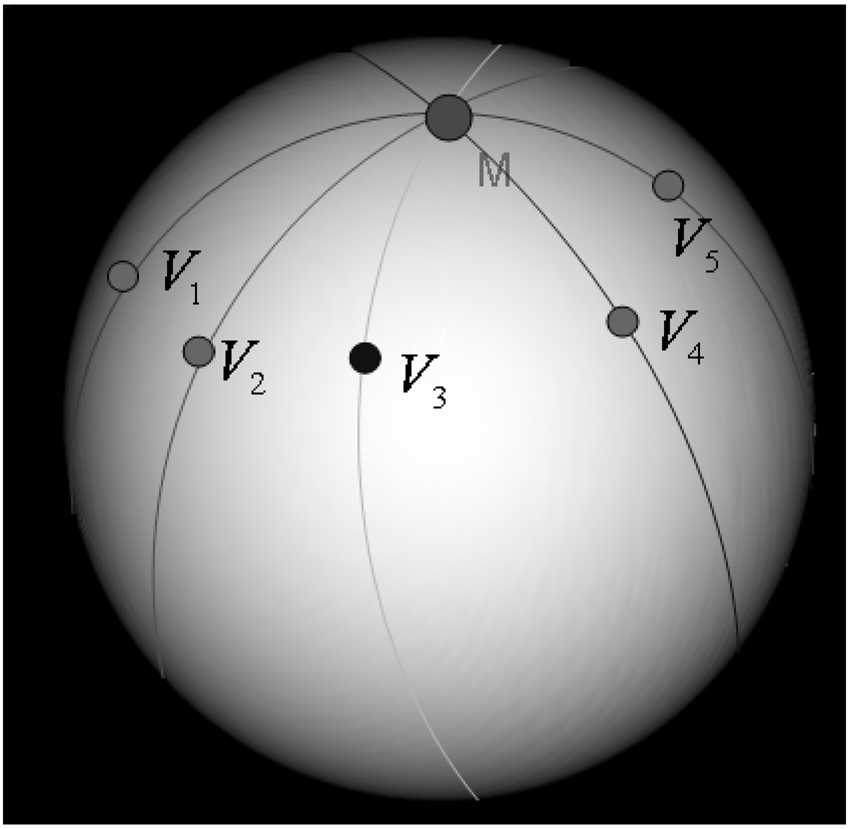


Figure 7.12 This graph shows how the sphere is re-parameterized. For any point *v* on the sphere, a great circle is drawn through it and a fixed point *M* (the centre of the kernel). Each great circle can then be re-parameterized according to the same function to the re-parameterization function.

Although, by accumulating thousands of kernels at different positions, arbitrary parameterizations can be achieved, this re-parameterization method produces the desired results in an inefficient way. Not only that, the main disadvantage is that it is a global modification. For example, adding one new kernel will change the locations of all landmarks. This is highly undesirable for this application. Therefore, we suggest another method for shape re-parameterization, which is based on kernels with strict local properties.

From the previous chapter, we have shown that the downhill direction of our cost function can be estimated by using products of SVD. Then, we assume that the gradient direction is $(\Delta\theta, \Delta\phi)$ in parameter space. Thus, we define a Gaussian Envelope function to change each spherical coordinates by $GE(x,\delta)\cdot(\Delta\theta,\Delta\phi)$. Function *GE* is given below:

$$GE(x,\delta)=\begin{cases} e^{\frac{-x^2}{2\delta^2}}-e^{\frac{-(3\delta)^2}{2\delta^2}}, \text{ for x<}3\delta, \\ 0, \quad \text{for } x\geq 3\delta \end{cases} \qquad \textbf{(7.8)}$$

However, this method will become restricted when kernels are allocated on either north or south pole, because landmarks will all be attracted to or pushed away from the poles depending on $\Delta\phi$. According to reference [55], by keeping the kernels away from the poles and randomly rotating our parameter space, the desired effect can be achieved. We show this process in Figure 7.13.

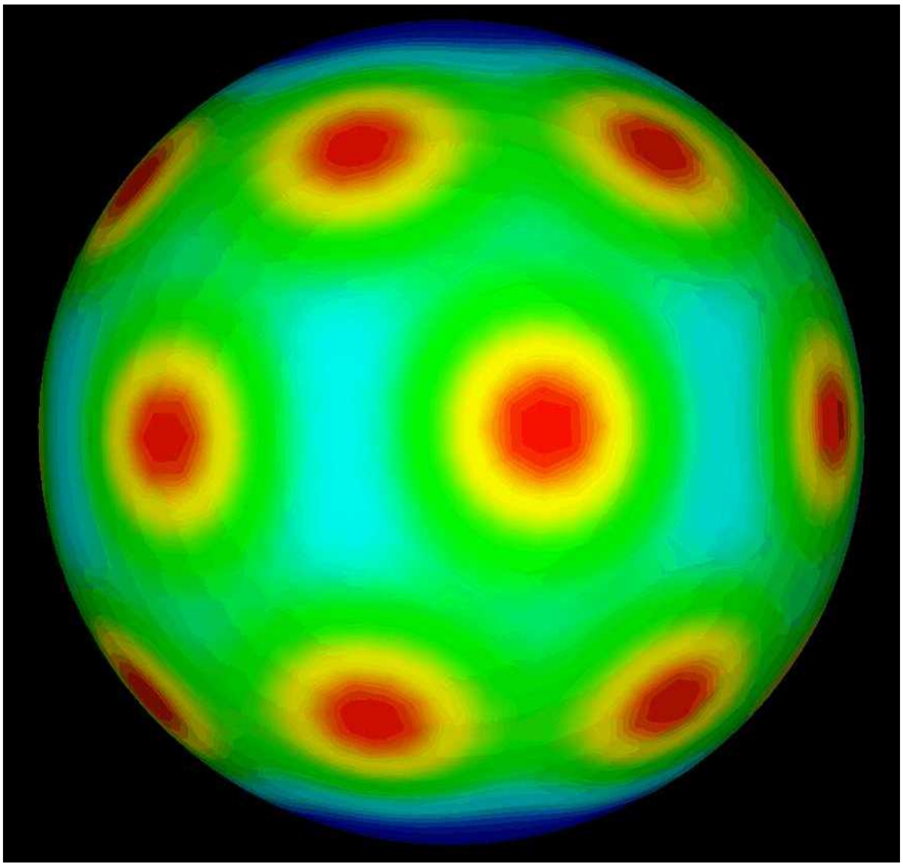


Figure 7.13 An example of kernels on unit sphere are shown here with $\delta = 0.2$. Red is the centre of the kernel. Colour changes from red to yellow to green, which shows the magnitude changing from high to low.

7.3.5 Initial Parameterization

Normally, after pre-processing, our datasets change from binary dataset to meshes with identical number of vertices and faces. Then, the next step is to provide a set of identical number of initial landmarks for the optimization strategy to work on. The initial landmarks can not be randomly generated, there has to be a rule which can be easily performed on each dataset. A good initialization can lead to a quick and reliable convergence. Therefore, in this section, we present our method of establishing initial landmarks on the mesh datasets.

Based on the previous section about Conformal Mapping and Point Intersection, we can easily have continuous mapping relations between mesh datasets and spheres. Thus, we can establish our initial landmarks by equally dividing the sphere along its latitude and longitude. An example of the initial set of marks on the sphere is shown in Figure 7.14.

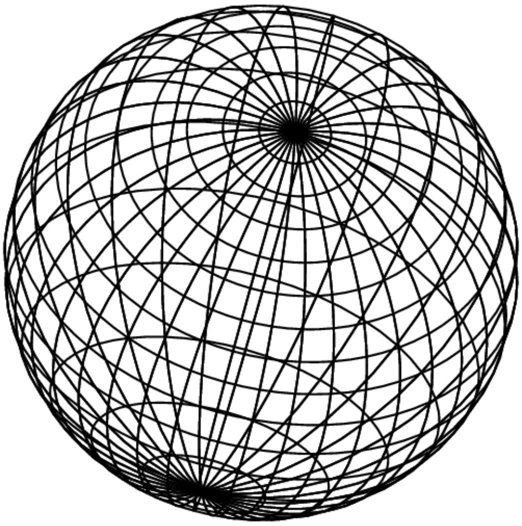


Figure 7.14 An initial shape re-parameterization is shown on the unit sphere.

Using reverse Conformal Mapping, we can map this mesh from parameter space to the original Cartesian shape space.

7.3.6 MEM in 3D

In this paragraph, we will discuss the MEM cost function in 3D and its gradient. On the aspect of cost function, MEM is the same as in the 2D case. The only difference is the co-ordinates system changes from two to three Dimensions. Recall Equation 5.13, we will use the same formula in Equation 7.9.

$$Costfunction = \sum_{j=1}^{t} \lambda_j H_j \qquad \textbf{(7.9)}$$

However, the gradient of cost function is slightly different between 2D and 3D scenarios. Here, we are keen to transform the calculated gradient fields into optimal kernel movements, e.g. $u = (\theta, \phi)$ on the parameter space of unit sphere. Therefore, we will calculate the gradient of the MEM objective function with respect to individual landmarks.

$$\frac{\partial MEM}{\partial u} = \frac{\partial MEM}{\partial X} \frac{\partial X}{\partial u} \qquad \textbf{(7.10)}$$

Recalling from 2D MEM gradient, we can get the similar results:

$$\frac{\partial MEM}{\partial u} = 2\sum_{i=1}^{n_s} H_i s_{ns} u_{mi} V \frac{\partial X}{\partial u} \qquad \textbf{(7.11)}$$

Here again, the surface gradient $\frac{\partial X}{\partial u}$ can be estimated by using finite differences. In practice, there are other variables, which will influence landmark positions, e.g. the random rotation in shape re-parameterization. By calculating the gradient of rotating parameterization sphere in 3D Euclidean space and substituting the surface gradients in

Equation 7.8, we can have a very efficient optimization strategy.

7.3.7 Optimization Scheme

In this section, we are going to present the MEM optimization framework details of the proposed algorithm. The framework is also shown as a flow chart in Figure 7.15.

Preparation: Datasets are pre-processed; meshes are generated from binary files. Initial landmarks (say 642) will be placed in parameter space.

Step 1: Based on the knowledge of conformal mapping, we first build the connection between shapes and unit sphere.

Step 2: The gradient of the MEM cost-function is calculated for the purpose of a quick convergence. A small step of movement is made along the direction of gradient on the sphere.

Step 3: Re-parameterization is used to move landmarks along the coherent gradient direction on the original shape space.

Step 4: MEM cost-function value is calculated and compared with previous calculated values, if not converged, the program will go back to Step 2. If converged, for example, the difference between the current and previous cost-function value is small enough (smaller than a predefined threshold), the algorithm will terminate.

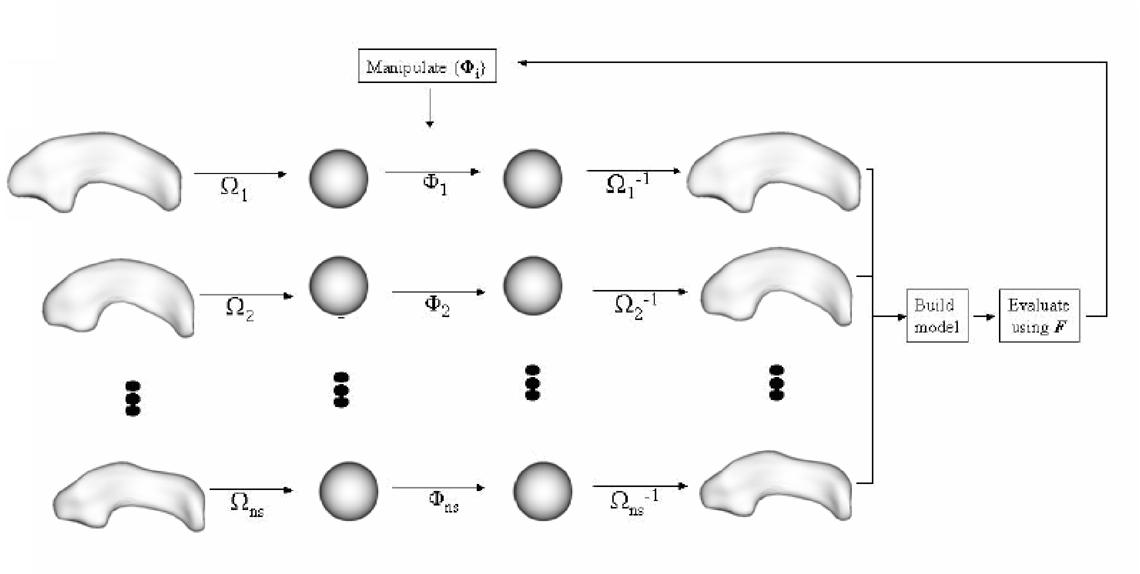


Figure 7.15 An illustration of 3D model building scheme, from reference [44].

In Figure 7.15, the whole scheme of automatic 3D model building is shown. Ω_i is the mapping from the i^{th} training shape to a sphere, Φ_i is the re-parameterization function, and F is the symbol for the MEM cost function. This graph is similar to reference [44].

In conclusion, MDL and MEM use similar shape parameterization techniques. For example, both parameterization methods using a reversible mapping technique to map 3D shapes to a unit sphere. However, in MEM, the angle is preserved during shape mapping. Therefore, moving a point in the shape space, the corresponding point on the sphere will move coherently. For correspondence manipulation, MEM uses the steepest descent algorithm and MDL uses the Genetic Algorithm. For initialization method, both MDL and MEM start optimization from equal spaced positions. In the next sections, we will perform some experiments on 3D artificial datasets and real medical image datasets to evaluate the performance of MDL and MEM quantitatively. During the comparisons with MDL, we use an published MDL approach from reference [101] after personal communications.

7.4 Experimental Results on 3D Datasets

This section presents the results of applying the MEM algorithm to one dataset of artificial cubes and hippocampus. A quantitative evaluation shows that the proposed MEM method provides better model properties than the alternative MDL approach. In the next few paragraphs, we show various graphs about Conformal Mapping results, quantitative comparison results, and correspondence points found by both MDL and MEM.

7.4.1 Visualization of 3D Datasets

7.4.1.1 Visualizing Node Correspondence

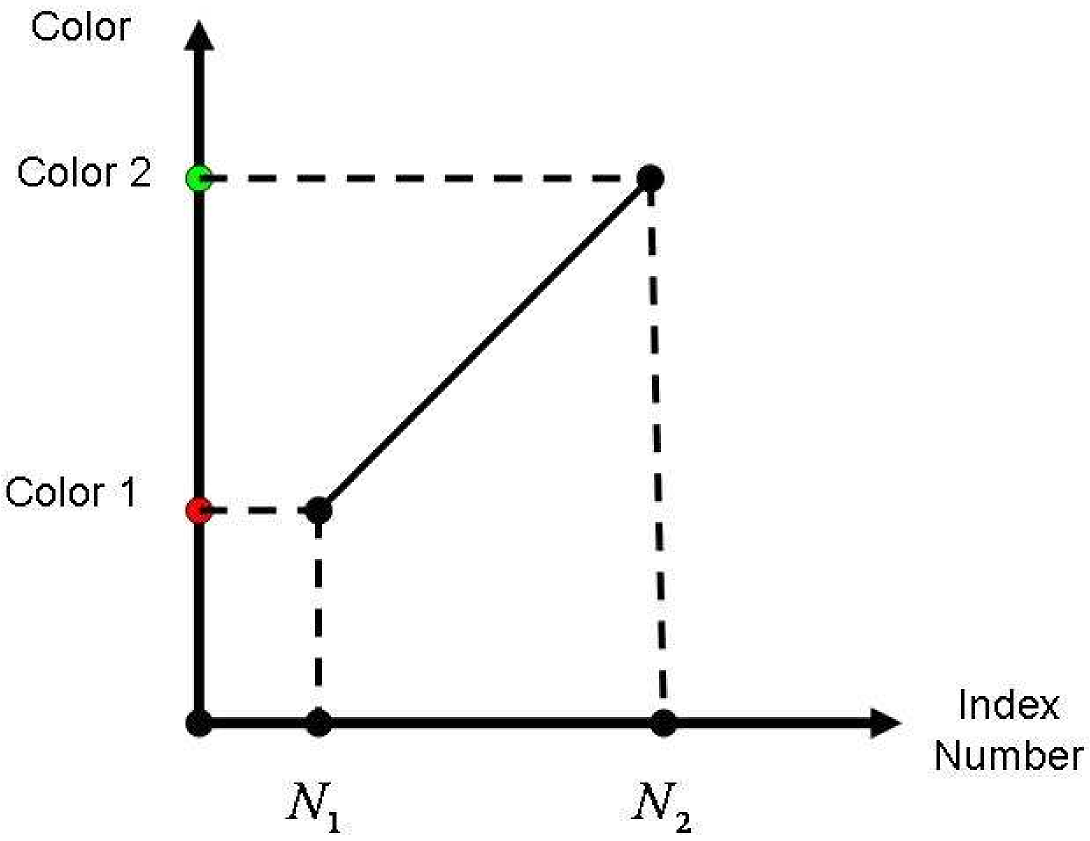


Figure 7.16 Colour Mapping method: X axis represents index number and Y axis represents colour space.

All our 3D datasets are saved by using more than a thousand surface points and during experiments, either 642 or 2562 number of landmarks will be used to identify the correspondence points. For the purpose of making the correspondence easier to see, we will use a colour mapping technique to ease this problem. The datasets are saved in such a format that points on the surface are indexed from 0 to N-1 (where N is the total number of nodes on the surface). Therefore, it is quite straightforward to use this index as a parameter to map into the colour space. A brief introduction about this colour mapping method is shown in Figure 7.16. As a result, all nodes on the original datasets and sphere are indexed, and assigned with proper colour. Since the colour mapping is a one-to-one mapping, the same colour will uniquely identify corresponding points. An example of the format of files used for this visualization provided in the Appendix section. By using this method, we can effectively identify the correspondence by finding the same colour across the shapes.

7.4.1.2 Visualizing Shape Variations

After applying the new technique to datasets, it is quite important to ensure the quality of the shape model visually. In this sense, some technique has to be utilized to show the properties of the shape model vividly. In Active Shape Model, the first few shape model variations of the model account for much of the shape model properties.

There are two ways to perform this particular visualization task. Firstly, we can perform the visualization by showing the effect of moving the first weighting component in the range of $\left[-3\sqrt{\lambda_1}, 3\sqrt{\lambda_1}\right]$. We can generate new shapes by using the shape model. Secondly, we can start from the mean shape, and assign each node with an arrow to point out shape variations.

An example of this visualization effect is shown in Figure 7.17. On the right, it shows the mean shape with blue arrows on each node; on the left, it shows two amplified views of the local structure. From this graph, we can observe that the shape variations captured by the first eigenvector are mainly located on the upper tail and bottom right corner. By using this visualization method, we can effectively detect the shape difference between subjects.

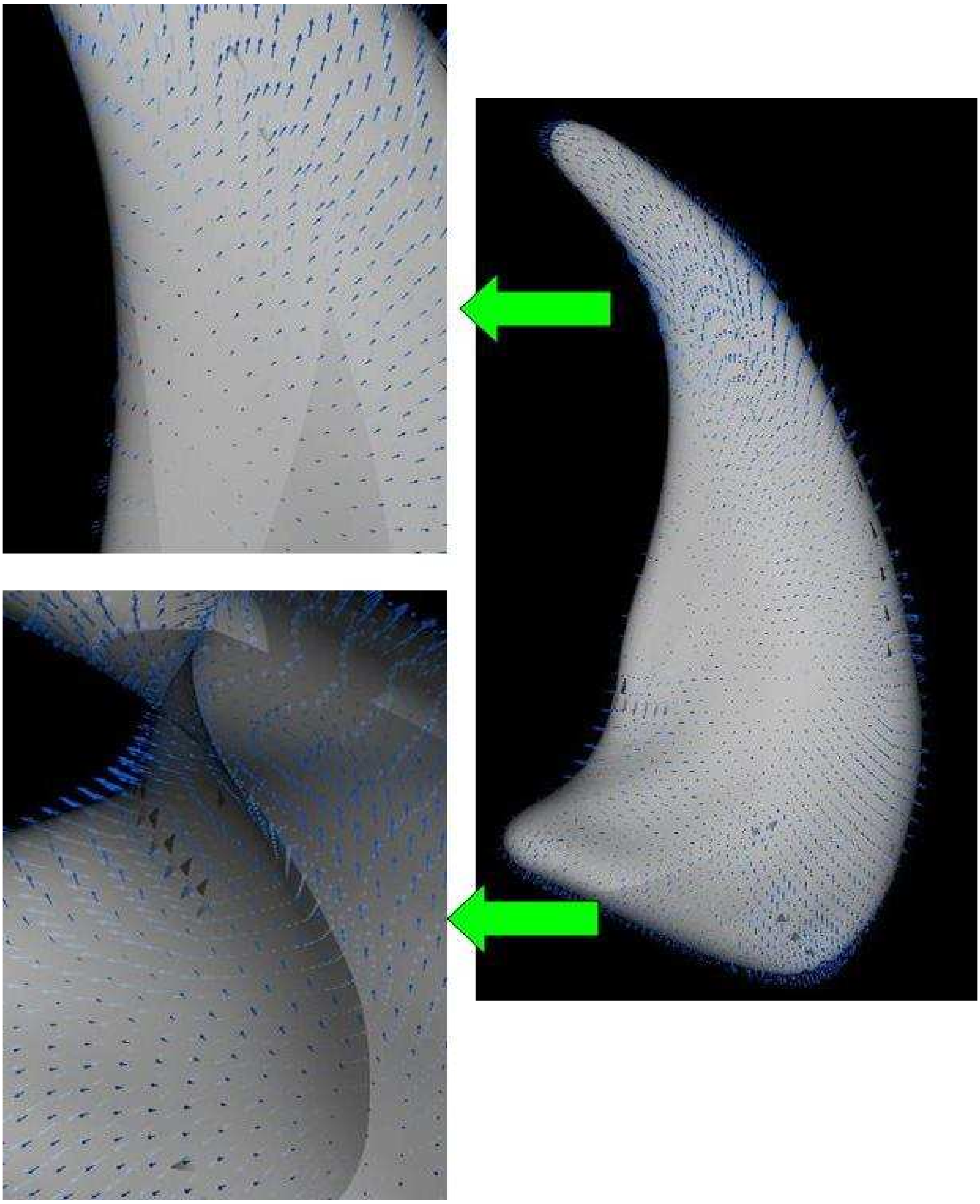


Figure 7.17 An example of the mean shape with the first shape variation vector captured by Active Shape Mode. The magnified images of some areas are also presented on the left. In this example, 4002 nodes are used.

7.4.2 3D Experiments on Artificial Datasets

For validating our proposed MEM algorithm, we first apply our method onto 3D artificial datasets. We are employing our algorithm onto artificial datasets as complementary results, because we are short of large amount of real medical image datasets. The cuboids with different sizes, orientation and length ratios are quite easy to generate in C++ code. Therefore, 20 cuboids are generated automatically by choosing different aspect ratios, orientation and size. In Figure 7.18, we show part of this artificial dataset. There are in total 1002 nodes on each data to represent one cube surface, 642 landmarks are used for optimization. As has been discussed previously, we first find the direct Conformal Mapping between cuboids and unit sphere. We will show the conformal mapping results on cuboids by using both coloured cuboids and unit spheres in Figure 7.19. From Figure 7.19, it can be seen that datasets are mapped to unit spheres successfully. In Figure 7.20, it can be seen that the corners on cuboids with 90° angle are very well preserved in Conformal Mapping. In the Figure 7.21, some of the datasets, which have been processed by MEM are shown. Again, the correspondence points can be identified by using unique colour.

The next step of our experiment is to compare the proposed MEM to a model built using the MDL without one master example, which is arguably the best published approach to defining the correspondence between sets of closed surface. This comparison is based on the results of Generalization Ability, Specificity, and Compactness and is presented in Figure 7.22. From these results, we can see that MEM achieved better Generalization Ability, Specificity and Similar Compactness.

The whole program is coded in C++ and run on a 1G CPU, 512M RAM laptop with platform VC 2005. The total computation time is 1 hour and 4 minutes for MEM and 3 hours 23 minutes for MDL. We attribute the computation efficiency of MEM to the usage of gradient-based optimization strategy in MEM. Another interesting observation is made when we re-construct our shape directly from the processed datasets and neglect the original shapes; we found that both MDL and MEM did not use corners as a one of the correspondence points. This can be seen in Figure 7.23. The reasons for this effect are complicated. We attribute this problem to both the ASM and finding correspondence as a learning process. ASM inherently introduces some simplifications and assumptions to shape variations, e.g. shape variations are composed by a linear combination of variations. Actually, in some case, e.g. what was observed in artificial 3D cuboids, corners or cuboids are nonlinearly corresponded to each other. Moreover, automatic correspondence finding methods, such as MDL and MEM find correspondence in a learning process. The correspondence is achieved by optimizing a pre-specified cost-function and the cost-function is related to correspondence point's statistics. From this scheme, it can be seen that, the correspondence does not necessarily relay on places with high curvatures such as corners. The only promise that MEM and MDL are trying to make is that final converged points are corresponding to each other.

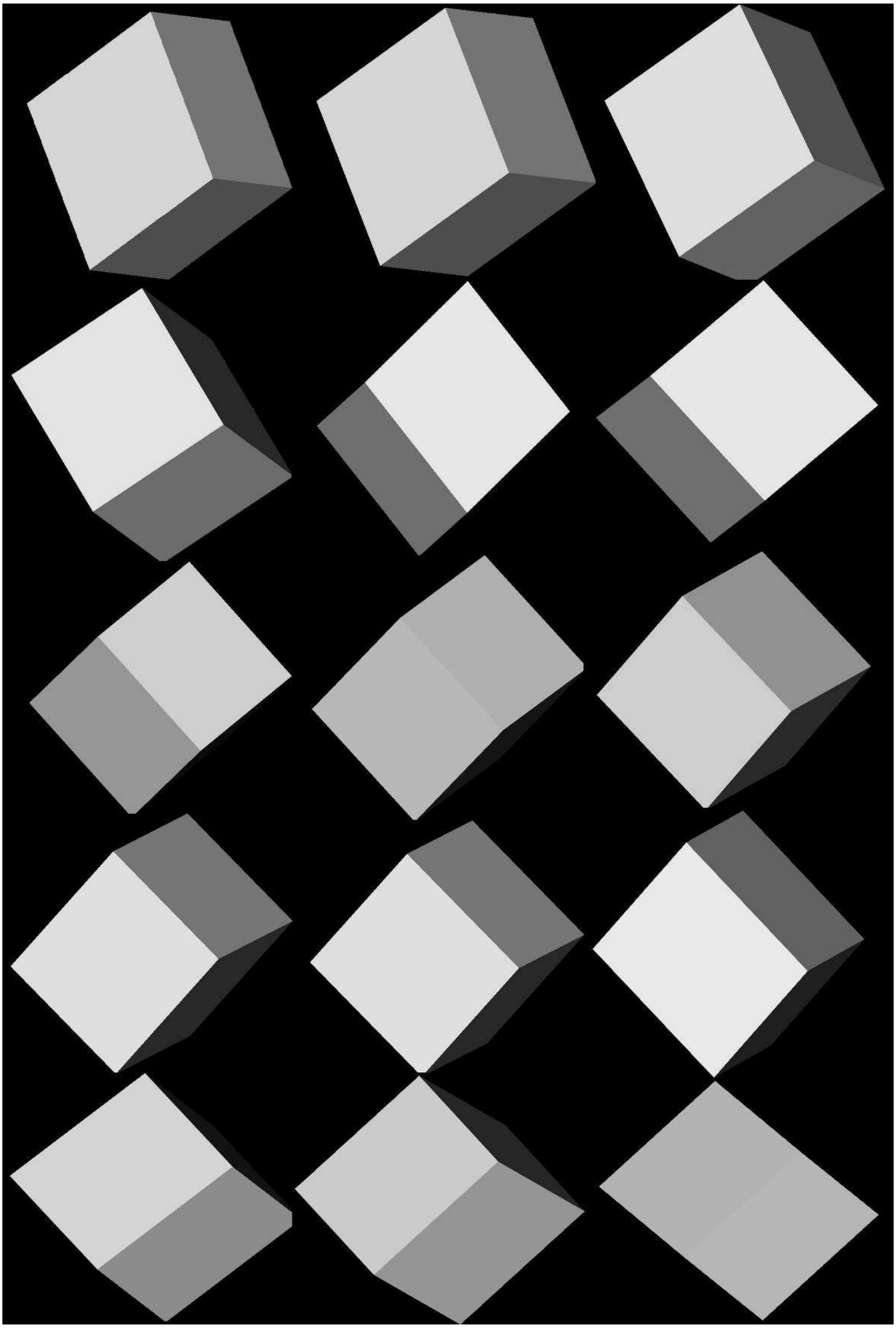


Figure 7.18 Parts of the 3D artificial datasets are shown here. It can be seen that each cube has different aspect ratio and orientation.

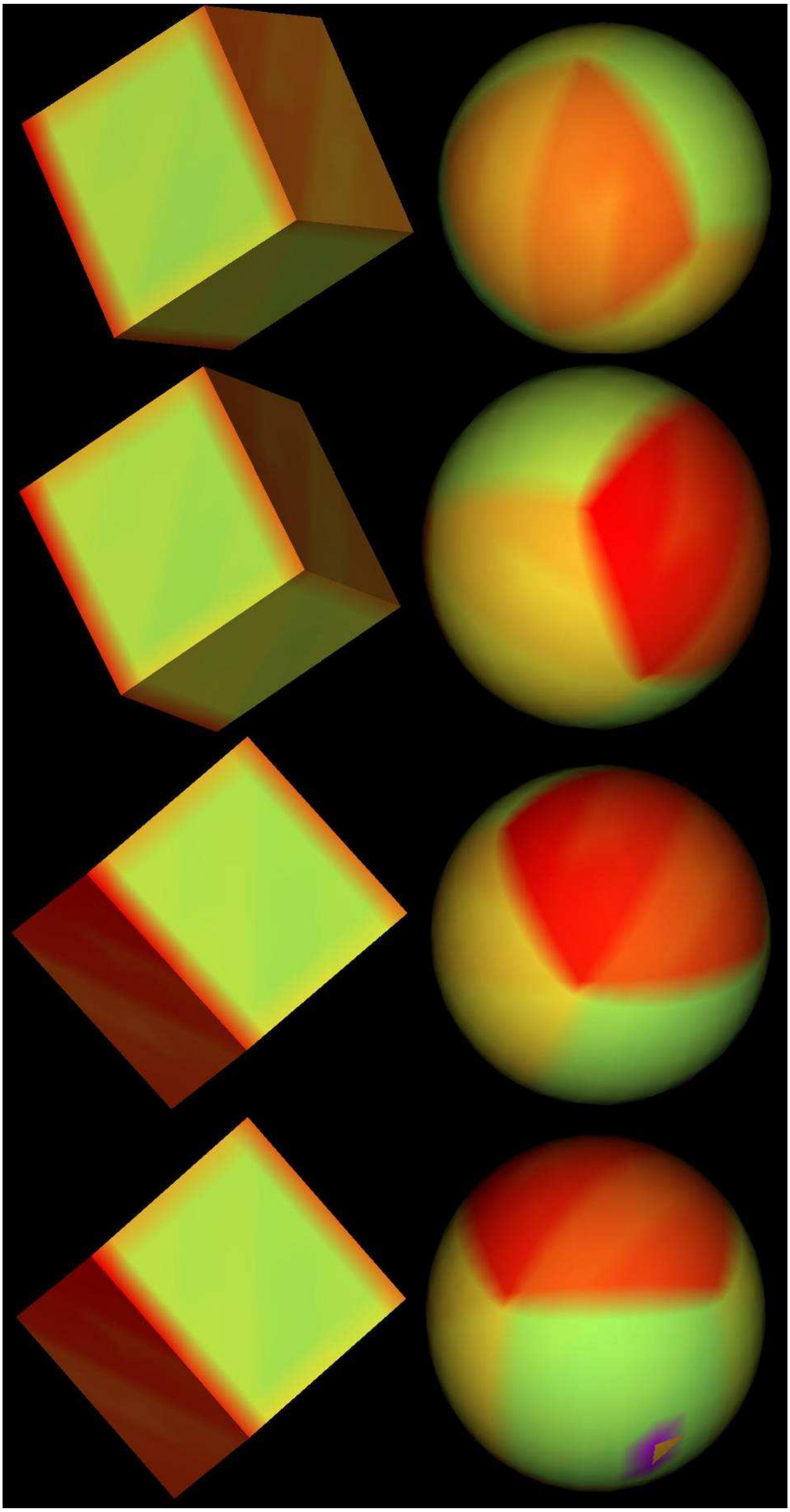


Figure 7.19 Left: Original cuboids with colour. Right: Unit sphere, with corresponding colour.

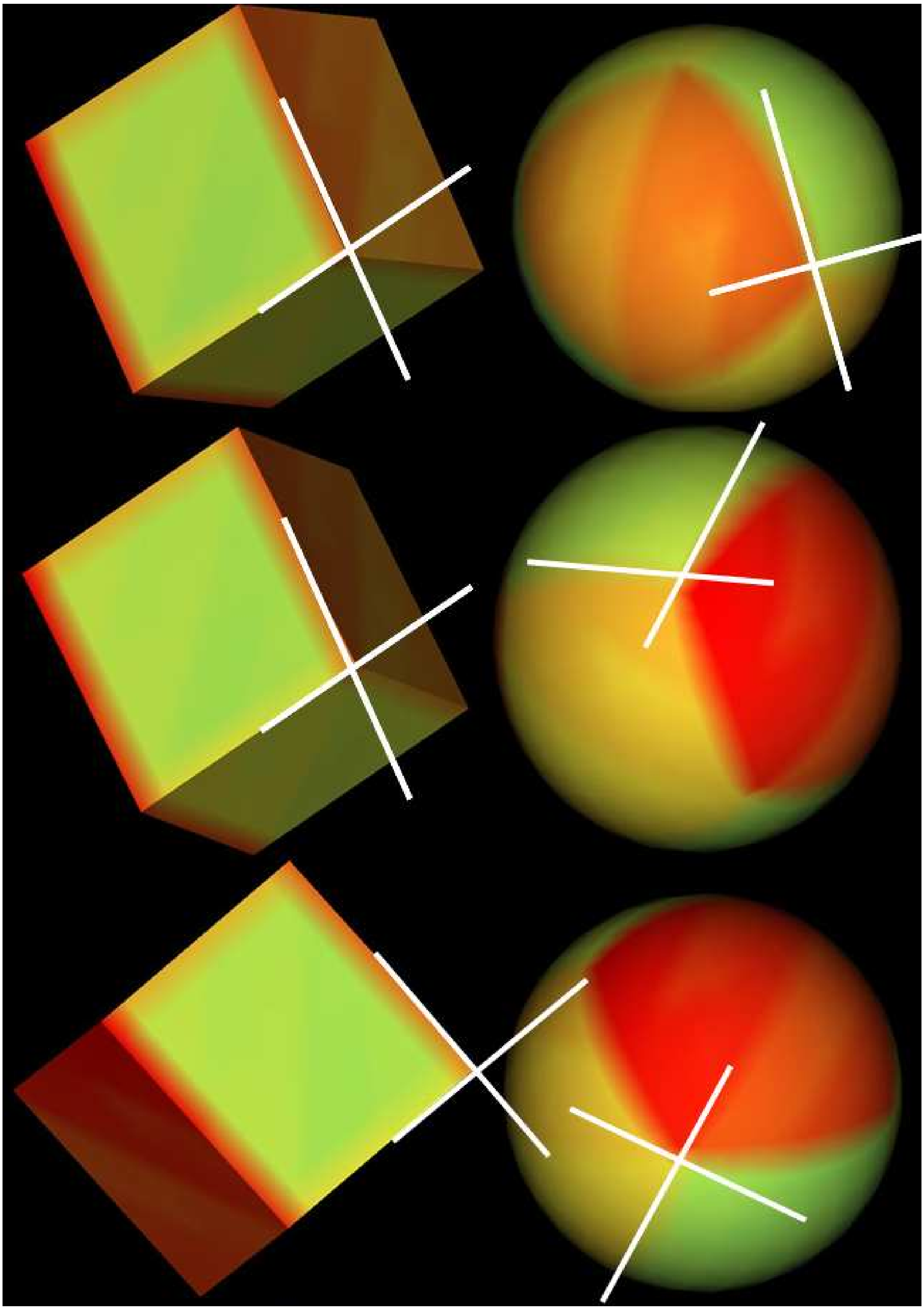


Figure 7.20 A demonstration of angle preserving during shape mapping. The content is same with Figure 7.18, white cross identifies the corner at the cuboids and their conformal mappings. This graph demonstrates Conformal Mapping's ability to preserve angles during shape mapping.

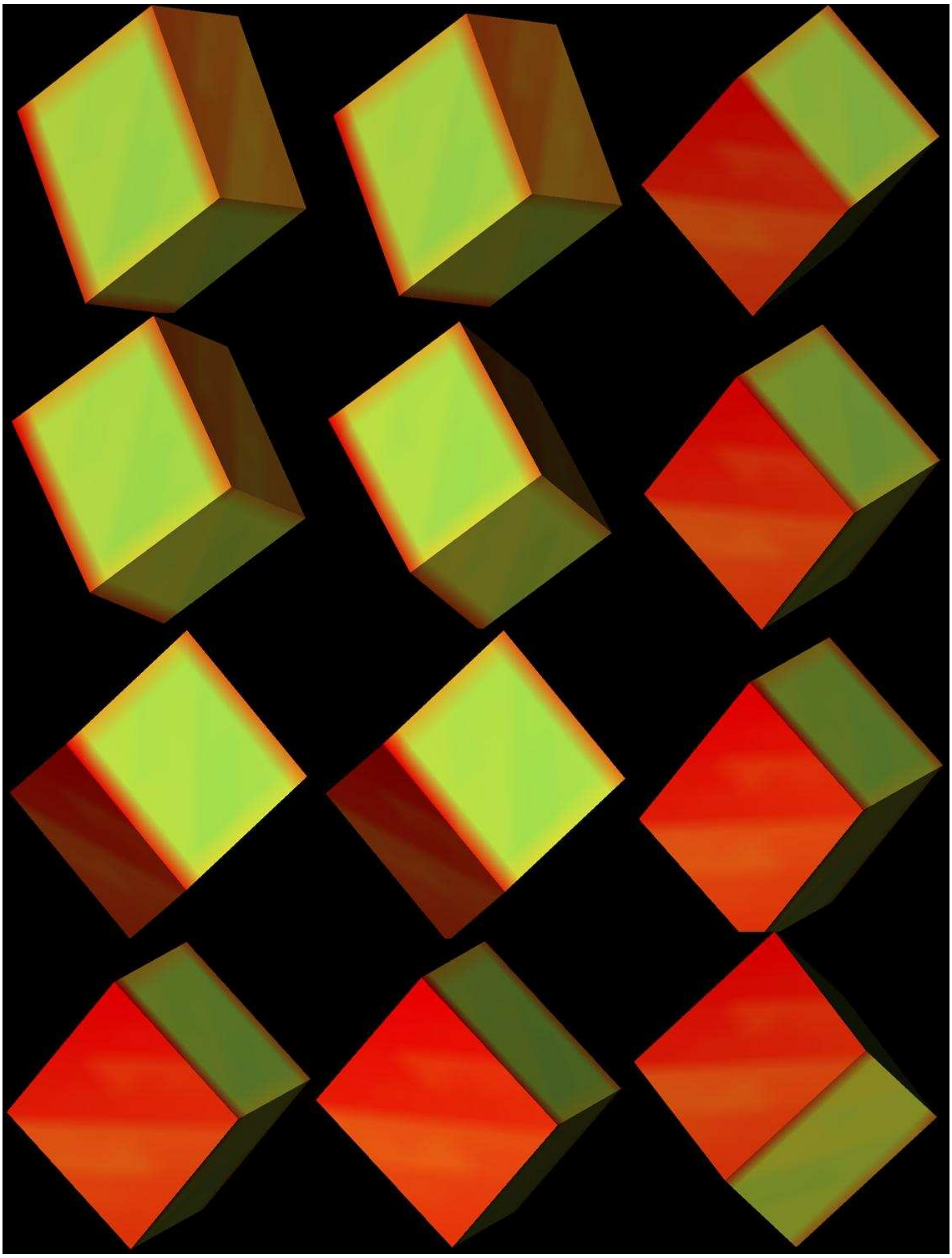


Figure 7.21 MEM results of 12 out of 20 cuboids are shown in this figure, correspondence points are identified with the same colour.

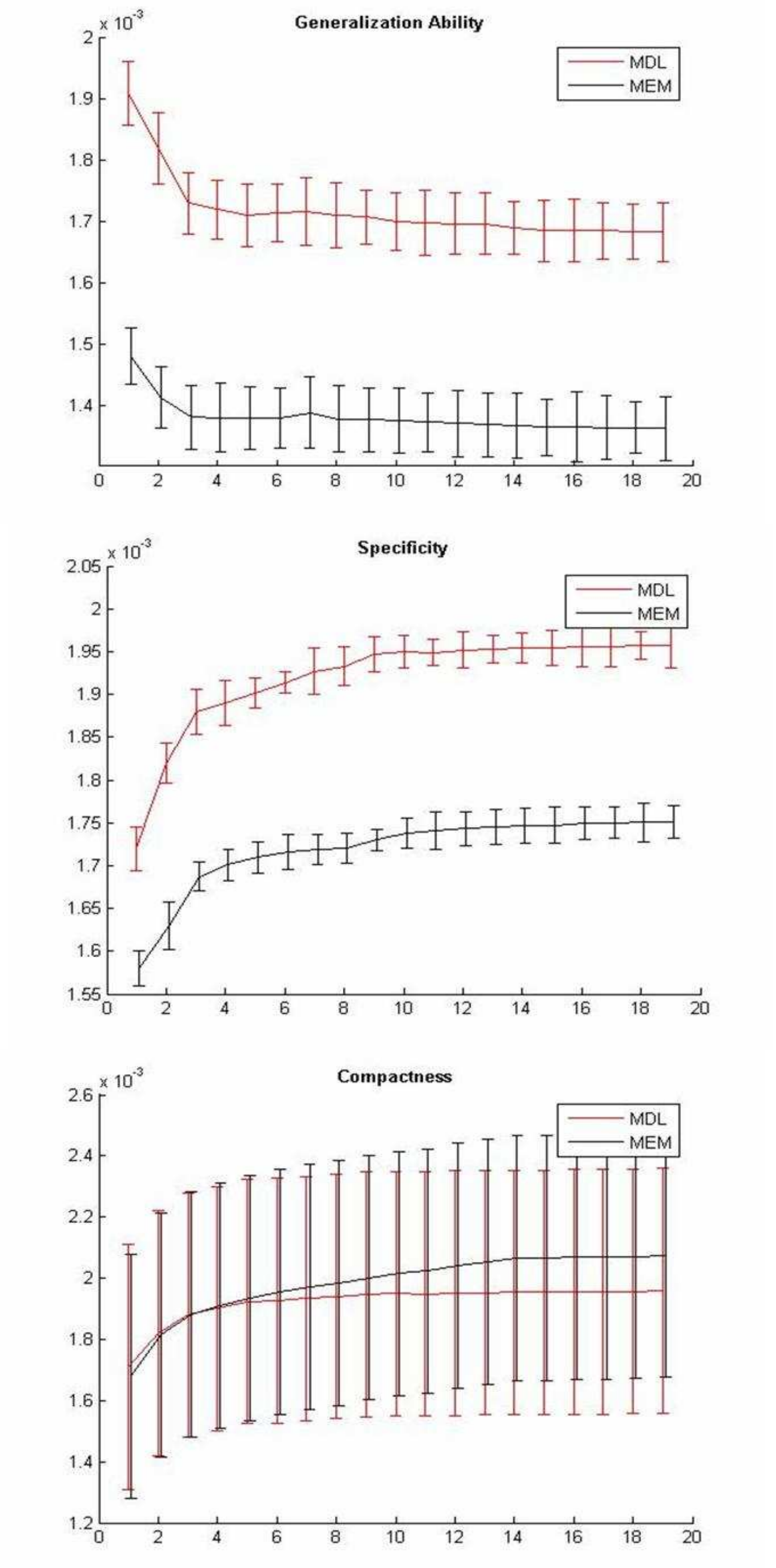


Figure 7.22 From top to bottom: Generalization Ability, Specificity, and Compactness. X-axis represents number of modes used in optimization Y-axis represents corresponding comparison score.

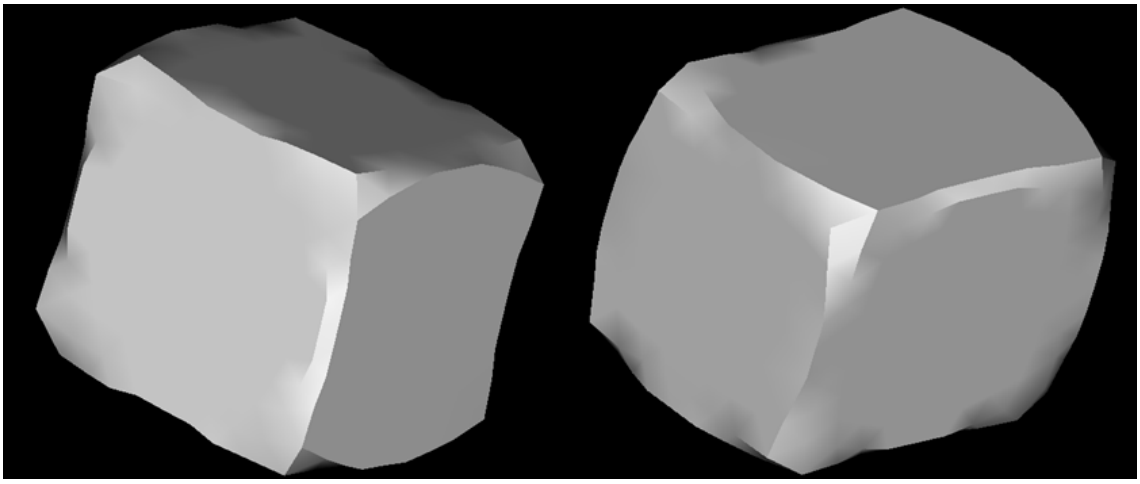


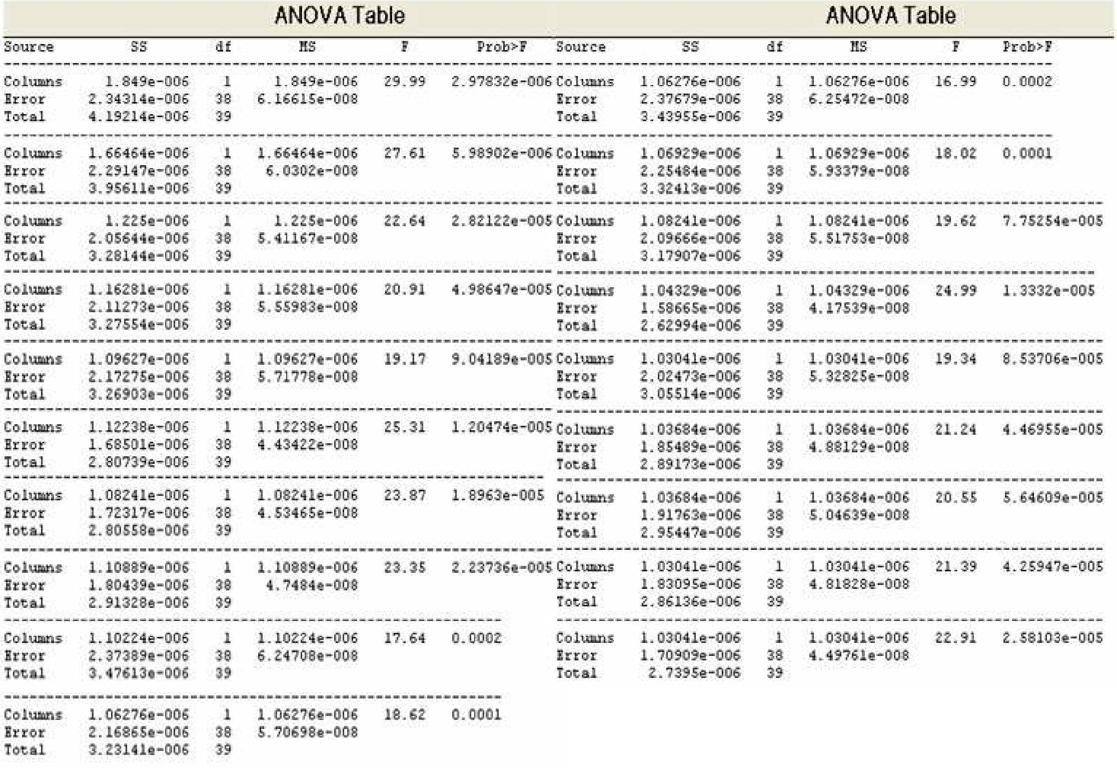
Figure 7.23 Left is correspondence found by MEM and right is from MDL results.

		MEM	MDL	Percentage Difference (%)
Generalization Ability	Mean – Standard Deviation	**23.81355932**	29.90227222	22.67008711
	Mean	**24.7440678**	30.7763452	21.72994428
	Mean + Standard Deviation	**25.67457627**	31.65041818	20.84899253
Specificity	Mean – Standard Deviation	**30.58375304**	34.10951863	10.89994521
	Mean	**30.92185346**	34.47968029	10.87994922
	Mean + Standard Deviation	**31.25995388**	34.84984196	10.86038168
Compactness	Mean – Standard Deviation	28.75609433	**27.5889462**	4.14286017
	Mean	35.80726603	**34.75246868**	2.98979964
	Mean + Standard Deviation	42.85843773	**41.91599116**	2.223421816

Table 7.1. A quantitative analysis on the three criteria comparisons based on Area Under the Curve. The smaller corresponding value is marked in bold character.

We perform the Area Under the Curve (AUC) to quantitatively calculate the differences of MEM vs. MDL using the three criteria. The AUC value is calculated for each criteria of mean value, mean value minus standard deviation and mean value plus standard deviation. The percentage difference is calculated as $\frac{|MDL-MEM|}{MDL+MEM}\times 200\%$.

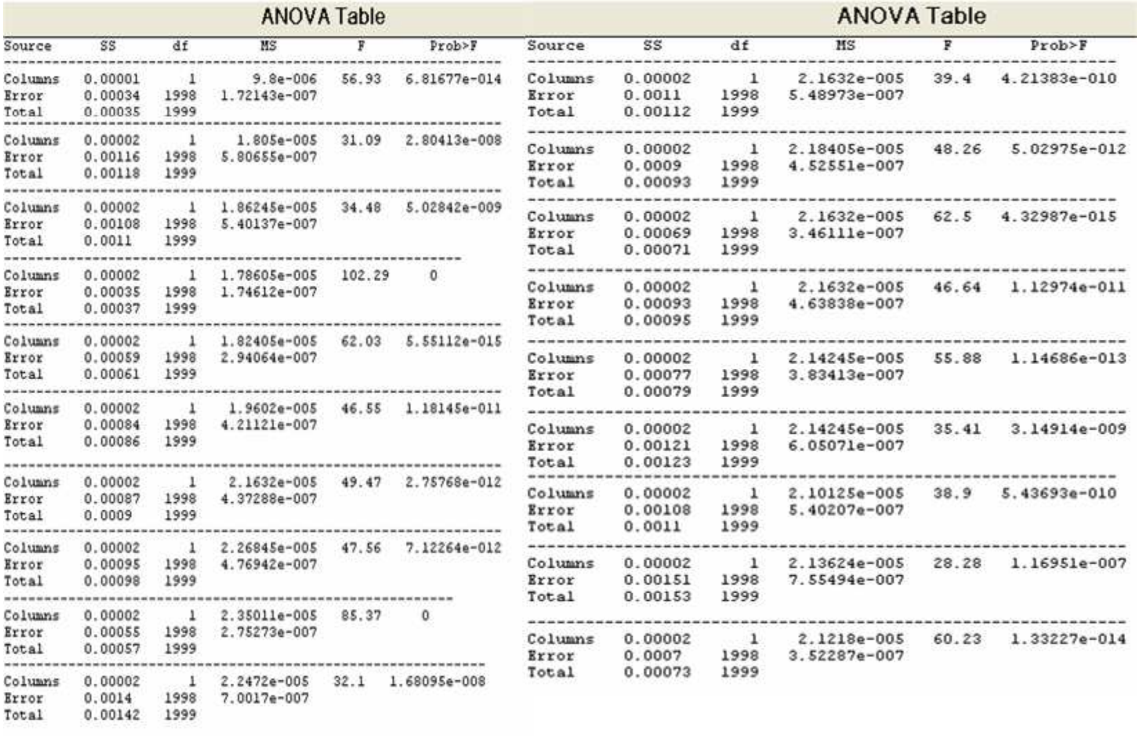
The results of AUC are presented in Table 7.1, where smaller values are made in bold characters. It can be seen that for Generalization Ability, MEM is better than MDL from 20.8% to 22.7%, for Specificity, MEM is better than MDL about 10.9% and MEM is worse than MDL from 2.2% to 4.1% in Compactness. Again, we use ANOVA table to test if results from MDL and MEM are from the same distribution. Within all parameters *p*-value is what we concern. If the *p*-value is near to zero, which means a strong indication that the two groups are from different distributions. From Tables 7.2, and 7.3, we can see that most of the *p*-values are zero or close to zero, therefore the statistical test rejects the hypothesis that samples are from the same mean. The MDL and MEM comparisons on Generalization Ability and Specificity are statistically different.



```
                          ANOVA Table
Source        SS         df       MS           F       Prob>F
-----------------------------------------------------------------
Columns     1.849e-006    1     1.849e-006   29.99   2.97832e-006
Error     2.34314e-006   38   6.16615e-008
Total     4.19214e-006   39
-----------------------------------------------------------------
Columns   1.66464e-006    1   1.66464e-006   27.61   5.98902e-006
Error     2.29147e-006   38     6.0302e-008
Total     3.95611e-006   39
-----------------------------------------------------------------
Columns     1.225e-006    1     1.225e-006   22.64   2.82122e-005
Error     2.05644e-006   38   5.41167e-008
Total     3.28144e-006   39
-----------------------------------------------------------------
Columns   1.16281e-006    1   1.16281e-006   20.91   4.98647e-005
Error     2.11273e-006   38   5.55983e-008
Total     3.27554e-006   39
-----------------------------------------------------------------
Columns   1.09627e-006    1   1.09627e-006   19.17   9.04189e-005
Error     2.17275e-006   38   5.71778e-008
Total     3.26903e-006   39
-----------------------------------------------------------------
Columns   1.12238e-006    1   1.12238e-006   25.31   1.20474e-005
Error     1.68501e-006   38   4.43422e-008
Total     2.80739e-006   39
---------------------------------------------------------------
Columns   1.08241e-006    1   1.08241e-006   23.87   1.8963e-005
Error     1.72317e-006   38   4.53465e-008
Total     2.80558e-006   39
-----------------------------------------------------------------
Columns   1.10889e-006    1   1.10889e-006   23.35   2.23736e-005
Error     1.80439e-006   38     4.7484e-008
Total     2.91328e-006   39
-----------------------------------------------------------
Columns   1.10224e-006    1   1.10224e-006   17.64   0.0002
Error     2.37389e-006   38   6.24708e-008
Total     3.47613e-006   39

-----------------------------------------------------------
Columns   1.06276e-006    1   1.06276e-006   18.62   0.0001
Error     2.16865e-006   38   5.70698e-008
Total     3.23141e-006   39
```

```
                          ANOVA Table
Source        SS         df       MS           F       Prob>F
-----------------------------------------------------------
Columns   1.06276e-006    1   1.06276e-006   16.99   0.0002
Error     2.37679e-006   38   6.25472e-008
Total     3.43955e-006   39
-----------------------------------------------------------
Columns   1.06929e-006    1   1.06929e-006   18.02   0.0001
Error     2.25484e-006   38   5.93379e-008
Total     3.32413e-006   39
-----------------------------------------------------------------
Columns   1.08241e-006    1   1.08241e-006   19.62   7.75254e-005
Error     2.09666e-006   38   5.51753e-008
Total     3.17907e-006   39
---------------------------------------------------------------
Columns   1.04329e-006    1   1.04329e-006   24.99   1.3332e-005
Error     1.58665e-006   38   4.17539e-008
Total     2.62994e-006   39
-----------------------------------------------------------------
Columns   1.03041e-006    1   1.03041e-006   19.34   8.53706e-005
Error     2.02473e-006   38   5.32825e-008
Total     3.05514e-006   39
-----------------------------------------------------------------
Columns   1.03684e-006    1   1.03684e-006   21.24   4.46955e-005
Error     1.85489e-006   38   4.88129e-008
Total     2.89173e-006   39
-----------------------------------------------------------------
Columns   1.03684e-006    1   1.03684e-006   20.55   5.64609e-005
Error     1.91763e-006   38   5.04639e-008
Total     2.95447e-006   39
-----------------------------------------------------------------
Columns   1.03041e-006    1   1.03041e-006   21.39   4.25947e-005
Error     1.83095e-006   38   4.81828e-008
Total     2.86136e-006   39
-----------------------------------------------------------------
Columns   1.03041e-006    1   1.03041e-006   22.91   2.58103e-005
Error     1.70909e-006   38   4.49761e-008
Total      2.7395e-006   39
```

Table 7.2 ANOVA table of the Generalization Ability on datasets of 3D Cuboids.



```
                          ANOVA Table
Source      SS       df        MS            F       Prob>F
------------------------------------------------------------------
Columns   0.00001      1      9.8e-006    56.93   6.81677e-014
Error     0.00034   1998   1.72143e-007
Total     0.00035   1999
------------------------------------------------------------------
Columns   0.00002      1     1.805e-005   31.09   2.80413e-008
Error     0.00116   1998   5.80655e-007
Total     0.00118   1999
------------------------------------------------------------------
Columns   0.00002      1   1.86245e-005   34.48   5.02842e-009
Error     0.00108   1998   5.40137e-007
Total     0.0011    1999
-------------------------------------------------------------
Columns   0.00002      1   1.78605e-005   102.29     0
Error     0.00035   1998   1.74612e-007
Total     0.00037   1999
------------------------------------------------------------------
Columns   0.00002      1   1.82405e-005   62.03   5.55112e-015
Error     0.00059   1998   2.94064e-007
Total     0.00061   1999
------------------------------------------------------------------
Columns   0.00002      1    1.9602e-005   46.55   1.18145e-011
Error     0.00084   1998   4.21121e-007
Total     0.00086   1999

------------------------------------------------------------------
Columns   0.00002      1    2.1632e-005   49.47   2.75768e-012
Error     0.00087   1998   4.37288e-007
Total     0.0009    1999
------------------------------------------------------------------
Columns   0.00002      1   2.26845e-005   47.56   7.12264e-012
Error     0.00095   1998   4.76942e-007
Total     0.00098   1999
------------------------------------------------------------
Columns   0.00002      1   2.35011e-005   85.37     0
Error     0.00055   1998   2.75273e-007
Total     0.00057   1999
----------------------------------------------------------------
Columns   0.00002      1   2.2472e-005   32.1   1.68095e-008
Error     0.0014    1998   7.0017e-007
Total     0.00142   1999
```

```
                          ANOVA Table
Source      SS       df        MS            F       Prob>F
------------------------------------------------------------------
Columns   0.00002      1    2.1632e-005   39.4    4.21383e-010
Error     0.0011    1998   5.48973e-007
Total     0.00112   1999
------------------------------------------------------------------
Columns   0.00002      1   2.18405e-005   48.26   5.02975e-012
Error     0.0009    1998   4.52551e-007
Total     0.00093   1999
------------------------------------------------------------------
Columns   0.00002      1    2.1632e-005   62.5    4.32987e-015
Error     0.00069   1998   3.46111e-007
Total     0.00071   1999

------------------------------------------------------------------
Columns   0.00002      1    2.1632e-005   46.64   1.12974e-011
Error     0.00093   1998   4.63838e-007
Total     0.00095   1999
------------------------------------------------------------------
Columns   0.00002      1   2.14245e-005   55.88   1.14686e-013
Error     0.00077   1998   3.83413e-007
Total     0.00079   1999
------------------------------------------------------------------
Columns   0.00002      1   2.14245e-005   35.41   3.14914e-009
Error     0.00121   1998   6.05071e-007
Total     0.00123   1999
----------------------------------------------------------------
Columns   0.00002      1   2.10125e-005   38.9   5.43693e-010
Error     0.00108   1998   5.40207e-007
Total     0.0011    1999
------------------------------------------------------------------
Columns   0.00002      1   2.13624e-005   28.28   1.16951e-007
Error     0.00151   1998   7.55494e-007
Total     0.00153   1999

------------------------------------------------------------------
Columns   0.00002      1    2.1218e-005   60.23   1.33227e-014
Error     0.0007    1998   3.52287e-007
Total     0.00073   1999
```

Table 7.3 ANOVA table of the Specificity on datasets of 3D Cuboids.

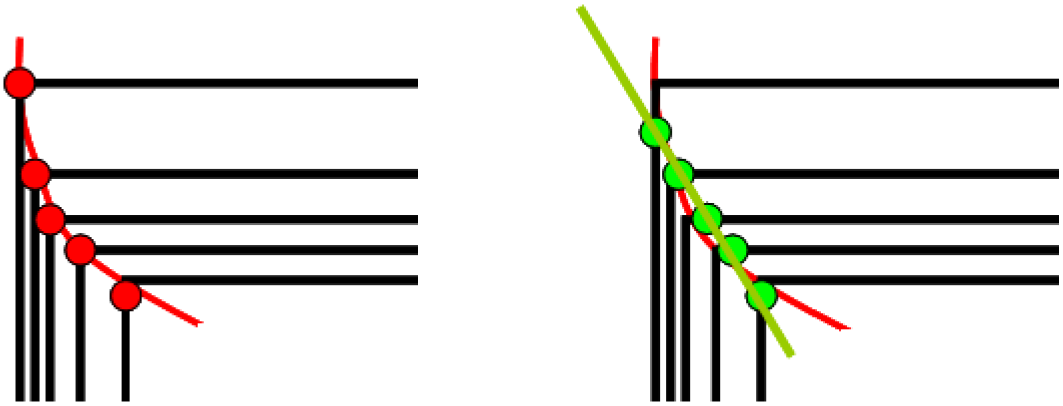


Figure 7.24 A demonstration of linear and nonlinear shape variations. Left: shapes with captured nonlinear variations (Red); Right: shapes with linear variations (Green).

The reasons for correspondence points not lying on the corners are quite complicated. One of the reasons that correspondence points do not lie on the corners can be attributed to the feature extraction method in ASM. In ASM, shape variations are assumed to be a combination of linear shape variations. In fact, nonlinear variations may exist in datasets. For example, Figure 7.24 shows an example of nonlinear shape variations in the artificial datasets. We are using linear variations to approximate nonlinear shape variations. This approximation in Active Shape Model may contribute error during correspondence optimization. Moreover, both MDL and MEM, are tools of minimizing the properties of statistics. Direct shape information, for example curvature, is not considered in these two approaches. Therefore, either MEM or MDL does not necessarily relay the correspondence points on the corners.

From the quantitative experiments Figure 7.22, we can observe that MEM is better in the shape properties of Generalization Ability and Specificity. For Compactness properties, MDL and MEM achieves similar scores. In terms of computation time, MEM is more than three times faster; this is due to the steepest gradient optimization method. This is only a simple example of using both MDL and MEM on artificial 3D datasets; we perform one more experiment to demonstrate the algorithms ability on real medical dataset. This experiment will be discussed in the next section.

7.4.3 3D Experiments on Hippocampus

The MEM is also performed on the 3D medical datasets of hippocampus, which are Magnetic Resonance Imaging (MRI) datasets segmented by an expert. The datasets of hippocampus are from Professor Styner [84], after personal communication. These datasets have been processed by Marching Cube algorithm, so binary-segmented hippocampus datasets have become a set of surface points Cartesian coordinates. The total 21 datasets are saved in *.meta* files, which are essentially the same as a combination of our *.pts* and *.fce* with an extra header file for image information. In Appendix, a clip of this *.meta*, *.pts* and *.fce* files is shown for interested readers. In the file *.pts*, all the points on the surface are numbered sequentially, and the location information is assigned to each location number. In the file *.fce*, all the neighbourhood information is provided, for example, it tells us which three points form a triangle face. Parts of the hippocampus and Conformal Mapping results are shown in Figure 7.25 and 7.26, respectively. From Figure 7.25 we can observe that the shape differences are mainly on the tails of hippocampus on both sides. This observation can later be confirmed by the shape eigenvectors captured by Active Shape Model. From Figure 7.26, it can be seen that the mapping between hippocampus and sphere can be identified by the same colour.

The same machine and software platform as in the previous artificial datasets section were used as well. In this the experiment, each dataset is represented by 1002 nodes and 2562 landmarks are used for optimization. Since we are using more landmarks than the number of nodes in the dataset, linear interpolation was used in this experiment. The computation time for MDL is 54 hours and 13 minutes, and 46 hours and 12 minutes for MEM. With the increase of computation complexity, it can be seen that more time is used for optimization. As discussed previously, we perform an experiment to compare the proposed MEM to a model built using the MDL without one master example. Again, this comparison is based on the results of Generalization Ability, Specificity and Compactness measurement. These results are presented in Figure 7.27. From these results, which are similar to the results of artificial datasets section, we can see that MEM is more general and specific than MDL model; MEM has similar Compactness as MDL. As previously, we also perform ANOVA test on Generalization Ability, and Specificity measurements. Generalization Ability results of this statistical test are shown in Table 7.4. From the results, we can conclude that the Generalization Ability scores of MDL and MEM are statistically different. The hypothesis that the two group are drawn from the same distribution is rejected. Results of ANOVA test for Specificity are shown in Table 7.5. The conclusion is the same as Generalization Ability's that the Specificity score from MDL and MEM are statistically different and drawn from two different distribution.

For making visual impression, we show the shape properties of MEM model by presenting the mean shape (Figure 7.28) and the first few variations vectors (Figure 7.29). As discussed earlier in Visualization section, we use two approaches to represent shape variations captured by Active Shape Model, as in Figure 7.29 and 7.30. The first three shape variations account for more than 99% of variations.

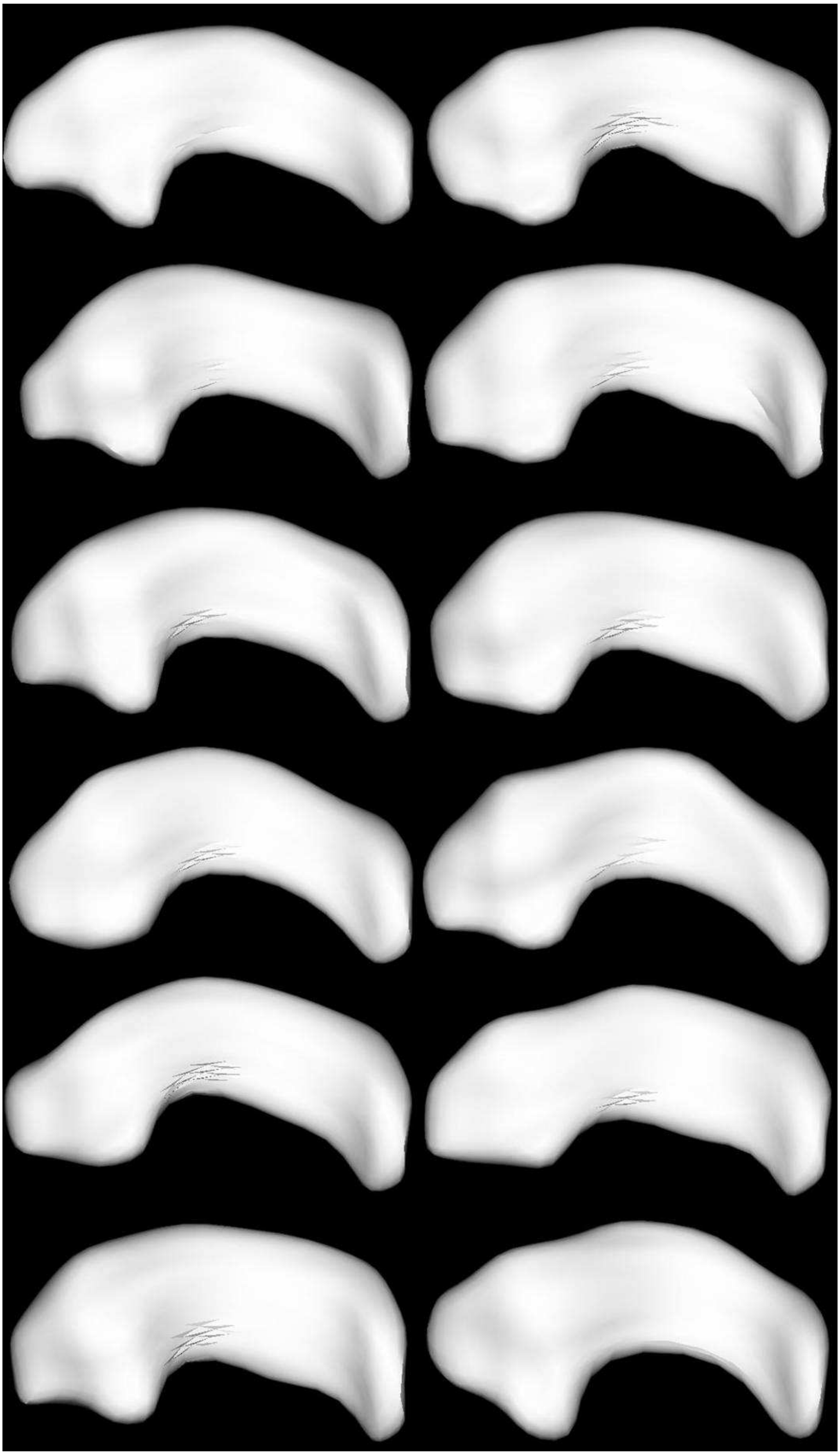


Figure 7.25 12 out of 21 datasets of hippocampus are shown

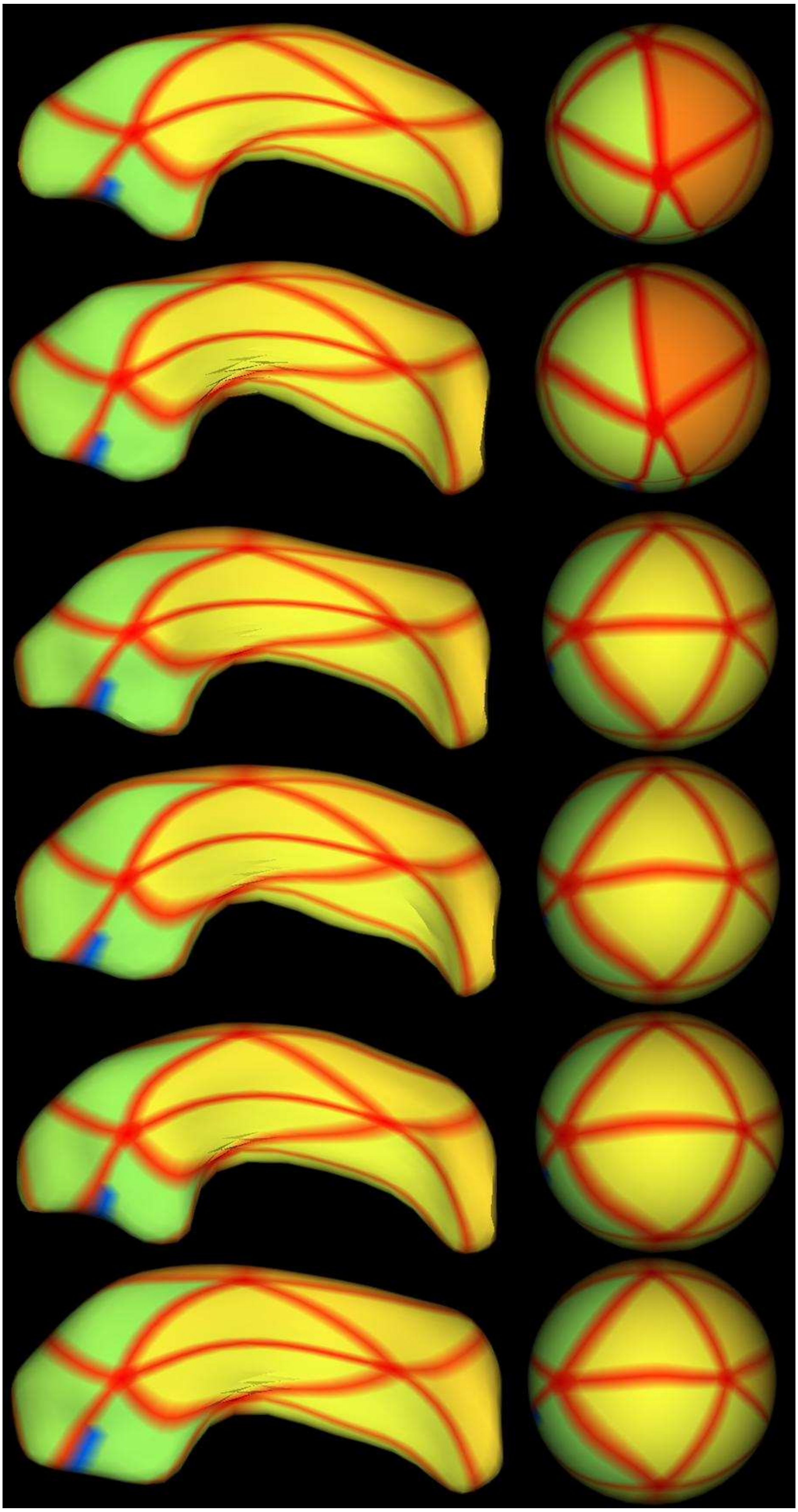


Figure 7.26 6 datasets and Conformal Mapping results are shown. As previously, the datasets are coloured the same for the purpose of visualizing correspondence points.

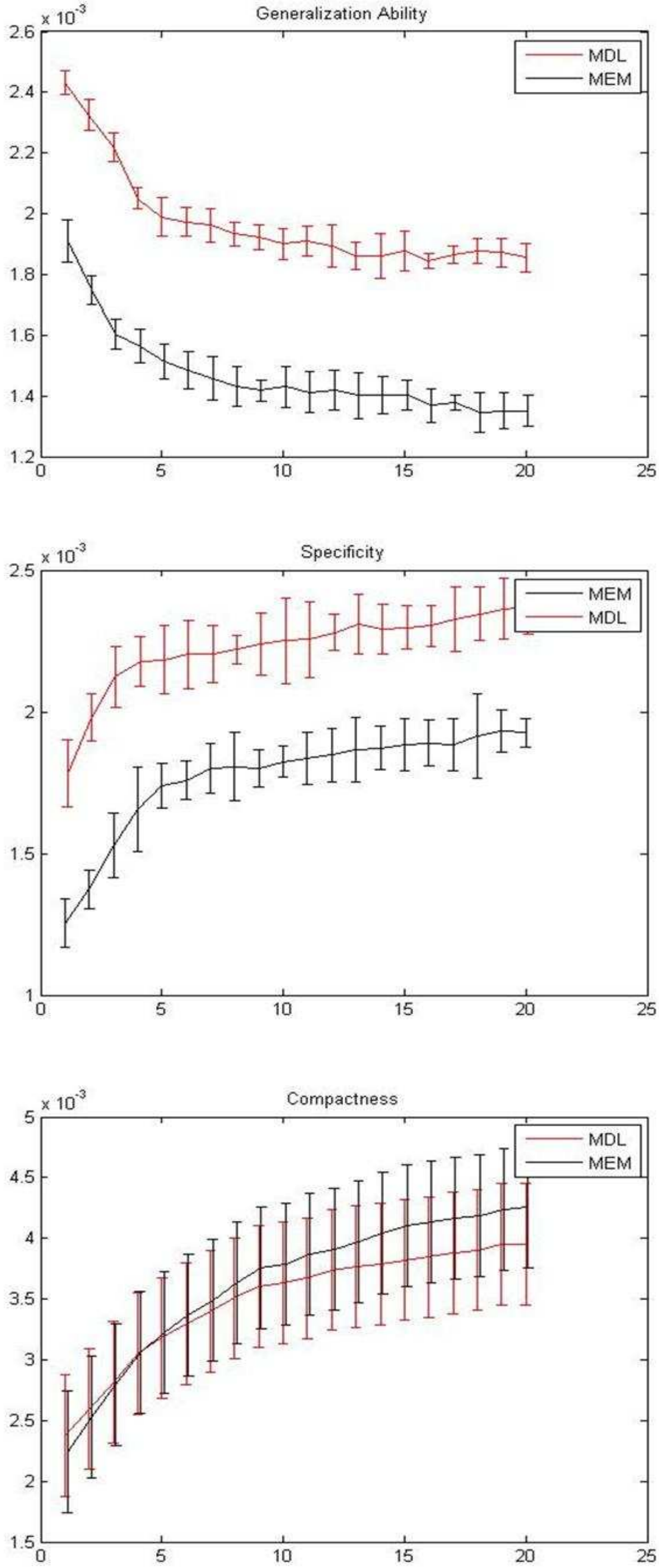
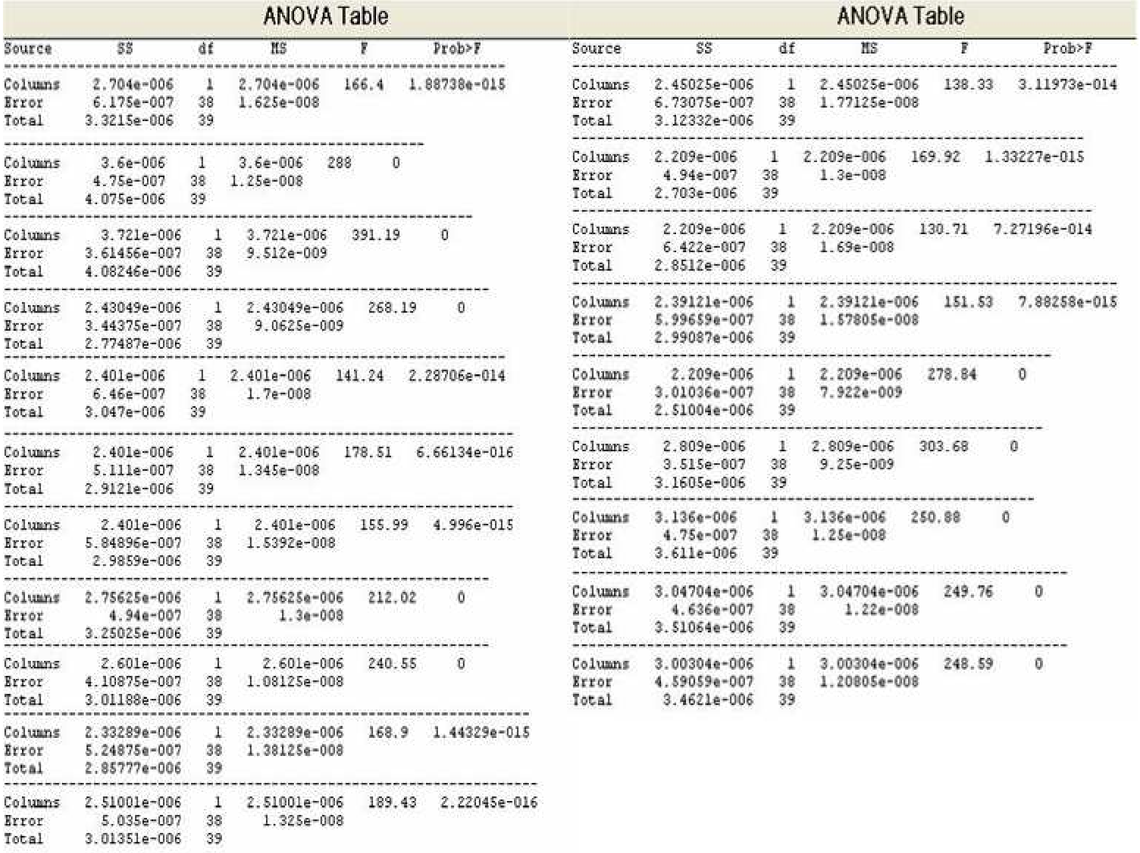


Figure 7.27 Top to bottom: Comparisons of Generalization Ability, Specificity, Compactness metrics on the datasets of hippocampus. X-axis represents number of shape modes used in comparisons and Y-axis represents the correspondence score.



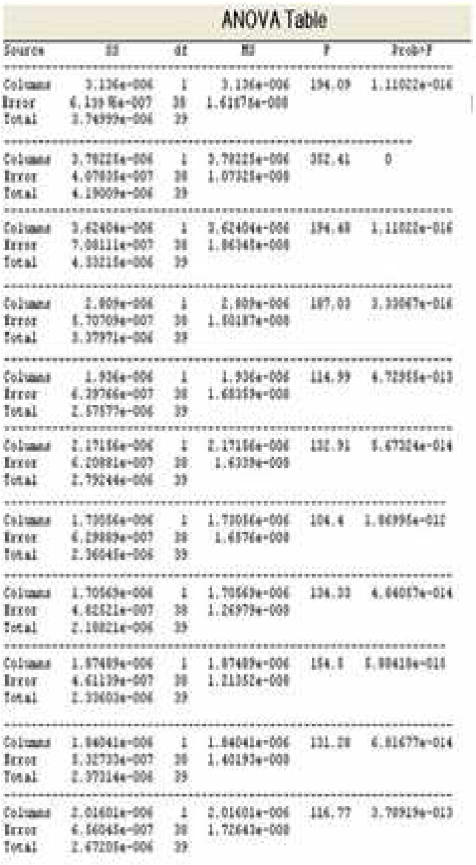
ANOVA Table

```
Source        SS          df       MS            F          Prob>F
--------------------------------------------------------------------
Columns    2.704e-006      1   2.704e-006    166.4    1.88738e-015
Error      6.175e-007     38   1.625e-008
Total      3.3215e-006    39
----------------------------------------------------
Columns      3.6e-006      1     3.6e-006    288       0
Error       4.75e-007     38    1.25e-008
Total      4.075e-006     39
----------------------------------------------------------
Columns     3.721e-006     1   3.721e-006    391.19      0
Error      3.61456e-007   38   9.512e-009
Total      4.08246e-006   39
------------------------------------------------------------
Columns    2.43049e-006    1   2.43049e-006    268.19      0
Error      3.44375e-007   38     9.0625e-009
Total      2.77487e-006   39
--------------------------------------------------------------
Columns    2.401e-006      1   2.401e-006    141.24   2.28706e-014
Error       6.46e-007     38      1.7e-008
Total      3.047e-006     39
---------------------------------------------------------------
Columns     2.401e-006     1   2.401e-006    178.51   6.66134e-016
Error       5.111e-007    38   1.345e-008
Total      2.9121e-006    39
---------------------------------------------------------------
Columns      2.401e-006    1     2.401e-006   155.99   4.996e-015
Error      5.84896e-007   38   1.5392e-008
Total       2.9859e-006   39
-------------------------------------------------------------
Columns    2.75625e-006    1   2.75625e-006   212.02      0
Error         4.94e-007   38        1.3e-008
Total      3.25025e-006   39
-------------------------------------------------------------
Columns      2.601e-006    1     2.601e-006   240.55      0
Error      4.10875e-007   38   1.08125e-008
Total      3.01188e-006   39
-----------------------------------------------------------------
Columns    2.33289e-006    1   2.33289e-006   168.9   1.44329e-015
Error      5.24875e-007   38   1.38125e-008
Total      2.85777e-006   39
------------------------------------------------------------------
Columns    2.51001e-006    1   2.51001e-006   189.43   2.22045e-016
Error        5.035e-007   38      1.325e-008
Total      3.01351e-006   39
```

ANOVA Table

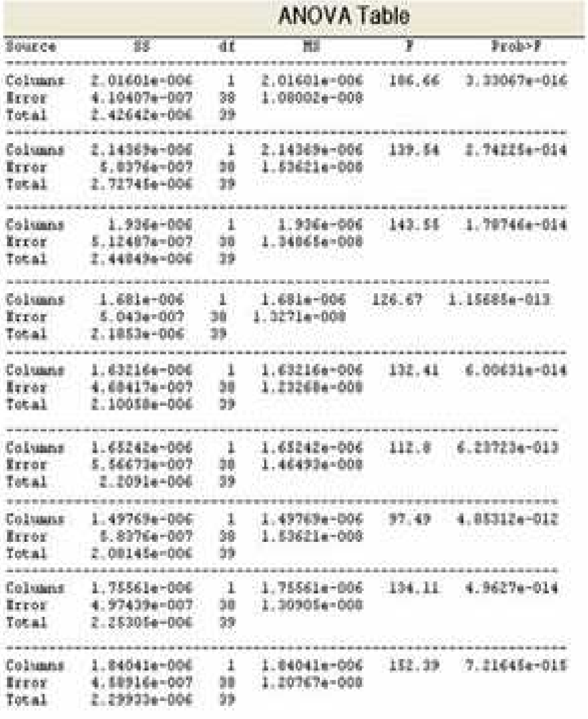
```
Source        SS          df       MS            F          Prob>F
--------------------------------------------------------------------
Columns    2.45025e-006    1   2.45025e-006   138.33   3.11973e-014
Error      6.73075e-007   38   1.77125e-008
Total      3.12332e-006   39
----------------------------------------------------------------
Columns    2.209e-006     1   2.209e-006    169.92   1.33227e-015
Error       4.94e-007    38      1.3e-008
Total      2.703e-006    39
-----------------------------------------------------------------
Columns     2.209e-006    1    2.209e-006    130.71   7.27196e-014
Error       6.422e-007   38    1.69e-008
Total      2.8512e-006   39
--------------------------------------------------------------------
Columns    2.39121e-006    1   2.39121e-006   151.53   7.88258e-015
Error      5.99659e-007   38   1.57805e-008
Total      2.99087e-006   39
-----------------------------------------------------------
Columns      2.209e-006    1   2.209e-006    278.84      0
Error      3.01036e-007   38   7.922e-009
Total      2.51004e-006   39
---------------------------------------------------------
Columns     2.809e-006    1   2.809e-006    303.68      0
Error       3.515e-007   38     9.25e-009
Total      3.1605e-006   39
--------------------------------------------------------
Columns    3.136e-006     1   3.136e-006    250.88      0
Error       4.75e-007    38    1.25e-008
Total      3.611e-006    39
-------------------------------------------------------------
Columns    3.04704e-006    1   3.04704e-006    249.76      0
Error        4.636e-007   38       1.22e-008
Total      3.51064e-006   39
-------------------------------------------------------------
Columns    3.00304e-006    1   3.00304e-006    248.59      0
Error      4.59059e-007   38   1.20805e-008
Total       3.4621e-006   39
```

Table 7.4 ANOVA table of the Generalization Ability on datasets of 3D hippocampus.



ANOVA Table

```
Source        SS          df       MS            F          Prob>F
--------------------------------------------------------------------
Columns      3.136e-006    1     3.136e-006   194.09   1.11022e-016
Error      6.13916e-007   38   1.61578e-008
Total      3.74999e-006   39
-----------------------------------------------------------
Columns    3.78225e-006    1   3.78225e-006   352.41      0
Error      4.07835e-007   38   1.07325e-008
Total      4.19009e-006   39
--------------------------------------------------------------------
Columns    3.62404e-006    1   3.62404e-006   194.48   1.11022e-016
Error      7.08111e-007   38   1.86345e-008
Total      4.33215e-006   39
--------------------------------------------------------------------
Columns      2.809e-006    1     2.809e-006   187.03   3.33067e-016
Error      5.70709e-007   38   1.50187e-008
Total      3.37971e-006   39
--------------------------------------------------------------------
Columns      1.936e-006    1     1.936e-006   114.99   4.72955e-013
Error      6.39766e-007   38   1.68359e-008
Total      2.57577e-006   39
--------------------------------------------------------------------
Columns    2.17156e-006    1   2.17156e-006   132.91   5.67324e-014
Error      6.20881e-007   38    1.6339e-008
Total      2.79244e-006   39
-------------------------------------------------------------------
Columns    1.73056e-006    1   1.73056e-006   104.4   1.86995e-012
Error      6.29889e-007   38    1.6576e-008
Total      2.36045e-006   39
--------------------------------------------------------------------
Columns    1.70569e-006    1   1.70569e-006   134.33   4.44057e-014
Error      4.82521e-007   38   1.26979e-008
Total      2.18821e-006   39
-------------------------------------------------------------------
Columns    1.87489e-006    1   1.87489e-006   154.1   5.88418e-015
Error      4.61139e-007   38   1.21352e-008
Total      2.33603e-006   39
--------------------------------------------------------------------
Columns    1.84041e-006    1   1.84041e-006   131.28   6.81677e-014
Error      5.32733e-007   38   1.40193e-008
Total      2.37314e-006   39
--------------------------------------------------------------------
Columns    2.01601e-006    1   2.01601e-006   116.77   3.78919e-013
Error      6.56045e-007   38   1.72643e-008
Total      2.67206e-006   39
```



ANOVA Table

```
Source        SS          df       MS            F          Prob>F
--------------------------------------------------------------------
Columns    2.01601e-006    1   2.01601e-006   186.66   3.33067e-016
Error      4.10407e-007   38   1.08002e-008
Total      2.42642e-006   39
--------------------------------------------------------------------
Columns    2.14369e-006    1   2.14369e-006   139.54   2.74225e-014
Error        5.8376e-007  38   1.53621e-008
Total      2.72745e-006   39
--------------------------------------------------------------------
Columns      1.936e-006    1     1.936e-006   143.55   1.78746e-014
Error      5.12487e-007   38   1.34865e-008
Total      2.44849e-006   39
------------------------------------------------------------------
Columns     1.681e-006    1    1.681e-006   126.67   1.15685e-013
Error       5.043e-007   38   1.3271e-008
Total      2.1853e-006   39
--------------------------------------------------------------------
Columns    1.63216e-006    1   1.63216e-006   132.41   6.00631e-014
Error      4.68417e-007   38   1.23268e-008
Total      2.10058e-006   39
-------------------------------------------------------------------
Columns    1.65242e-006    1   1.65242e-006   112.8   6.23723e-013
Error      5.56673e-007   38   1.46493e-008
Total       2.2091e-006   39
-------------------------------------------------------------------
Columns    1.49769e-006    1   1.49769e-006   97.49   4.85312e-012
Error        5.8376e-007  38   1.53621e-008
Total      2.08145e-006   39
-------------------------------------------------------------------
Columns    1.75561e-006    1   1.75561e-006   134.11   4.9627e-014
Error      4.97439e-007   38   1.30905e-008
Total      2.25305e-006   39
--------------------------------------------------------------------
Columns    1.84041e-006    1   1.84041e-006   152.39   7.21645e-015
Error      4.58916e-007   38   1.20767e-008
Total      2.29933e-006   39
```

Table 7.5 ANOVA table of the Specificity on datasets of 3D hippocampus.

From Tables 7.4 and 7.5 we can see that most of the *p*-values are zero or close to zero, therefore ANOVA test showed that the MDL and MEM are different in both Generalization Ability and Specificity.

		MEM	MDL	Percentage Difference (%)
Generalization Ability	Mean – Standard Deviation	**26.56703206**	36.27255895	30.88984744
	Mean	**27.66764923**	37.16663596	29.30235664
	Mean + Standard Deviation	**28.7682664**	38.06071297	27.80963187
Specificity	Mean – Standard Deviation	**32.04142336**	40.5774635	10.89994521
	Mean	**33.79881387**	42.46532847	10.87994922
	Mean + Standard Deviation	**35.55620438**	44.35319343	10.86038168
Compactness	Mean – Standard Deviation	59.98375092	**57.31890199**	4.543544183
	Mean	69.43375092	**66.6853353**	4.03825165
	Mean + Standard Deviation	78.88375092	**76.05176861**	3.655691506

Table 7.6. A quantitative analysis on the three criteria comparisons based on Area Under The Curve. The smaller corresponding value is marked in bold character.

We perform the Area Under the Curve (AUC) to quantitatively calculate the difference between the three criteria, under MEM and MDL. The AUC value is calculated for each criteria of mean value, mean value minus standard deviation and mean value plus standard deviation. The results of AUC are presented in Table 7.6, where smaller values are made in bold characters. It can be seen that for Generalization Ability, MEM is better than MDL from 27.8% to 30.9%, for Specificity, MEM is better than MDL from

10.86% to 10.90% and MEM is worse than MDL from 3.7% to 4.5% in Compactness.

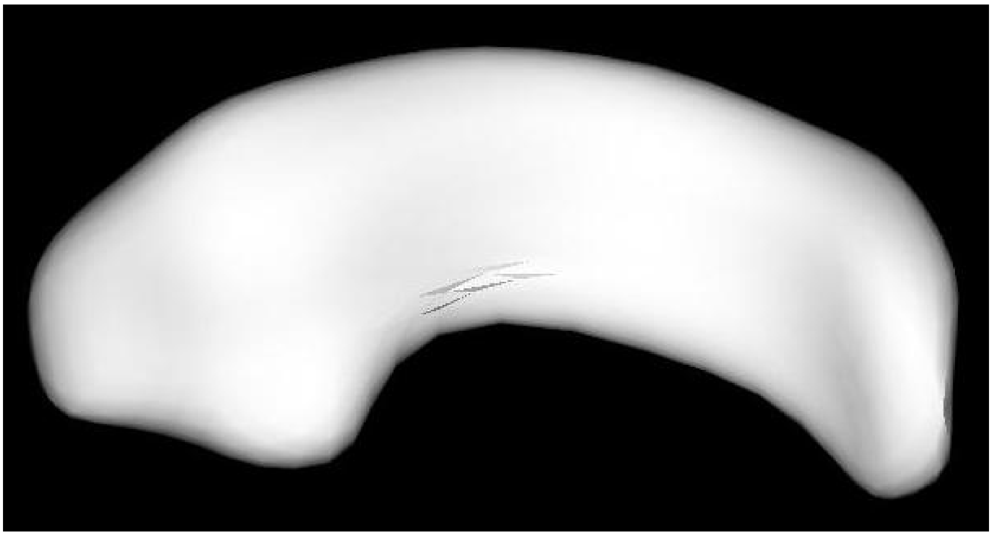


Figure 7.28 This is an example mean shape drawn from 21 training sets.

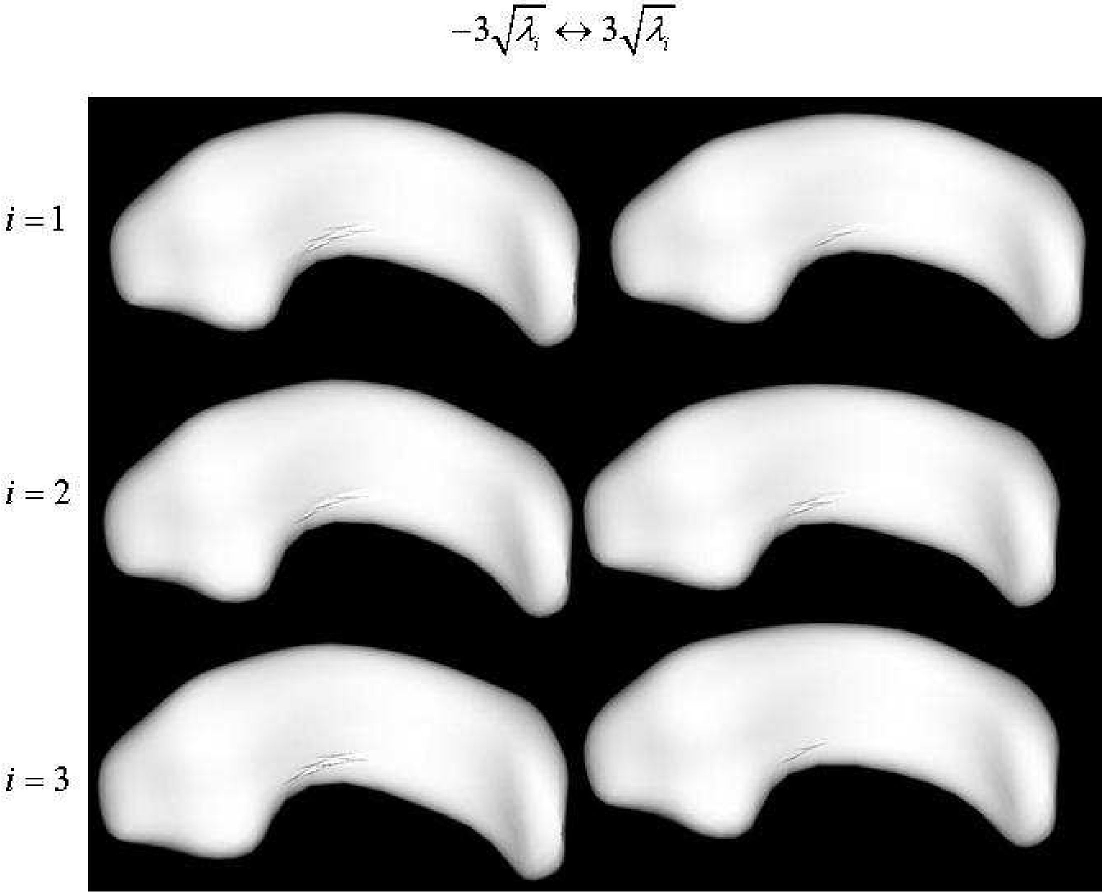


Figure 7.29 The first three modes of the MEM hippocampus model. Some of the shape differences can be seen from the corners (or tails) of hippocampus.

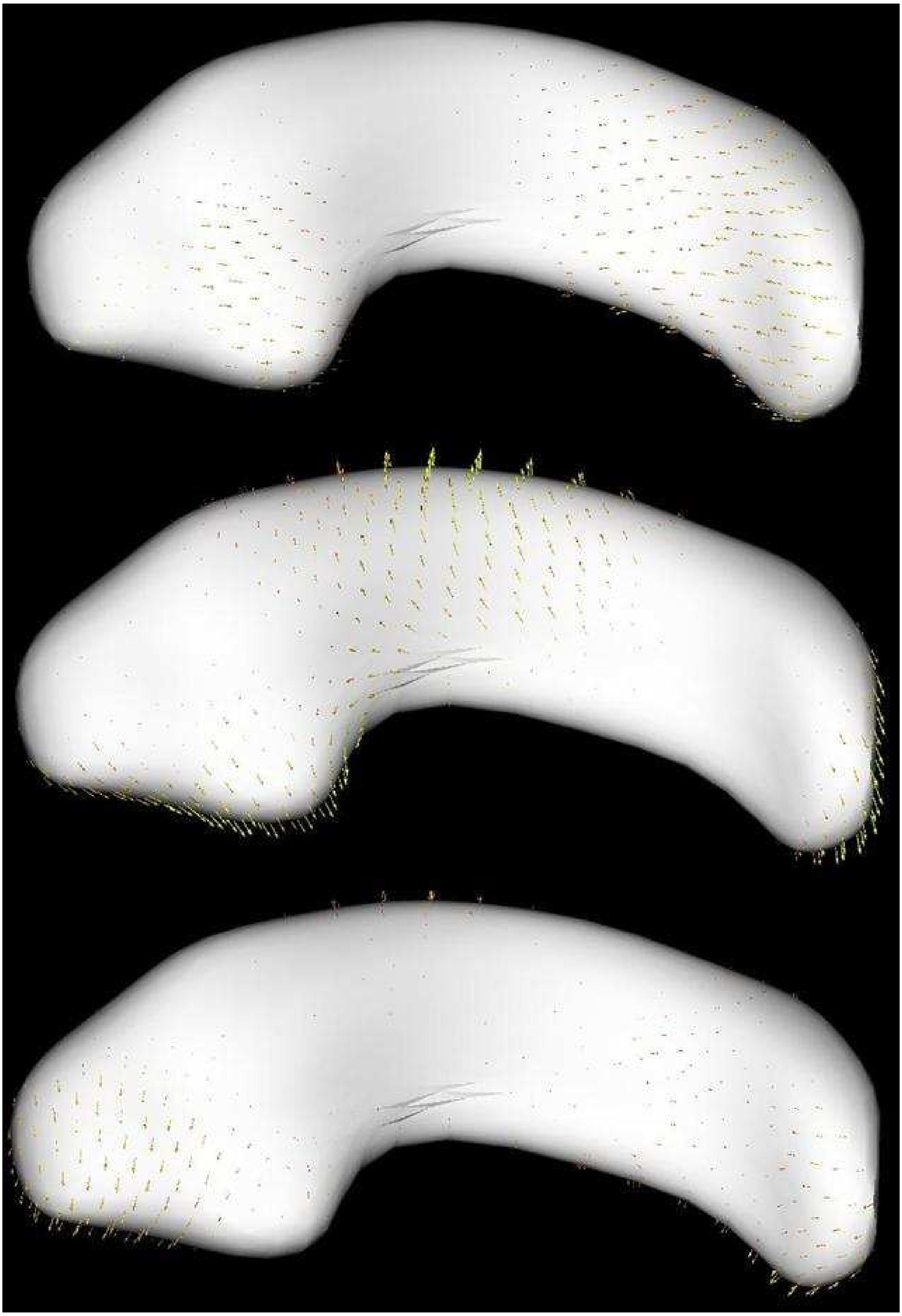


Figure 7.30 The first three modes of the MEM hippocampus model. In this approach, each node is assigned with an arrow pointing the shape variation direction.

In Figure 7.29, shapes are generated by choosing the weighting components (recall Equation 2.8 about the ASM) from $\left[-3\sqrt{\lambda_i}, 3\sqrt{\lambda_i}\right]$. In the graph, three shape variations are shown.

In Figure 7.30, it can be seen that each node is assigned with an arrow, pointing to the direction captured by the Active Shape Model. From Figure 7.30, the results show that shape variations are congested in the left and right side tail of the shape and very small variations in the middle. This observation agrees with our finding from the datasets, which was concluded in the previous paragraph.

7.4.4 Conclusions on 3D Experiments and Discussion.

In this chapter, the limitations of the framework used in 2D has been discussed which demands the introduction of a 3D framework. Therefore, the MEM extension in 3D has been discussed, and the MEM gradient in the 3D case is also presented. For validating the MEM method, we applied the proposed method to datasets of artificial cuboids and hippocampus. From our direct observation, it can be seen that MEM finds the correspondence reasonably accurately in 3D cases. Shape variations captured by the Active Shape Model; agree with what we observed from the training set that most shape variations are congested on both sides of the tail area. Again, for quantitative comparisons, measurements of Generalization Ability, Specificity and Compactness are performed. It can be seen that the quantitative comparisons show significant difference between the proposed MEM and MDL model. The AUC results show that the MEM offers better Generalization Ability and Specificity than MDL on both 3D datasets. Moreover, the MEM has slightly worse Compactness than the MDL. However, Compactness is an application-based criterion. In a recent publication by Davies [65], they only use the criteria of Generalization Ability and Specificity for assessment of shape model properties. According to the experiment, both MDL and MEM can use the first three shape variations to cover more than 90% of all variations, which is more than enough for most of the applications.

The 3D optimization algorithm takes much longer than in the 2D case due to increase of complexity (one more dimension and more nodes on the surface) even when a lower-level language (C++) is used (recall that the 2D code is built in Matlab). In the 3D case, MEM is a bit faster than MDL, due to the refinement of shape parameterization and re-parameterization method. Another interesting observation is that during the experiments on cuboids, corners are not located as corresponding points by both MDL and MEM. We argued that this is caused mainly by two reasons. Firstly, if the variations are nonlinear, linear variations used in ASM will introduce some error. In this case, errors may be represented as missing corners. Secondly and more importantly, as a method of finding corresponding points is a learning process, both MDL and MEM do not necessarily find points, which have distinguishable features such as shape corners with high curvatures.

In the end, we conclude that the proposed MEM can be successfully extended into 3D scenarios.

Chapter 8 Applications of Using MEM & MDL for Classification

In this chapter, we will present, in terms of an application, some results on how to use the proposed automatic correspondence finding algorithm, in order to perform classification on genetic separated subjects. For example, face profiles can be easily separated by their gender. In this chapter, we performed a similar experiment, which was also introduced in [38] by Thodberg. During the experiment, a number of face profile photos were first collected. Manual or automatic face segmentation was performed to extract the face profile contours. These face profiles were used as an input to the automatic correspondence finding algorithm. In this experiment the facial profiles were considered as a 2D open curve with flexible end points problem. Normally, we will first set the number of correspondence points, which should be found during optimization. After optimization, on each set of face profiles, a number of correspondence points are located. An Active Shape Model (ASM) can be easily built from these points. As we recall from previous chapters, each shape profile is identified by concatenating landmark coordinates. Then error minimization is performed on reducing possible errors due to translation, rotation and scaling. After applying Principal Component Analysis to the shape covariance matrix to extract shape variations, each face profile is rewritten as a combination of mean shape and shape variations multiplying weighting vector, since all the profiles are using the same mean shape and the same set of shape variations. For simplicity, each profile can be identified by the weighting vector *b* (please recall Equation 2.8) on the shape variations. Therefore, by using classification methods, such as Logistic Regression [100] on the ASM weighting vector *b*, a gender prediction model can be generated. In this experiment, since the gender of each profile is known, leave-one-out validation can be incorporated to evaluate the performance of the prediction model and the classification accuracy can be easily estimated. In this chapter, comparisons were performed between the classification accuracy from direct human observers, and shape model built from MDL, MEM, and manual landmarked results.

The paragraphs below will introduce the details of methodology used in the experiments, results, and conclusions. We will first start from the preparation of the dataset, then, the details of the classification algorithm. A detailed comparison between the MDL, MEM and manual land marking is shown. In the end, we conclude our experiments.

8.1 Introduction of Datasets

We collected datasets of 131 facial profiles, in which, 64 are male and 67 are female. The datasets are collected by using a Digital Camera to collect pictures of the author's friends and their friends. These people are all Chinese and aged from 20 to 30. From personal communication, they all claim that they did not have relatives from outside China for the last three generations; therefore, we can claim that these datasets belong to subjects of Chinese extraction. As in the experiments in the previous chapters about facial profiles, the shape profiles have to be manually segmented from digital pictures before further processing. We demonstrate the segmentation result by showing two segmented graphs in Figure 8.1. In this figure, a male profile photo and a female profile photo are shown. On them, red lines are manually placed on the profiles in order to extract the facial boundaries.

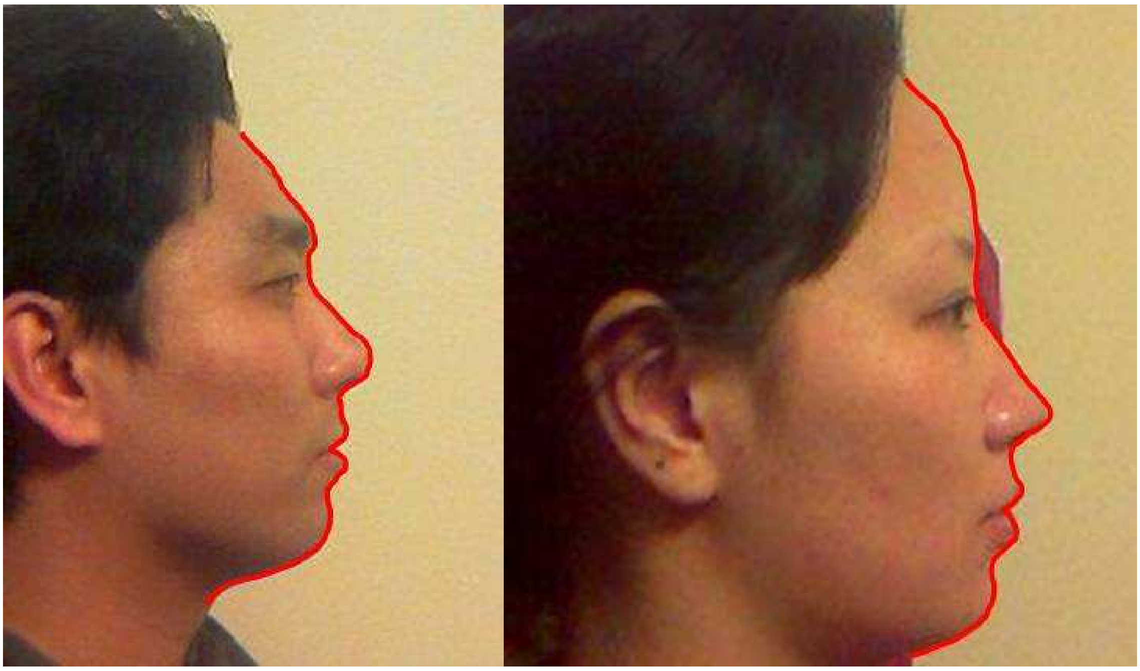


Figure 8.1 This graph shows manual segmentation on a male profile photo and a female profile photo. The red line is the segmentation result.

The manual segmenting scheme is simply placing points on the boundary from forehead to chin area. The red line is reconstructed by a spline interpolation algorithm based on the placed points. Figure 8.2 shows the points markers used in manual segmentation. It can be seen that the black circles/dots are the manually placed points, the face contour and the red curve is reconstructed based on these black circles by using 2D spline interpolation. For manual segmenting the datasets efficiently and quickly, different manual point placing schemes are used to make sure that the interpolated 2D curve is correct. For example, facial contours with high curvature, such as nose, are marked carefully with more points, and places where landform seems to be flat, such as the chin and forehead are marked with fewer points. During MDL or MEM optimization, the points along the curve needed to be moved freely to find the correspondence between each other. Since the curve is reconstructed from existing manually placed points, if a new point needs to be established between existing points, linear interpolation will be used to find a new point between nearest two existing points.

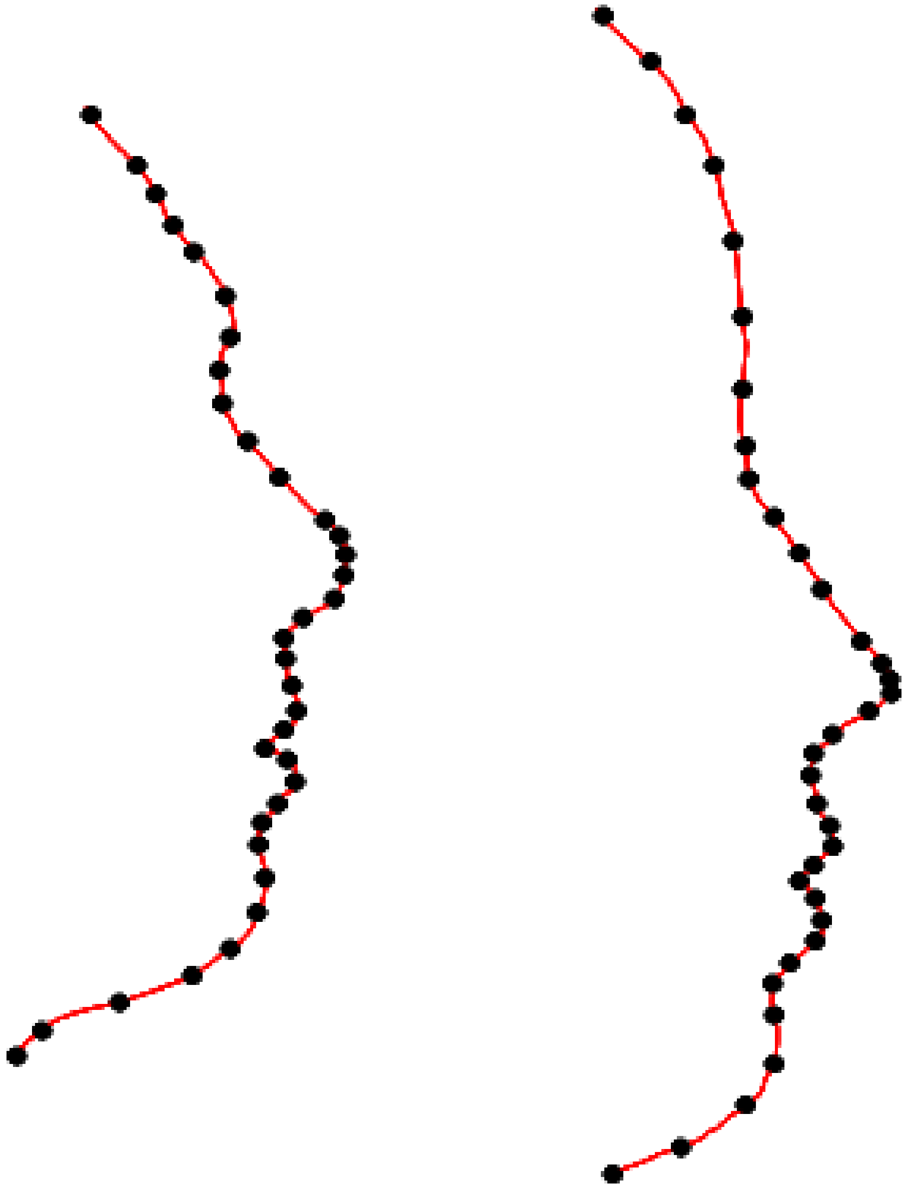


Figure 8.2 This graph shows the points placed on the face contours during manual segmentation. The red contours are the same as in Figure 8.1.

After performing manual segmentation to all 131 facial profile photos, we now have all the contours information. We show part of the segmentation results in Figure 8.3, where the first row is composed of female subjects, and the second row is composed of male subjects.

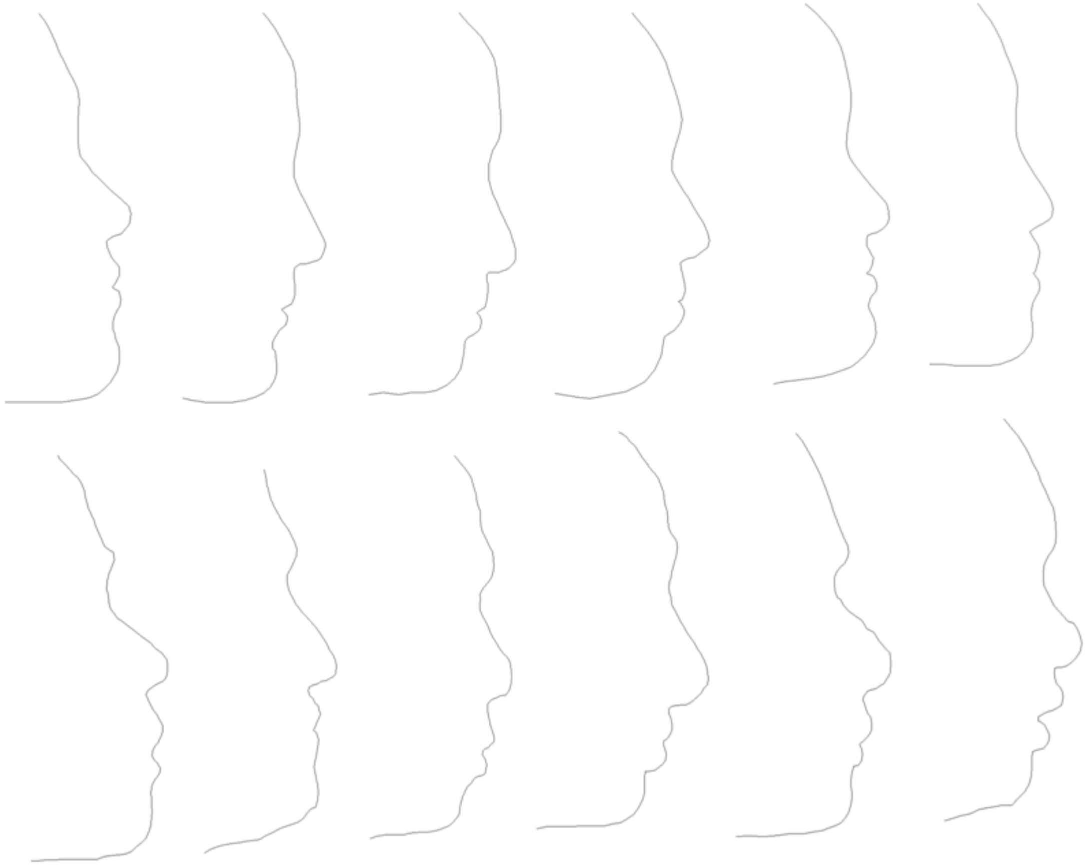


Figure 8.3 This graph shows twelve of the segmented examples from 131 facial profiles collected. Faces in the first row are female and at the bottom row are male.

Compared with female subjects, we can see that the male subjects tend to have larger nose, lips are bigger and men's eyes positioned deeper. As we discussed with several experiments participants, we concluded that they all use this information to judge a profile's gender.

8.2 Classification Method

As have been discussed in previous chapters, both MDL and MEM can be used to find correspondence points across 2D shapes contours automatically. In the experiment, we perform both technique to the datasets of 131 manually segmented facial profiles. After correspondence landmarks are found among shapes, Active Shape Models (ASM) are constructed from both MDL's results and MEM's results. After ASM is built, each shape is made by two components, which are mean shape and the result of multiplying shape variations and their weighting components. For the same ASM, the mean shape and shape variations are standard. The only difference between each shape is its weighting components/vector. We can therefore use the weighting vector as an input to the classification method to explore the difference between two groups with different gender. In our case, we are going to find the difference between the facial profiles between male and female. In this chapter, we perform Logistic Regression (LR) [100] on the weighting component/vector. LR (sometimes called the logistic model or logit model) is used for prediction of the probability of occurrence of an event by fitting data to a logistic curve. In our case, the probability of occurrence will be the subject's gender, which is either male or female. A simple logistic curve is shown in Equation 8.1.

$$Curve(t) = \frac{1}{1+e^{-t}} \qquad \textbf{(8.1)}$$

Where t is the curve‘s parameter. We choose Logistic Regression as our classification method, for its simplicity and easy usage for prediction within two natural categories. More information about Logistic Regression, is provided in reference [100].

A leave-one-out cross-validation of the prediction model is performed. For example, we have 131 datasets; we build correspondence points by using MEM, MDL or manual method. After correspondence points are found, an Active Shape Model can be built accordingly. Since the gender of each subject has been known, we can use the known information to train our Logistic Regression model based on 130 datasets, and make a prediction of the remaining dataset. A prediction of the model's gender will be given under the rule that: 1) $p(male) \geq 0$, $p(female) \geq 0$, and 2) $p(male) + p(female) = 1$. Here, p stands for the probability of the subject being male or female. In the validation, the assumption will be rejected if the probability is lower than 0.5. For example, let $p(male)$ be the probability of the subject being a man. Therefore, if $p(male)$ is smaller than 0.5, the prediction indicates that the sample under test is more likely to be a woman and vice versa.

For measuring accuracy of our classification results, we use overall classification accuracy, Sensitivity and Specificity. Please be noted that the Specificity used in this paragraph is different from the Specificity Ability term used in the previous chapters, which are used for comparisons of different correspondence points. Since the ground truth is known for each subject. The overall classification accuracy is simply the results of number of correct classified cases divides number of total cases. Sensitivity (also called recall rate in some fields) measures the proportion of actual positive cases, which are correctly identified as such (e.g. the percentage of male subjects who are correctly identified as being male). Specificity measures the proportion of negatives cases which are correctly identified (e.g. the percentage of female subjects who are correctly identified as being female). A theoretical, optimal prediction can achieve 100% Sensitivity (i.e. predict all people from the male group as male) and 100% Specificity (i.e. not predict anyone from the female group as male). In our case, Sensitivity is the classification accuracy within male groups, and Specificity is the classification accuracy within female groups.

In addition, we also asked 15 observers to guess the gender of the facial profiles independently. As we have known the gender of the subjects, the overall classification accuracy, sensitivity and specificity can also be easily calculated. For representing the overall accuracy for the whole 15 individual observers, we use the mean Overall Accuracy, Sensitivity, Specificity and their variations among all observers.

8.3 Experimental results

Before presenting the accuracy of each model's performance, we show some of the correspondence landmarks results achieved by MDL and MEM in Figures 8.4 and 8.5, respectively.

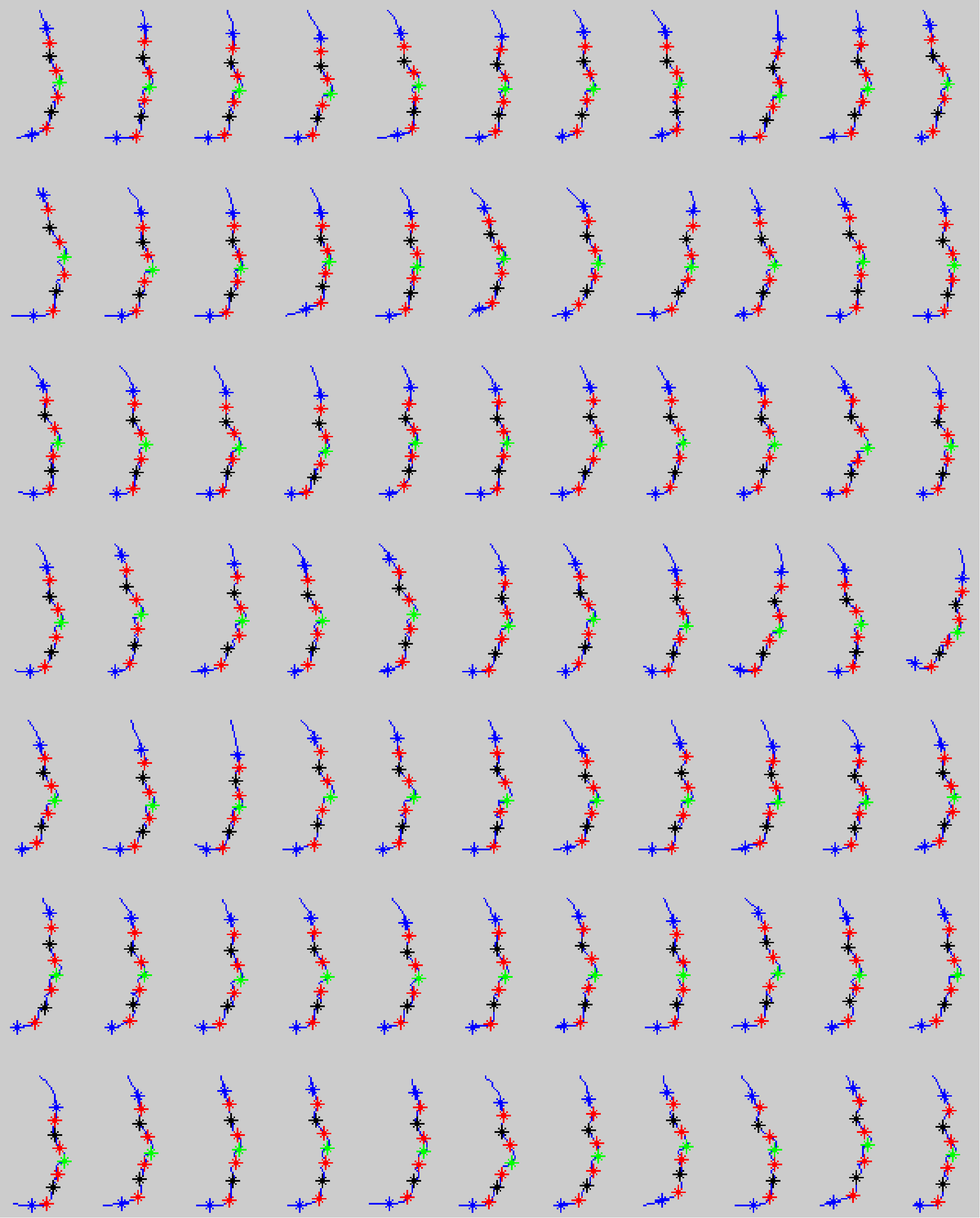


Figure 8.4 This graph shows the correspondence points found by MDL. The correspondence can be identified by the same colour.

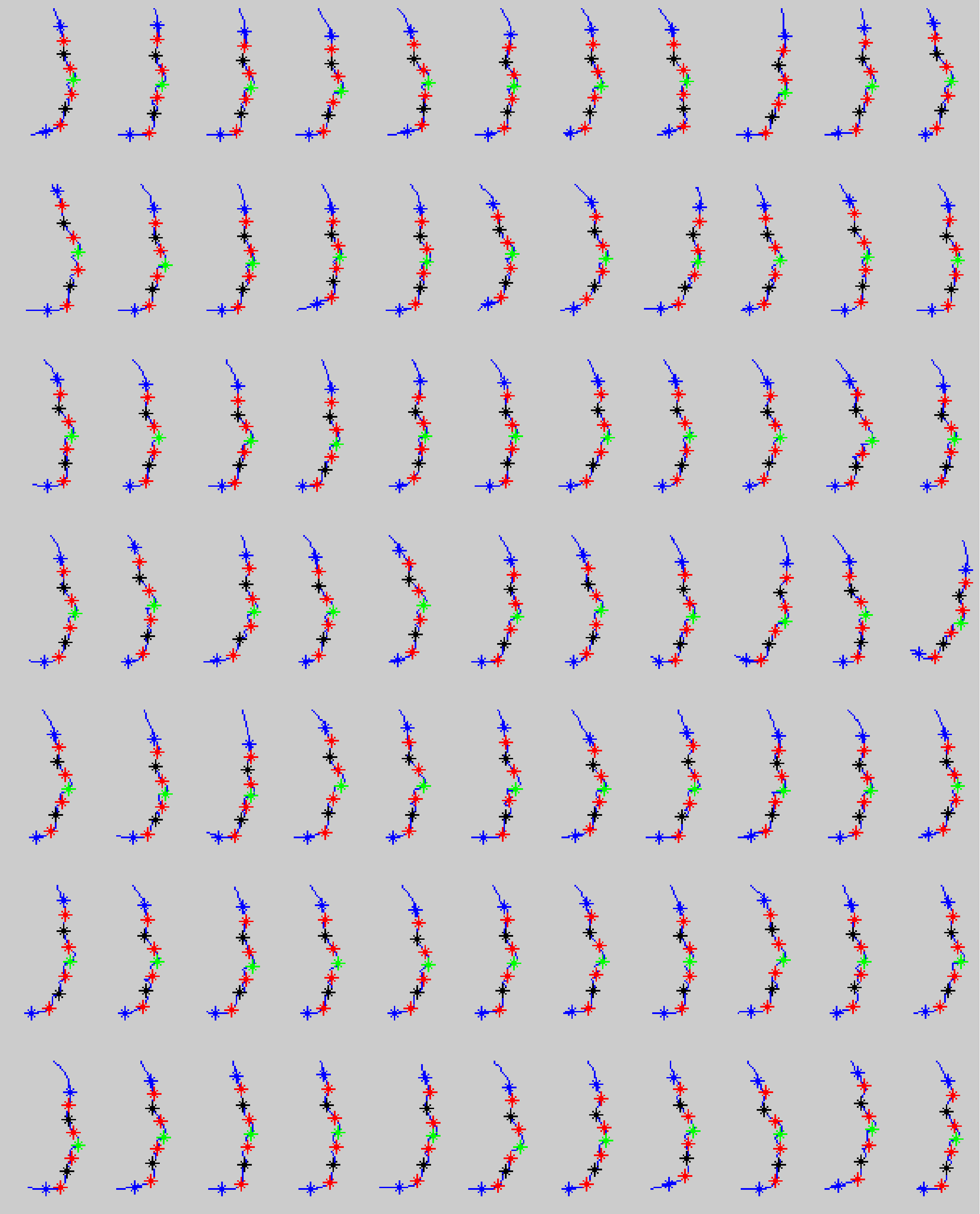


Figure 8.5 This graph shows the correspondence points found by MEM. The correspondence can be identified by the same colour.

From Figures 8.4 and 8.5, it can be seen that different colour represents the correspondence points in different level. The visual difference between MDL and MEM may not appear that significant, but there could be difference when using these two sets of correspondence points on a classification application.

After finding the correspondence points among shapes, we then build the ASM by using PCA. The mean shape from MDL and MEM correspondence points and the captured shape variations are represented in Figure 8.6.

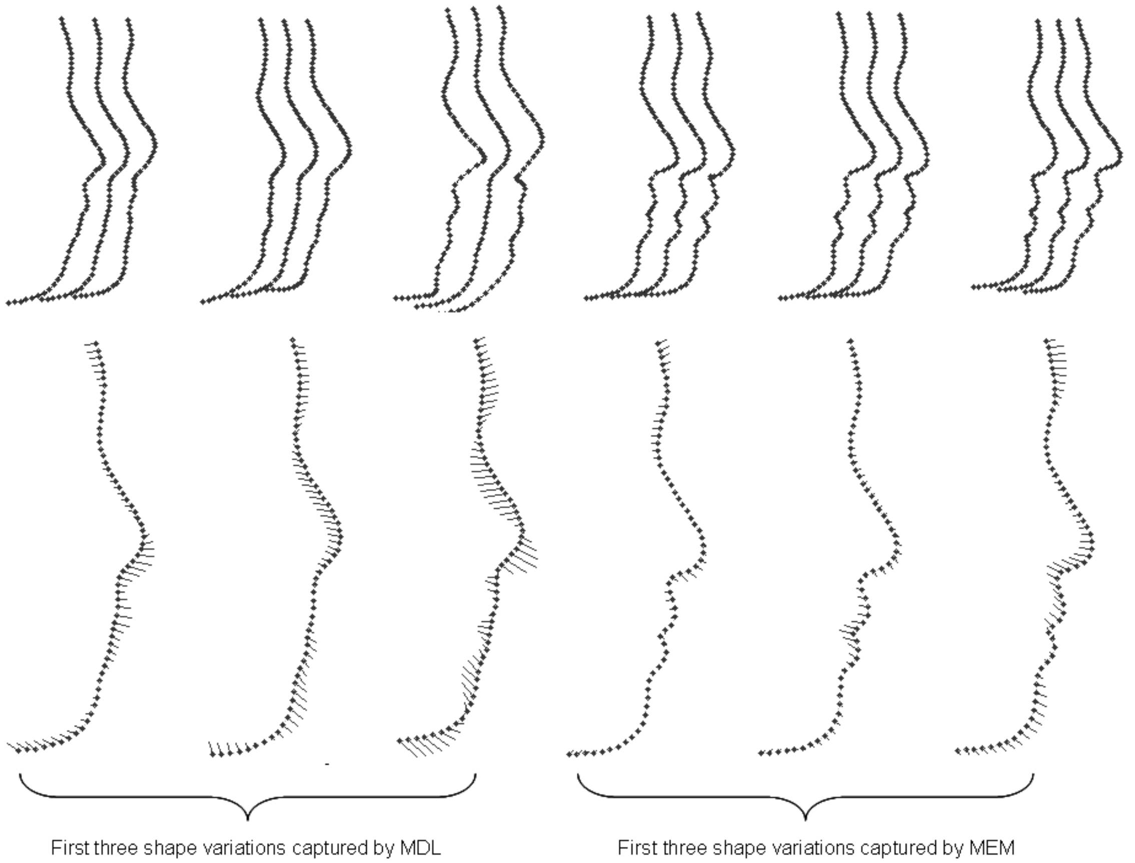
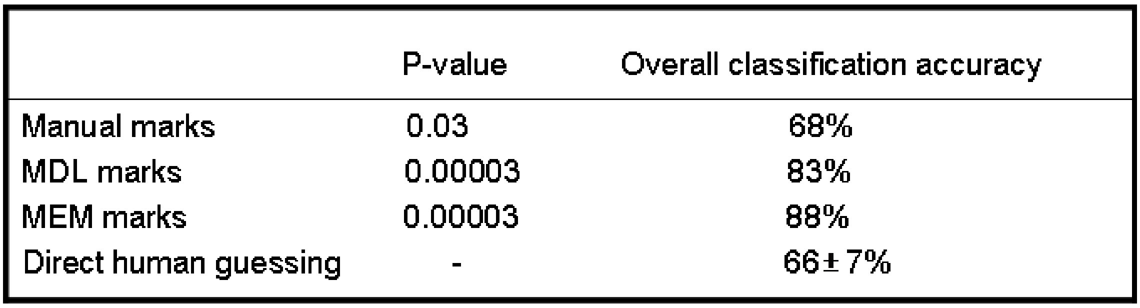

First three shape variations captured by MDL
First three shape variations captured by MEM

Figure 8.6 On the first row, the first three subjects are shape variations captured by MDL and the last three subjects show are shape variations captured by MEM. On the bottom row the contour shows the mean shape from MDL and MEM. Same as previous figures, the wiskers represent the shape variations.

After building the ASM, we use each shape's weighting vector as an input to the Logistic Regression to perform gender prediction. In this section, we give the comparison results between the four studies. Among these four studies, three are automatic prediction methods and one is based on direct human observation. For the direct human observation test, we invited 15 observers to guess the gender of the facial profiles independently. As the gender of each profile is known, we can calculate the Classification Accuracy, Sensitivity and Specificity for each observer's guess. Since there are multiple observers, we use the mean value and standard deviation to represent the direct observers' gender prediction ability. For the automatic methods, we used correspondence points found by manually method, MDL and MEM to build ASM. Randomly select 130 datasets to train Logistic Regression and make gender prediction on the remaining subject. The overall classification accuracy by four methods and their standard deviation among the 15 evaluations and the statistic *P*-value for automatic methods as recorded are shown the Table 8.1:

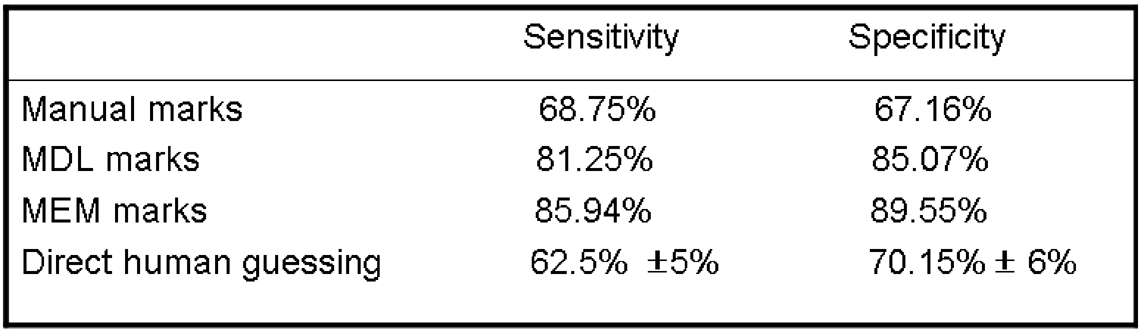


	P-value	Overall classification accuracy
Manual marks	0.03	68%
MDL marks	0.00003	83%
MEM marks	0.00003	88%
Direct human guessing	-	66± 7%

Table 8.1 This table shows the scores of different methods. In the second column, the *P* value is the confidence level, the third column shows the overall classification accuracy.

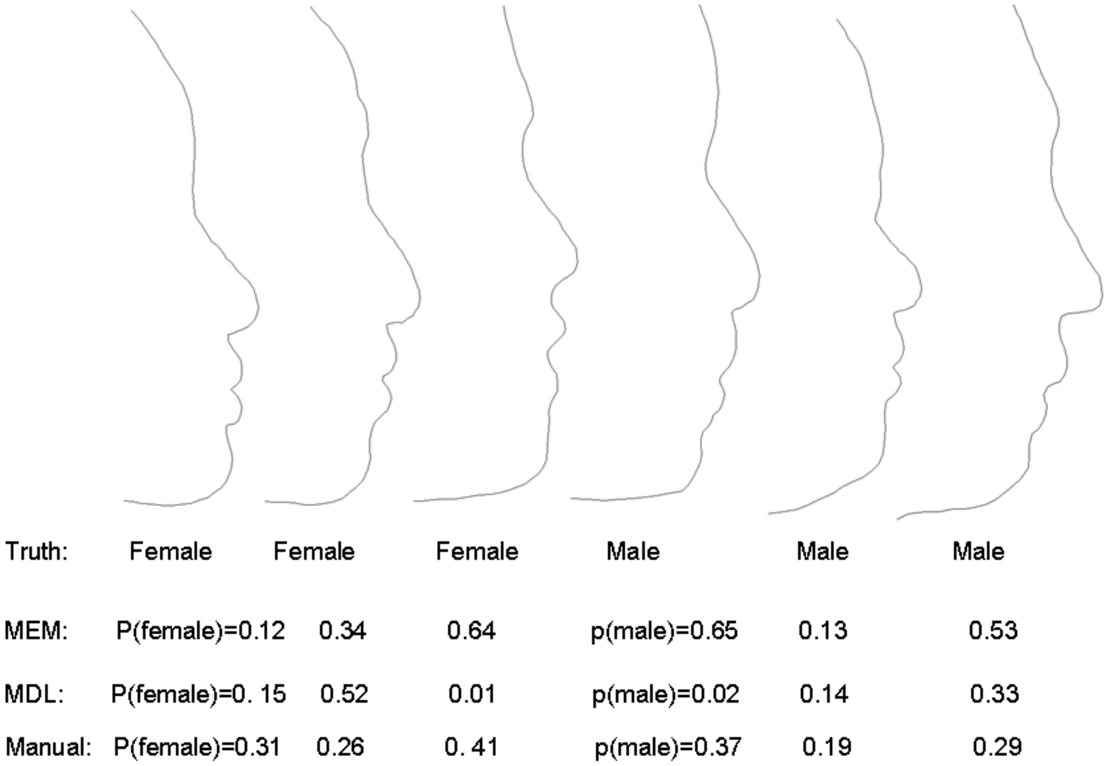
From Table 8.1, it is interesting to observe that the model created by MDL and MEM methods gave prediction of the gender more accurate than the manual labelled model and direct human observing. A very small P-value means a very high certainty of the classification accuracy estimation. Since all the P-values are very small, our overall classification accuracy can be statistically trusted. Between the MDL and MEM models, MEM outperforms MDL by 5%. During the experiment, it is also observed that it is quite important to include the chin area to guarantee a good performance, which agrees with the findings of reference [38]. If we exclude the chin area, the model by manual landmarks, MDL and MEM model's correction ratio will all degrade to about 50%. Other than that, we found that the forehead area is also very important in the performance of the model. This is actually quite interesting finding, since that in our previous finding in the facial contours that the difference between female and male are in the eye, lip and nose areas. Maybe that is one of the reasons that human direct observer are lower than machine's automatic prediction results. Machine uses all the information in the datasets whereas a human observer only picks up the most relevant information he or she believes to be relevant. Additionally, different individuals all agreed that they mainly use the curvature as criteria for prediction. This is due to recognition that in China, female faces have more flat landforms and male faces are more popped out and full landforms. The landform difference can also be observed from Figure 8.3. Another interesting finding by the observers is that some of the good-looking female and handsome male subjects are more likely to be miss-predicted by only looking at their facial profiles and their prediction probability is about 0.4. Therefore, the more "beautiful" faces are more involved with features from both male subjects and female subjects.

The Sensitivity and Specificity results are shown in Table 8.2. From this table, we can observe that MEM marks achieves the highest score in Sensitivity, which means MEM marks classification is the most accurate in male subjects and MEM marks also achieves the highest score in Specificity, which means MEM marks classification is the most accurate in the female subjects. One interesting observation on this part of results is that the direct human observation is more accurate than manual marks classification in Specificity, and less accurate in Sensitivity.



	Sensitivity	Specificity
Manual marks	68.75%	67.16%
MDL marks	81.25%	85.07%
MEM marks	85.94%	89.55%
Direct human guessing	62.5% ±5%	70.15% ± 6%

Table 8.2 This table shows Sensitivity and Specificity of the four classication results. From this table, it can be seen that the MEM marks based classification achieves the highest scores on both Sensiticy and Specificity.



Truth:	Female	Female	Female	Male	Male	Male
MEM:	P(female)=0.12	0.34	0.64	p(male)=0.65	0.13	0.53
MDL:	P(female)=0. 15	0.52	0.01	p(male)=0.02	0.14	0.33
Manual:	P(female)=0.31	0.26	0. 41	p(male)=0.37	0.19	0.29

Table 8.3 This graph shows some of the wrong prediction results made by MEM, MDL and manual model. In these examples, manual results made wrong prediction on all six examples; MEM had three correct guess whereas MDL had only one correct guess.

Table 8.3 shows some of the results, whose gender has been wrongly predicted by most of the observers and also the automatic prediction models. From this table, the ground truth is that the first three examples are female and the last three examples are male. For the model built from manual results, since the probability in all is lower than 0.5, the predictions were wrong for all six cases. For model built from MEM, the prediction was correct for the third, fourth and sixth case. For model built from MDL, the prediction was correct for only the second case.

8.4 Conclusions

In this chapter, we demonstrated one of the applications of our proposed MEM method, which is using the automatic found correspondence points to build Active Shape Model (ASM), and use this model's parameters as an input to the classification method to perform gender prediction on facial contours. In the experiments, we also perform prediction accuracy comparisons between four different gender prediction methods. Three of them are automatic methods, which are using ASM built from manual landmarks, MDL found landmarks and MEM found landmarks. The fourth method is by inviting various people to participate direct guessing based on observing the facial contours. The overall classification accuracy results show that the ASM built from landmarks found by MEM and MDL are the first tier performers and the model built from MEM landmarks can provide 5% more accurate estimation than MDL. Direct human observing and model built from manual landmarks are in the second tier with accuracy 66% and 68% respectively. For Sensitivity, the performance from best to worst is MEM marks, MDL marks, manual marks and direct observation. For Specificity, the performance from best to worst is MEM marks, MDL marks, direct observation and manual marks.

The conclusion is that using computer to perform classification of profiles is relatively more accurate than direct human guessing. We attribute this reason to the fact that machines are actually using all the input information such as forehead, eye, nose, lip, chin, etc, but human observer only focus on local information such as eye and lip shape without considering the whole picture. Another interesting finding is that, we pick up some of the wrong prediction picture made by MDL and MEM based classification. Most of the observers agree that their wrong predicted faces seem to be more handsome or pretty than the correct predicted ones. It looks that for machines, these category of faces are involved with features from both male subjects and female subjects. When using automatic methods found correspondence to perform classification, MEM found correspondence can provide 5% more accurate overall classification accuracy than MDL found correspondence points.

Chapter 9 Conclusions and Future Work

9.1 Contributions

In the previous chapters, we have presented a new framework of finding the surface correspondence points across either 2D or 3D datasets automatically. In this section, we will summarize the conclusions and directions for future research work.

The main contributions of this thesis are listed as follows:

1) A new objective function, which provides a measurement of model utility

A Minimum Entropy based objective function is derived to compose the cost-function. Unlike MDL where, the total cost function is simply added up directly, MEM has different weights for each Entropy component.

2) The gradient of the MEM cost function is derived for a faster convergence

Based on the results of SVD products on the shape covariance matrix, the gradient of the MEM cost function can be derived successfully. By using this gradient, the Steepest Descent optimization algorithm can be incorporated. Compared with the original simplex approach in MDL, the new gradient method is reasonably faster.

3) A more shape feature preserving shape parameterization and re-parameterization method

Unlike the simplified version of spherical harmonics used by Davies, we use conformal mapping as our 3D shape parameterization method. According to references [52], [53], this mapping technique can preserve more shape information and minimize angle distortion. A more efficient re-parameterization method is used, which allows new kernel to be added without disturbing other landmarks.

4) The framework is applicable to both 2D and 3D datasets

The scheme for both 2D and 3D implementation of MEM algorithm is provided. This framework has been successfully applied to several 2D and 3D datasets. In addition comparisons against the MDL algorithm based on objective metrics we performed both on 2D and 3D, on different datasets and scenarios.

5) Solving the "Pile Up" problem

This well reported problem is inherently solved by the replacement of MDL cost-function to MEM cost-function. In both 2D and 3D scenarios, MDL runs into "Pile Up" several times, however, MEM did not encounter this problem at all, since MEM favours a distributed correspondence, and MDL favours a congested correspondence.

6) Using MEM to perform an automatic classification scheme building and perform comparisons with other methods

I used MEM, MDL, and manually labelled landmarks on the facial contours to build up a gender classifier. The comparison is performed between these three classifiers and direct human observation on a large dataset of facial data. Overall classification accuracy, sensitivity and specificity are used here. The results showed that the MEM's classifier outperform other methods in all three evaluations.

9.2 Future Work

Although the proposed MEM has been successfully applied to 2D and 3D datasets and it has been shown that MEM preserves better shape properties than MDL does, the proposed MEM algorithm is not perfect yet. There are still some limitations of the approach to automatic shape modelling; we present some future research directions here.

9.2.1 Discrimination Analysis

In chapter 8 we presented preliminary 2D discrimination analysis results, it is quite straightforward to think about extending the current technique to 3D cases. The MEM model can be used as a basis for exploring differences in shape between normal and abnormal objects. By doing this analysis, we can translate the technical superiority into real practical applications. In medical image processing, many datasets exist in 3D format and 3D information can give a more direct measurement of potential illness. For example, in Davies's thesis [22], he suggests that shape information provides better discrimination of schizophrenia and normal subjects, than volume measurements. The discrimination objective can be pursued by constructing the MEM model on the training set consisting with both schizophrenia subjects ($\sum SC_i$) and normal subjects ($\sum SN_i$). After that, a classification method, such as Logistic Regression used in Chapter 8, can be used on the parameters consisted of shape weighting components. In Davies's thesis, he simply uses Linear Discriminate Analysis (LDA) to classify these two groups. The discrimination direction can be found at the same time. Figure 9.1 shows the results of the LDA analysis results on 56 schizophrenia datasets and 26 normal datasets from reference [22]. It can be seen from the graph that the schizophrenic patients will be most likely to have a hippocampus with longer and thinner tails. From left to right, the graph shows the shape changes from normal to schizophrenic shapes.

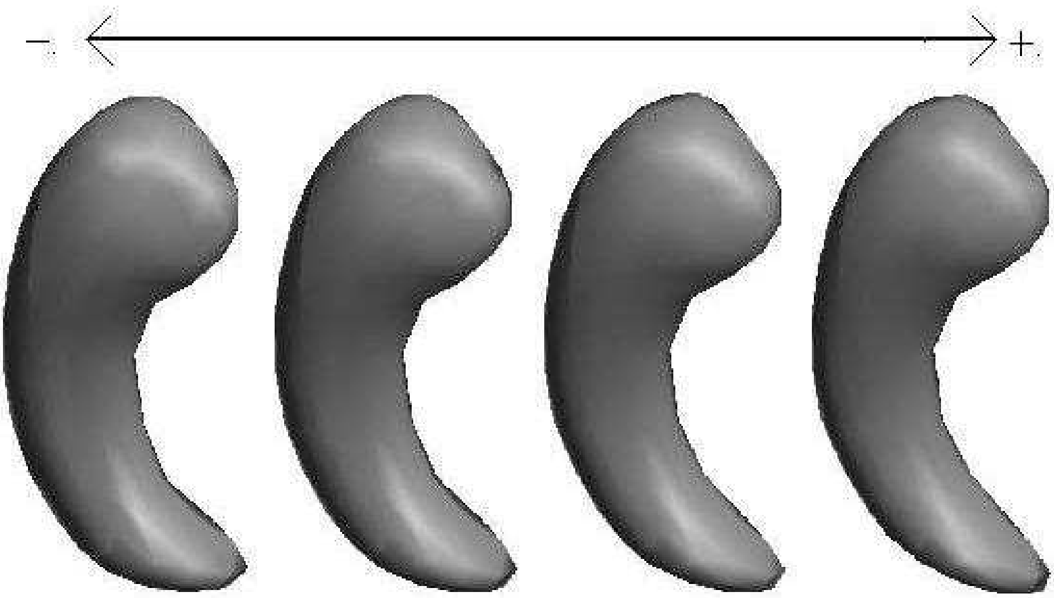


Figure 9.1 The model was built for visualizing the shape difference between Normal subjects and Schizophrenic subjects. In the graph, "–" indicates Normal and "+" indicates Schizophrenic.

The accuracy of the MEM model can be justified by performing Leave-One-Out analysis. The pseudo code below, shows how to perform this analysis.

```
For shape example $i = 1 \ldots n_s$
    1) MEM is performed on the training set with $x_i$ excluded
    2) Shape parameters are separated into two groups
    3) LDA is performed to find the separating vector
```

4) The excluded example is mapped to discrimination space

5) Go to step one, if this is not the last example

More details and results about this LDA analysis are discussed in reference [22]. Another interesting future direction about the Discrimination Analysis is trying to use other classification or regression method rather than linear decimation method, e.g. Support Vector Machines [67], [68], Discrimination Analysis by using SVM can be found in [69], [70].

Essentially, we are planning to find the connection between shapes and biology. For example, shapes of brain may indicate gender; shapes of hippocampus may indicate some potential illness. Moreover, there is no ground truth for measuring the accuracy of correspondence found by different approaches. It may be a good idea to use the automatic found correspondence points to build a shape model, therefore test the shape model's ability in some applications. Testing the model's ability in application may be a more intuitive way to measure the model correspondence accuracy, rather than using the three comparison criteria.

9.2.2 More Datasets

As has been discussed in the introduction and literature review, the automatic shape modelling is an important technique, which has broad applications. Besides discrimination analysis on the datasets of healthy and unhealthy hippocampus, Brain modelling can be another good application.

Brain modelling is directly related to Neuroimaging and Human Brain Mapping, as brain can be segmented into different functionality zones each playing a different role. Therefore, it can be quite helpful for a physician to know the corresponding zone or points between a patient brain and brain atlas. An example of the brain atlas is shown in Figure 9.2.

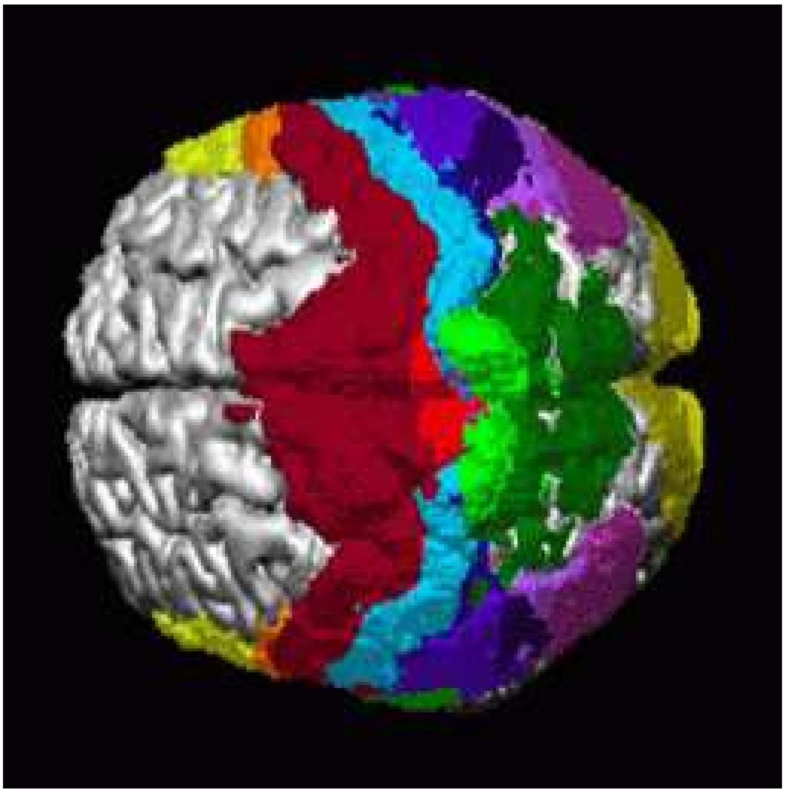


Figure 9.2 A brain atlas example, different colour represent different functionality zone.

The approach for tackling this problem is simple; datasets are segmented and classified by an expert. Therefore, when a new example (patient dataset) joins the datasets, MEM will be performed and will find the corresponding points between the new coming dataset and prepared datasets. In this way, the corresponding zone will be found. There is some similar work on this brain surface correspondence finding problem. Figure 9.3, is from reference [48], where the author found 69 corresponding points across datasets.

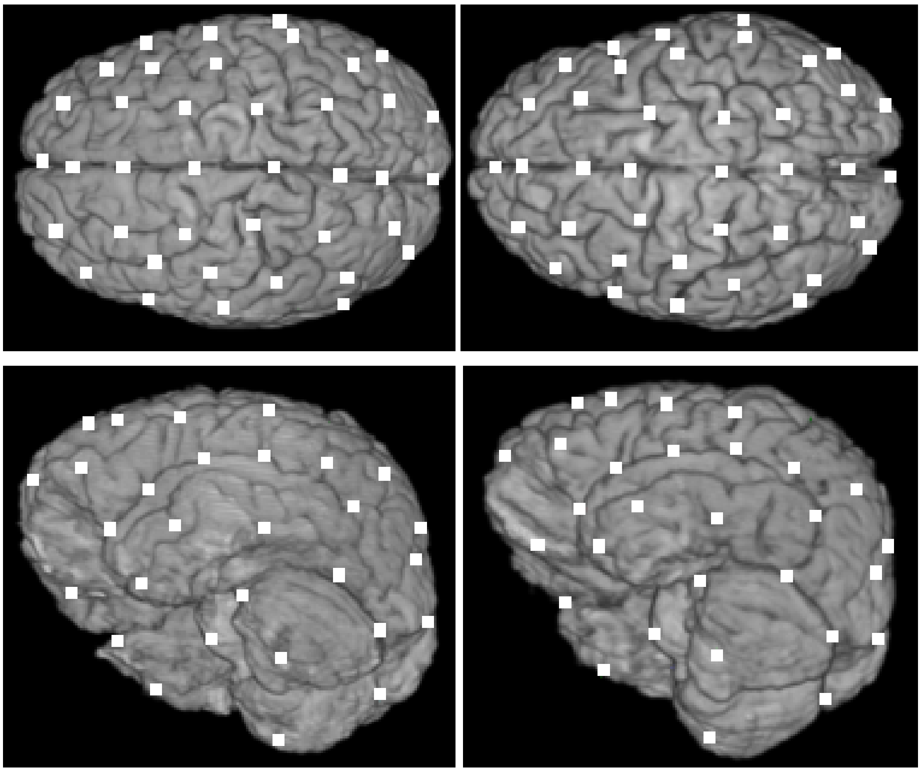


Figure 9.3 From top to bottom, it shows different views of brain. Left: Atlas surface hand-labelled 69 points; Right: Correspondence points found in [48] .

9.2.3 MEM with Appearance Information

As reported in reference [16], shape information is sometimes not sufficient for finding the correspondence an ASM, especially when the shapes contain considerable variations. Fortunately, we can incorporate appearance information into the current MEM model and form a MEM appearance model.

Different from ASM, the model is parameterized by using both shape coordinates and gray level information of the landmarks.

$$X = (x_1, y_1, z_1, \ldots, x_n, y_n, z_n, I_1, I_2, \ldots, I_n) \qquad \textbf{(9.1)}$$

Where x_i, y_i, z_i are landmark coordinates in X, Y, Z direction respectively and I_i is the intensity information on the landmarks.

According to reference [16], with the help of the appearance information, shape model can provide better properties, for example segmentation accuracy. In this sense, we are hoping the MEM appearance model can convey better model properties than the original MEM. Some of the preliminary results about MDL appearance model appeared in [71], [72]. In reference [71], the basic theory about MDL and MDL appearance model was introduced. In paper [72], the author represented work of facial recognition by using MDL appearance model. The results demonstrated that with the appearance information the MDL model can find correspondence points more accurately.

9.2.4 MEM with Arbitrary Topology Structure

It has been shown in our 3D work section that the MEM can be applied to shapes with genus zero topology. In other words, any shape with sphere topology can be modelled by MEM automatically. Therefore, it is quite straightforward to ask if we can model shapes with arbitrary topology. For example, diaphragm is an important structure, which divides the human trunk to chest and abdomen. In many applications, e.g. liver/heart segmentation, it will be of great help if the position of diaphragm can be found. However, in terms of intensity, there is little difference between diaphragm, bottom of heart and top of liver. Other researchers in reference [73] have tried to use Active Appearance Model to segment the diaphragm. However, the manual landmark placing makes the method difficult to use in practice. By incorporating the proposed MEM, we can ease this problem. In order to model diaphragm, we have to refine our framework.

We suggest using the framework below for modelling shapes like diaphragm, which can be considered as open-shapes.

Scheme for modelling open-surface shape

1) *Pre-process datasets: centring, roughly aligning datasets*

2) *Rough initial landmarks: initial landmarks can be roughly placed by algorithms like ICP*

3) *Measure MEM cost function: MEM is calculated in this step*

4) *Re-parameterization: Move landmarks in shape space*

5) *Stop criteria: finish optimization when convergence is achieved*

Comparing this approach with our original approach, there are some differences: 1) Shapes are not parameterized by a sphere anymore; therefore optimization is performed directly in shape space. 2) For the same reason of absence of parameterization, ICP is used for initial landmarks placement.

The advantage of this approach is that it can model shapes with open surface. However, a suitable re-parameterization has to be developed to cope with different surfaces.

9.2.5 Shapes with Non-Linear Variations

One of the essential goals of Active Shape Model is to extract shape variations from aligned landmarks cloud. A standard technique for such variations extraction is using Principal Component Analysis (PCA). PCA assumes that variations are linear in Euclidean vector space, which is insufficient and inefficient on datasets with non-linear variations and these non-linear variations are common in medical datasets. To some extent, the non-linear variation problem can be solved by approximations using a combination of linear components; however, the use of linear components increases the dimensionality of the model and allows for non-valid shapes [82]. Therefore, this approximation error remains in MDL and MEM approach, and sometimes can influence the results. Algorithms presented in [80], [81] have been developed to complement this approximation error in PCA. In this section, we are going to discuss the possibility of a nonlinear approach, which incorporates the nonlinear PCA with finding correspondence in a learning process. Preliminary results are shown, which are based on the comparison between PCA MDL and Nonlinear PCA MDL. Again, comparisons were performed by evaluating the Generalization Ability, Specificity and Compactness.

9.2.5.1 Introduction of Nonlinear PCA

The idea of nonlinear PCA is quite intuitive. PCA can be effectively performed on a set of observations that are linear. When variations are not linear [75], they will be mapped to a higher dimension where shape variations are again linear. PCA can then be applied in the higher dimensional space. A simple illustration of this mapping from lower dimension to higher dimension is in Figure 9.4.

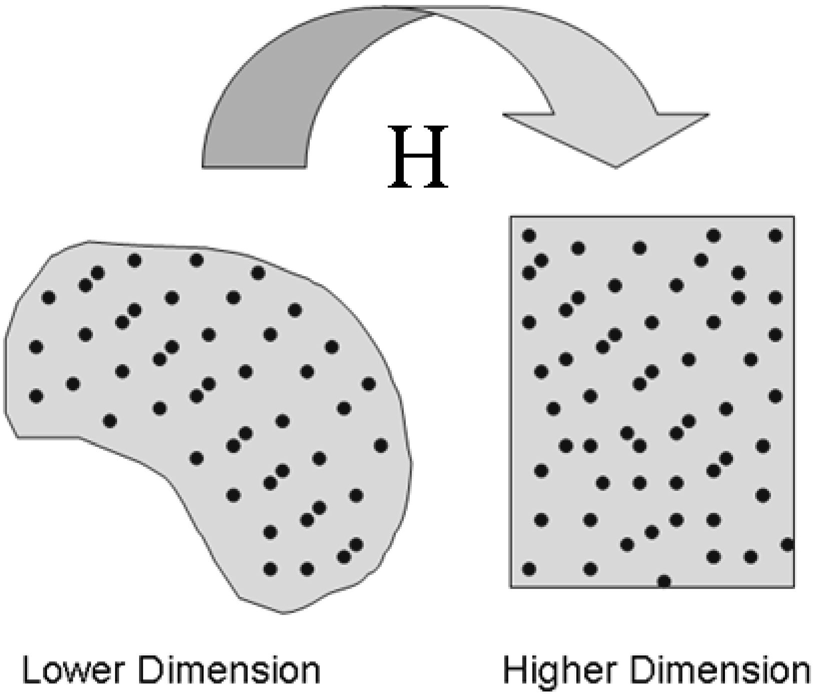


Figure 9.4 The graph shows the process of mapping the original shape vector to a higher dimension, nonlinear variations can be mapped to linear variations at the same time. Here H **denotes this mapping process.**

More precisely, PCA is going to be performed in the linear higher dimensional space [45, 76]. Given a set of aligned *N* shape vectors $\sum_{i=1}^{N} x_i$, we are going to decouple the nonlinear correlations through diagonalising their covariance matrix. For example:

$$A = \frac{1}{N}\sum_{i=1}^{N} \mathrm{H}(x_i)\mathrm{H}(x_i)^T \qquad \textbf{(9.2)}$$

Same as Figure 9.4, $\mathrm{H}(\bullet)$ is a nonlinear mapping function which projects the input shape vectors from input space to feature space. To decouple the covariance matrix *A*, we have to solve the Eigen problem in Equation 9.3:

$$\lambda p = Ap \qquad \textbf{(9.3)}$$

Where λ is the eigenvalue, and *P* is a matrix where each column is the eigenvector of the matrix *A*.

If we multiply *p* on each side of Equation 9.2, we will get Equation 9.4

$$Ap = \frac{1}{N}\sum_{i=1}^{N}(H(x_i)p)H(x_i)^T \qquad \textbf{(9.4)}$$

Therefore, there must be coefficients c_i so that

$$p = \sum_{i=1}^{N} c_i \mathrm{H}(x_i) \qquad \textbf{(9.5)}$$

If we combine Equation 9.2, 9.3 and 9.5 together we will find that:

$$\lambda \sum_{i=1}^{N} c_i (\mathrm{H}(x_k)\mathrm{H}(x_i)^T) = \frac{1}{N}\sum_{i=1}^{M} c_i (\sum_{j=1}^{N} (\mathrm{H}(x_k)\mathrm{H}(x_j)^T)(\mathrm{H}(x_j)\square\mathrm{H}(x_i)^T)^T) \qquad \textbf{(9.6)}$$

This equation provides a clue that the previous mentioned eigenproblem can be solved by dot products of mapped shape vectors in higher dimension. Since computing such dot product in high dimensional space is still expensive, Support Vector Machine (SVM) [83] can be used to ease this problem.

We can define a $N \times N$ matrix S, where $S_{ij} = \mathrm{H}(x_i)\mathrm{H}(x_j)$, therefore, Equation 9.6 can be rewritten as Equation 9.7:

$$N\lambda P = SP \qquad \textbf{(9.7)}$$

Where $P = [\,p_1; p_2; \ldots p_N\,]^T$ (MATLAB notation), is the eigenvector.

We adopt the Gaussian Kernel to model the matrix *S*, as in Equation 9.8:

$$S(x, y) = e^{(-\frac{\|x-y\|^2}{2\sigma^2})} \qquad \textbf{(9.8)}$$

Where *e* is the base of natural logarithm, and σ is the standard deviation. As we are discussing the shape model in nonlinear cases, we will first discuss the weighting vector

(Please recall what we defined it in the previous linear cases in Equation 2.8). The weighting component b of a shape vector x_i can then be extracted by projecting $\mathrm{H}(x_i)$ on each column of eigenvector matrix P as in Equation 9.9, were a_i is the weighting component.

$$b = P \cdot \mathrm{H}(x_i) = \sum_{i=1}^{N} a_i (\mathrm{H}(x_i) \cdot \mathrm{H}(x_i)) \qquad \textbf{(9.9)}$$

Again, this can also be solved by using a dot product from Gaussian Kernel. However, the nonlinear process discussed previously can be represented differently by giving a specific *exponential* mapping and its reverse *logarithmic* mapping, which was developed in Principal Geodesic Analysis [81]. *Exponential* mapping will be used as H mapping and *logarithmic* mapping will be the inverse of H mapping. In reference [81], Geodesics are used extensively, to catch variations in high dimensional space. Given a set of shapes $x_1, x_2, \ldots, x_n$ and a fixed mean shape μ on a complete, connected manifold *M*, the definition of variations will be given by:

$$\sigma^2 = E\left[d(\mu, x_i)\right] \qquad \textbf{(9.10)}$$

Where *d* means the Riemann distance and *E* is the mathematic notation for “Expectation”. We can see that the variations of the data are equal to the expected value of the square Riemannian distance from the intrinsic mean. By using the Exponential Map and Log Map concepts, the Equation 9.10 can be rewritten as in Equation 9.11.

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^{n} d(\mu, x_i)^2 = \frac{1}{n} \sum_{i=1}^{n} \left\| \log_{\mu}(x_i) \right\|^2 \qquad \textbf{(9.11)}$$

The projection of one vector to another is also intuitive, which is defined as a minimization of distance process. Although there is no guarantee that such projection exists, the authors of reference [81] argued that given a small enough neighbourhoods around the mean shape, the unique shape can be assured. We adopt this nonlinear approach and combine it with the existing MDL approach. The purpose of this technique combination is trying to find if the nonlinear shape variations extraction method will somehow improve our correspondence finding algorithm. In the next section, we will present our preliminary experimental results on this approach.

9.2.5.2 Experiments on Nonlinear MDL

In order to validate our proposed algorithm, our experiments are conducted on the dataset of facial contours, which were used in the 2D work section. The same datasets were also used in the section of solving the “Pile Up” problem, where the original MDL met the “Pile Up” problem.

It can be seen from Figure 9.5 that by using the nonlinear MDL optimization the algorithm did not meet the “Pile Up” problem. All landmarks are placed in a seemingly same manner. In Figure 9.6, it can be seen that the cost function stabilized after 7000 steps in optimization. Again, we show the movement of each node in Figure 9.7

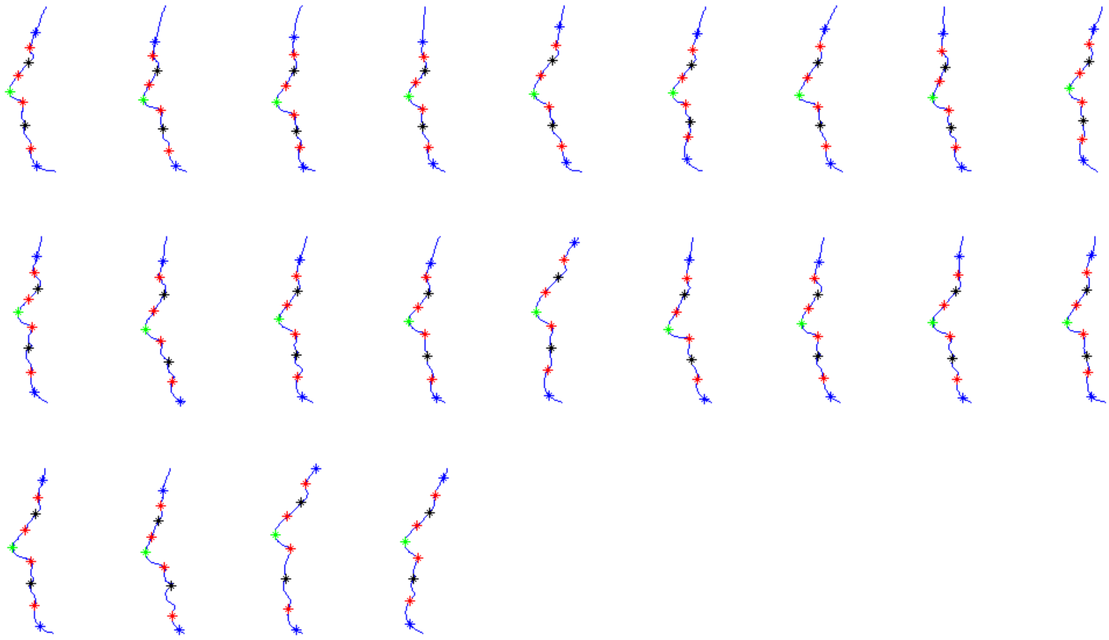


Figure 9.5 Results of nonlinear MDL analysis of facial contours.

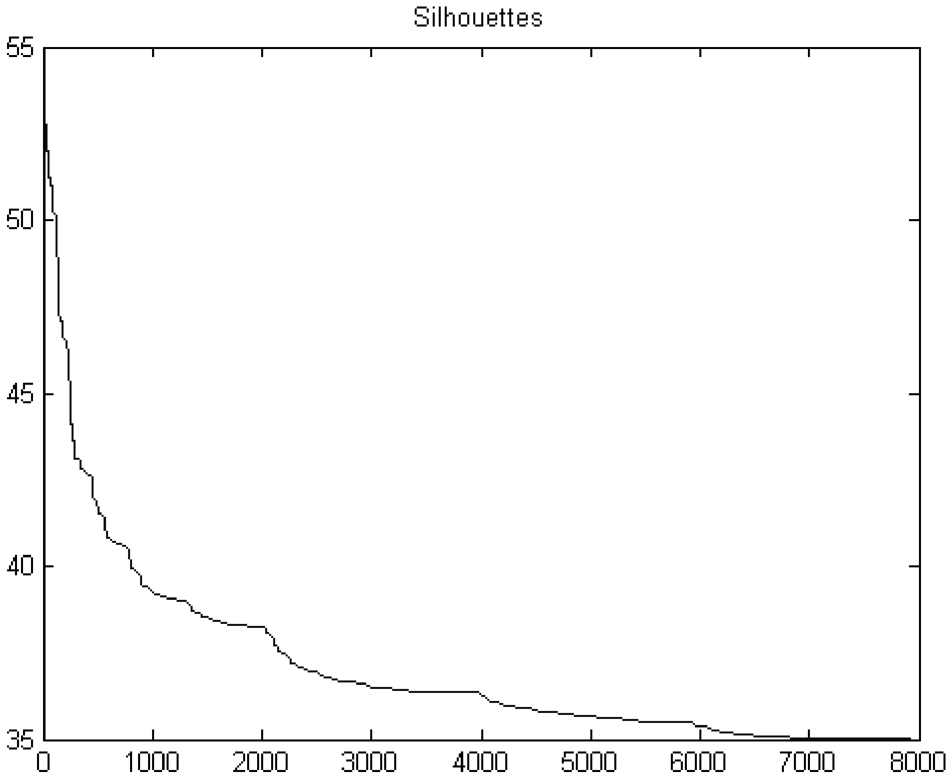


Figure 9.6 Results of performance of cost function. X-axis represents number of steps and Y-axis represents value of cost function.

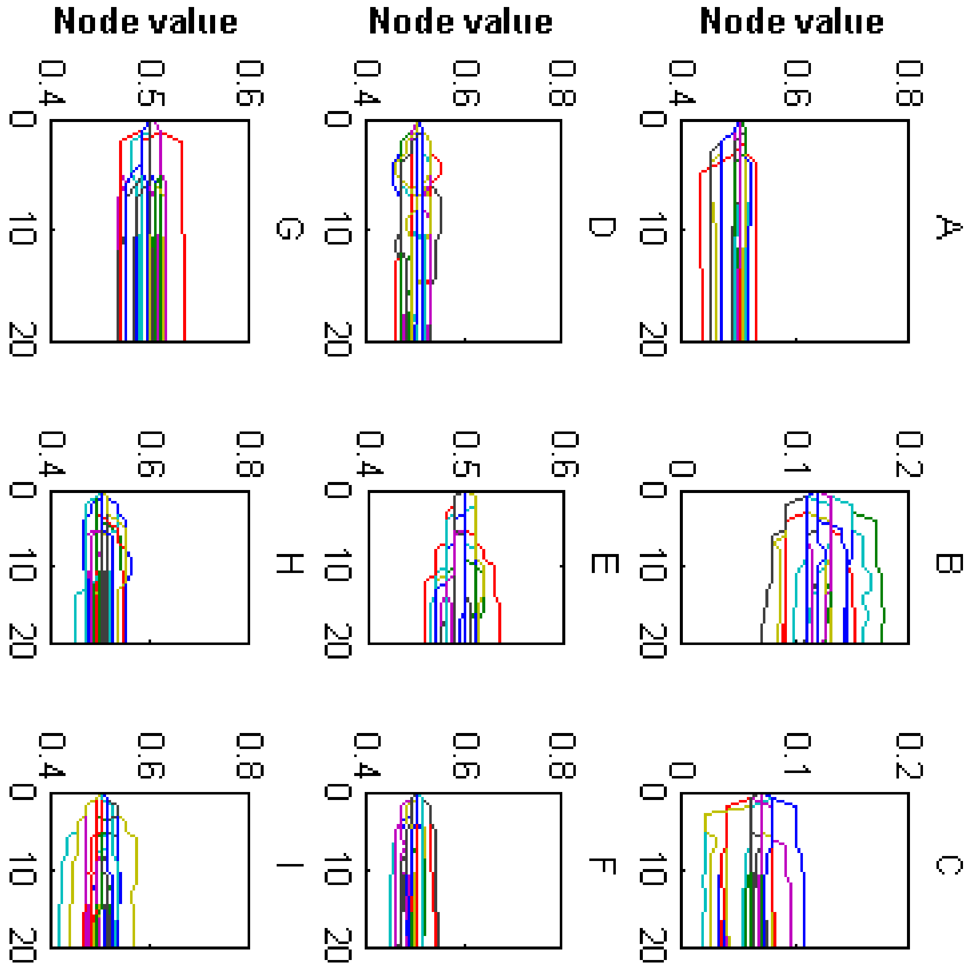


Figure 9.7 Movement of nine nodes are shown here. Each graph gives the movement of the node in 22 examples represented by different colour. X coordinate is step number and Y coordinate is node value. All nodes start to move from parameter 0.5 and stabilize around the 15th step.

9.2.5.3 Conclusions

In this section, our preliminary results on using nonlinear analysis to find correspondence are discussed. By mapping shape vectors into a higher dimension, nonlinear shape variations can become linear in that dimensional space. Therefore, PCA and MDL can be applied in this higher dimension. From our experimental results, we can see that the proposed nonlinear MDL can find correspondence across datasets automatically. It can also be observed from Figure 9.5 that, by using nonlinear PCA to replace linear PCA, MDL did not meet the “Pile Up” problem on the datasets of facial profiles. Another interesting preliminary result is that, compared with results on using linear MDL, nonlinear method can find correspondence much faster.

9.3 Final Conclusions

In this chapter, we presented the conclusions of our contributions to the areas of automatic Active Shape Model building, especially the “Correspondence Problem”. Rather than using Description Length as a measurement of cost-function, we proposed using Entropy as an alternative. Besides that, we proposed using a proper weighting for energy component extracted from each eigen-shape vector. Compared with the MDL approach, the newly proposed MEM can have better shape properties in the criteria of Generalization Ability and Specificity and also similar in the criterion of Compactness. With the derivation from Single Value Decomposition, MEM’s gradient can be computed efficiently. Therefore, MEM method incorporating the Steepest Descent approach can run faster than MDL’s approach. MEM also shows great potential to deal with the “Pile UP” problem, which is encountered in MDL. The proposed MEM is so flexible that it can be applied to both 2D and 3D scenarios. In terms of applications, we demonstrate the capability of using MEM to find correspondence points across 2D facial contours and therefore build up gender classifier automatically. The comparison with other automatic classification schmes shows that the MEM based classifier shows better performance in terms of overall classification accuracy, sensitivity and specificity. We also listed some future work directions in this chapter. I am hoping that with the help of this thesis, the automatic Active Shape Model building problem (i.e. correspondence problem) can, to some extent, be solved or making some progress towards the perfect solution.

Appendix

Data Format

This appendix section shows the datasets formats, which are used in the thesis. They are *meta*, *pts*, *fce*, vector and visualization.

Meta file is a common data format used in Insight Segmentation & Registration Toolkit (ITK www.itk.org).

Table A1. Meta

```
ObjectType = Scene
NDims = 3
NObjects = 1
ObjectType = Mesh
NDims = 3
ID = 0
TransformMatrix = 1 0 0 0 1 0 0 0 1
Offset = 0 0 0
CenterOfRotation = 0 0 0
ElementSpacing = 1 1 1
PointType = MET_FLOAT
PointDataType = MET_DOUBLE
CellDataType = MET_DOUBLE
NCellTypes = 1
PointDim = ID x y ...
NPoints = 4002
Points =
0 -10.9759 1.50363 -1.73291
1 12.036 0.184908 -3.7455
2 -10.9527 1.55506 4.44425
3 12.098 0.294071 5.49359
4 2.00945 8.81335 -1.50159

……………

3999 -23.8896 -5.50267 -2.19734
4000 -23.9489 -5.67883 -2.4453
4001 -23.8785 -5.77109 -2.63706
CellType = TRI
```

```
NCells = 8000
Cells =
0 12 0 31
1 582 31 32
2 582 12 31
3 13 12 582
4 583 32 33
5 583 582 32
6 584 582 583
7 584 13 582
...............
7994 275 4000 3999
7995 276 4000 275
7996 276 4001 4000
7997 277 4001 276
7998 277 543 4001
7999 11 543 277
```

It can be seen that the number of cells (N_c) and number of points (N_p) have the relations as follows:

$$N_c = 2N_p - 4 \tag{A.1}$$

Table A2. pts

```
15.3607    14.5341    22.7496
13.3707    36.1425    12.7825
15.4711    11.7889    15.1863
...............
13.8812    35.2222    8.36011
22.5363    25.9905    16.9296
23.6575    23.0942    12.8139
11.6803    21.1842    16.7248
```

The *pts* file is composed by point coordinates. Each line is a 3D coordinate for a landmark and number of rows is equal to number of points.

Table A3. fce

```
12  0   21
282 21  22
282 12  21
13  12  282
283 22  23
283 282 22

……………

284 282 283
284 13  282
14  13  284
285 23  24
```

The *fce* file shows the relations between points. For example, the first line of sample data shows that the 12th, 0th and 21st form a triangle surface.

Table A4 Visualization

```
NUMBER_OF_POINTS = 1002
DIMENSION = 1
TYPE = Scalar
0.000000

0.000999

0.001998

……………

1.000000
```

From the first point to the end point, each point is assigned with a value from 0 to 1.

Table A5 Variations Vector

```
NUMBER_OF_POINTS = 1002
DIMENSION = 3
TYPE = Vector
    0.024348     0.000863    -0.006797
   -0.028662    -0.003137     0.009808
    0.026247    -0.002523    -0.000842
```

```
-0.029640   -0.000341    0.003859

 ……………

-0.000846   -0.000448   -0.004895
 0.001895   -0.001639   -0.003297
```

Each node is assigned with a vector. This vector can be with length 1.

References

[1] T. F. Cootes, D. Cooper, C. J. Taylor and J. Graham. A trainable method of parametric shape description. In Proc. British Machine Vision Conference pub. Springer-Verlag, pp 54-61, 1991

[2] R. H. Davies, T. F. Cootes, C. J. Twining and C. J. Taylor. An information theoretic approach to statistical shape modelling. In 12th British Machine Vision Conference, pp 1-10, 2001

[3] J. F. Canny. A computational approach to edge detection. IEEE Trans Pattern Analysis and Machine Intelligence, Vol 8(6): pp 679-698, 1986

[4] M. A. Walker, J. R. Highley, M. M. Esiri, B. McDonald, H. C. Roberts, S. P. Evans and T. J. Crow. Estimated neuronal populations and volumes of the hippocampus and its subfields in schizophrenia. Am J Psychiatry, Vol 159: pp 821-828, 2002

[5] J. B. A. Maintz and M. A. Viergever. A Survey of medical image registration. In U. Spetzger, H.S. Stiehl and J.M. Gilsbach (Eds.), Navigated Brain Surgery, pp 117-136, 1999

[6] M. Kass, A. Witkin and D. Terzopoulos. Snakes: Active contour models. In Proc. 1st CCV, pp 259-268, London, UK, 1987

[7] A. A. Amini, T. E. Weymouth and R. C. Jain. Using dynamic programming for living variational problems in vision. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 12(9): pp 855-867, 1990

[8] L. D. Cohen. On active contour models and balloons. Computer Vision, Graphics and Image Processing: Image Understanding, Vol 53: pp 211-218, 1991

[9] P. Radeva, J. Serrat and E. Mart. A snake for model-based segmentation. Proceedings of the Fifth International Conference on Computer Vision, pp 816-821, 1995

[10] F. L. Bookstein. Shape and information in medical images: A decade of the morphometric synthesis. Proceedings of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis, pp 2-12, 1996

[11] F. L. Bookstein and W. D. K. Green. A feature space for derivatives of deformations. Information Processing in Medical Imaging, pp 1-16, 1993

[12] L. Staib and S. Duncan. Parametrically deformable contour models. In CVPR, pp 98-103, 1989

[13] T. F. Cootes, C. J. Taylor, D. Cooper and J. Graham. Active shape models-training

and applications. Computer Vision and Image Understanding, Vol 61: pp 38-59, 1995

[14] C. R. Goodall. Procrustes methods in the statistical analysis of shape (with discussion). Journal of the Royal Statistical Society, Series B, 53, pp 285-339, 1991

[15] I. Jollife. Principal component analysis. Springer-Verlag, New York, 1986

[16] T. F. Cootes, G. J. Edwards and C. J. Taylor. Active appearance models. In H. Burkhardt and B. Neumann, editors, 5th European Conference on Computer Vision, Vol 2: pp 484-498, Springer, Berlin, 1998

[17] C. Broit. Optimal registration of deformed images. Ph.D. dissertation, University of Pennsylvania, 1981

[18] G. E. Christensen. Deformable templates using large deformation kinematics. IEEE Transactions on Image Processing, vol 5(10), 1996

[19] A. Pentland and S. Sclaroff. Closed-form solutions for physically based shape modeling and recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 13(7): pp 715-729, 1991

[20] C. Nastar and N. Ayache. Fast segmentation, tracking and analysis of deformable objects. In 4th International Conference on Computer Vision, pp 275-279, Berlin, IEEE Computer Society Press, 1993

[21] T. F. Cootes and C. J. Taylor. Active Shape Models - `Smart Snakes'. In Proc. British Machine Vision Conference. Springer-Verlag, pp 266-275, 1992

[22] R. H. Davies. Learning shape: optimal models for analysing natural variability. Ph.D. dissertation, University of Manchester, 2003

[23] R. H. Davies, C. Twining, T. F. Cootes and C. J. Taylor. A minimum description length approach to statistical shape modelling. IEEE Transactions on Medical Imaging, Vol 21: pp 525-537, 2002

[24] P. Besl and N. McKay. A method for registration of 3D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 14(2): pp 239-255, 1992

[25] L. H. Staib and J. S. Duncan. Boundary finding with parametrically deformable models. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 14(11): pp 1061- 1075, 1992

[26] Y. Wang and L. H. Staib. Boundary finding with correspondence using statistical shape models. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp 338-345, Santa Barbara, California, June 1998

[27] Z. Su, T. Lambrou and A, Todd-Pokropek. Independent component analysis based active shape model with spatial relations for finding correspondence, MIUA, Manchester, 2006

[28] V. Jain and R. (Hao) Zhang. Robust 2D shape correspondence using geodesic shape context. Proceedings of 13th Pacific Conference on Computer Graphics and Applications (Pacific Graphics) (short paper): pp 121-124, 2005

[29] A. Hill and C. J. Taylor. A method of non-rigid correspondence for automatic landmark identification. British Machine Vision Conference, pp 323-332, BMVA Press, 1996

[30] A. Hill and C. J. Taylor. Automatic landmark generation for point distribution models. 5th British Machine Vision Conference, pp 429-438, 1994

[31] A. C. W. Kotcheff and C. J. Taylor. Automatic construction of eigenshape models by direct optimization. Medical Image Analysis, pp 303-314, 1998

[32] H. H. Thodberg. A minimum description length approach to statistical shape modelling. Lecture Notes in Computer Science, Vol 2732: pp 525-537, 2003

[33] T. F. Cootes, C. J. Taylor. Active shape model search using local grey-level models: A quantitative evaluation. Proceedings of British Machine Vision Conference, BMVA Press, pp 639-648, 1993

[34] M. S. Floater and K. Hormann. Surface parameterization: a tutorial and survey. Advances in Multi-resolution for Geometric modelling, pp 157-186, 2002

[35] R. H. Davies, T. F. Cootes and C. J. Taylor. A minimum description length approach to statistical shape modelling. Proceedings of the 14th International Conference on Information Processing in Medical Imaging (IPMI), 2001

[36] A. Baumberg and D. Hogg. An adaptive eigenshape model. In D. Pycock, editor, 6th British Machine Vision Conference, pp 87-96, 1995

[37] M. G. Roberts, T. F. Cootes, E. M. Pacheco and J. E. Adams. Quantitative vertebral fracture detection on DXA images using shape and appearance models. Academic Radiology, Vol 14: pp 1166-1178, 2007

[38] H.H Thodberg and H. Olafsdottir. Adding curvature to minimum description length shape models. In 14th British Machine Vision Conference, Vol 2: pp 251-260, 2003

[39] C. E. Shannon. The mathematical theory of communication (parts 1 and 2). Bell System Technical Journal, Vol 27: pp 379-423, 623-56, 1948

[40] T. Cootes, A. Hill, C.J. Taylor, J. Haslam. The use of active shape models for

locating structuring in medical imaging. Image Vision Computation, pp 355-366, 1994

[41] D. Fisher. Iterative optimization and simplification of hierarchical clusterings. Technical Report, Vanderbilt University, TR CS-95-01, 1995

[42] L. Robert. Camera calibration without feature extraction. Int. conf. Pattern Recognition, pp 704-706, 1994

[43] M. Bichsel. Segmenting simply connected moving objects in a static scene. PAMI, Vol 16 (11): pp. 1138-1142, 1994

[44] R. H. Davies, C. Twining, T. F. Cootes, J. C. Waterton and C. J. Taylor. 3D statistical shape models using direct optimization of description length. In 7th European Conference on Computer Vision, Vol 3: pp 3-20, Springer, 2002

[45] P. T. Fletcher, C. Lu, S. M. Pizer and S. Joshi. Principal geodesic analysis for the study of nonlinear statistics of shape. IEEE Transactions on Medical Imaging, Vol 23: pp 995-1005, 2004

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

[47] B. Geiger. Three-dimensional modelling of human organs and its application to diagnosis and surgical planning. Ph.D. dissertation, INRIA, France, 1993

[48] Y. Wang, B. S. Peterson and L. H. Staib. Shape-based 3D surface correspondence using geodesics and local geometry. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Vol 2: pp 644-651, 2000

[49] A. Ericsson and K. Astrom, Minimizing the description length using steepest descent. Porceedings of British Machine Vision Conference, 2003

[50] David Bau IIII, and Lloyd N. Trefethen. Numerical linear algebra. Society for 2nd Industrial and Applied Mathematics, Philadelphia, 1997

[51] C. Brechbuhler, G. Gerig and O. Kubler. Parameterization of closed surfaces for 3D shape description. Computer Vision and Image Understanding, Vol 61: pp 154-170, 1995

[52] X. Gu, Y. Wang, Tony F. Chan, Paul M. Thompson and Shing-Tung Yau. Genus zero surface conformal mapping and its application to brain surface mapping. Proceedings of IPMI, pp 172-184, 2003

[53] X. Gu, Y. Wang, Tony F. Chan, Paul M. Thompson and Shing-Tung Yau. Genus zero surface conformal mapping and its application to brain surface mapping. IEEE Transactions on Medical Imaging, Vol 23(7), 2004

[54] T. Moller and B. Trumbore. Fast, minimum storage ray-triangle intersection. Journal of Graphics Tools. Vol 2(1): pp 21-28, 1997

[55] J. Arvo. Fast random rotation matrices. In Graphics. Gems III. David Kirk, editor, pp 117-120, Academic Press, 1992

[56] J. P. W. Pluim, J. B. A. Maintz and M. A. Viergever. Mutual information based registration of medical images: a survey. IEEE Transactions on Medical Imaging, 2003

[57] D. Rueckert, P. Aljabar, R. A. Heckemann, J. V. Hajnal and A. Hammers. Diffeomorphic registration using B-spines. Medical Image Computing and Computer Assisted Intervention, 2006

[58] W. R. Crum, C. Tanner and D. J. Hawkes. Anisotropic multi-scale fluid registration: evaluation in magnetic resonance breast imaging. Physics in Medicine and Biology Vol 50: pp 5153-5174, 2005

[59] S. Lin and Y. Ho. 3-D Active shape image segmentation using a scale model. Signal Processing and Information Technology. pp 168-173, 2006

[60] A. D. Brett and C. J. Taylor. A framework for automated landmark generation for automated 3D statistical model construction. In 16th Conference on Information Processing in Medical Imaging, pp 376-381, Visegr´ad, Hungary, 1999

[61] G. J. Edwards, C. J. Taylor and T. F. Cootes. Learning to identify and track faces in image sequences. In Proceedings of the 3rd International Workshop on Automatic Face and Gesture Recognition, pp 260-265, Japan, 1998

[62] A. Rangagajan, H. Chui and F. L. Bookstein. The soft assign procrustes matching algorithm. In 15th Conference on Information Processing in Medical Imaging, pp 29-42, 1997

[63] M. Uzumcu, A. F. Frangi, M. Sonka, J. H. C. Reiber and B. P. F. Lelieveldt. ICA vs PCA active appearance models: application to cardiac MR segmentation. Medical Image Computing and Computer-Assisted Intervention, PT 1 2878: pp 451-458, 2003

[64] W. E. Lorensen and H. E. Cline. Marching cubes: a high resolution 3d surface construction algorithm. Computer Graphics (SIGGRAPH 87 Proceedings), Vol 21(4): pp 163-170, 1987

[65] C. J. Twining, R. Davies and C. J. Taylor. Efficient construction of MDL statistical shape models. Medical Image Understanding and Analysis, 2007

[66] T. Heap and D. C. Hogg. Automated pivot location for the Cartesian-Polar hybrid point distribution model. In 7th British Machine Vision Conference, pp 97-106, BMVA Press, 1996

[67] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, Vol 2: pp 121-167, 1998

[68] C. M. van der Walt and E. Barnard. Data characteristics that determine classifier performance. Proceedings of 16th Annual Symposium of the Pattern Recognition Association of South Africa, pp 160-165, 2006

[69] P. Golland, W. E. L. Grimson, M. E. Shenton and R. Kikinis. Deformation Analysis for Shape Based Classification. Lecture Notes in Computer Science, Vol 2082: pp 517-530, 2001

[70] P. Golland. Statistical shape analysis of anatomical structure. Ph.D thesis, Department of Electrical Engineering and Computer Science, MIT, 2001

[71] H. H. Thodberg. Minimum description length shape and appearance models. Lecture Notes in Computer Science, Vol 2732: pp 51-62, 2003

[72] S. Baker, I. Matthews and J. Schneider. Automatic construction of active appearance models as an image coding problem. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 26(10), 2004

[73] R. Beichel, H. Bischof, F. Leberl and M. Sonka. Robust active appearance models and their application to medical image analysis. IEEE Transactions on Medical Imaging, Vol 24(9), 2005

[74] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling. Numerical recipes. Cambridge University Press, 1987

[75] B. Scholkopf, A. Smola and K. Muller. Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation, Vol 10(5): pp 1299-1319, 1998

[76] P. T. Fletcher and S. Joshi. Principal geodesic analysis on symmetric spaces: statistics of diffusion tensors. Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis, Springer Berlin / Heidelberg, pp 87-98, 2004

[77] Y. M. Wang and L. H. Staib. Elastic model based non-rigid registration incorporating statistical shape information. Medical Image Computing and Computer Assisted Intervention, Vol 1496: pp 1162-1173, 1998

[78] T. F. Cootes, S. Marsland and C.J. Twining. Groupwise diffeomorphic non-rigid registration for automatic model building. In 8th European Conference on Computer Vision, pp 316-327, 2004

[79] D. E. Goldberg. Genetic algorithms in search, optimisation and machine learning. Addison-Wesley, Wokingham, UK, 1989

[80] B. Scholkopf, A. Smola and K. Muller. Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation, Vol 10(5): pp 1299-1319, 1998

[81] P. T. Fletcher, C. Lu, S. M. Pizer, S. Joshi: Statistics of shape via principal geodesic analysis on lie groups. Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp 95-100, 2003

[82] R. Bowden, T. A. Mitchell and M. Sarhadi. Reconstructing 3d pose and motion from a single camera view. British Machine Vision Conference, pp 904-913, 1998

[83] V. Vapnik. The Natrual of Statistical Learning Theory. Springer Verlag, 1995

[84] M. Styner, G. Gerig, J. Lieberman, D. Jones and D. Weinberger. Statistical shape analysis of neuroanatomical structures based on medial models. Medical Image Analysis, Vol 7 (3): pp 207-220, 2003

[85] A. Dempster, N, Laird and D. Rubin. Maximum likelihood for incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Vol 39: pp 1-38, 1977

[86] J. M. Hammersley and P. Clifford. Markov field on finite graphs and lattices, 1971

[87] D. German. Random fields and inverse problems in imaging. Lecture Notes in Mathematics, Springer-Verlag, pp 113-193, 1990

[88] S. Z. Li. Markov random field modelling in image analysis. Computer Science Workbench Series, pp 323, 2001

[89] B. Horn and B. Schunck. Determining optical flow. Artificial Intelligence, Vol 17: pp 185-203, 1981

[90] R. J. Adrian. Particles imaging techniques for experimental fluid mechanics. Annual Review of Fluid Mechanics, Vol 23: pp261-304, 1991

[91] J. Weickert, A. Bruhn and C. Schn¨orr. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. Technical Report 82, Dept. of Mathematics, Saarland University, Saarbr¨ucken, Germany, 2003

[92] J. Weickert and C. Schn¨orr. A theoretical framework for convex regularizers in PDE-based computation of image motion. International Journal of Computer Vision, Vol 45(3): 245-264, 2001

[93] J. Weickert and C. Schn¨orr. Variational optic flow computation with a spatio-temporal smoothness constraint. Journal of Mathematical Imaging and Vision, Vol 14(3): pp 245-255, 2001

[94] T. Brox, A. Bruhn, N. Papenberg and J.Weickert. High accuracy optic flow

estimation based on a theory for warping. In Computer Vision - ECCV 2004, T. Pajdla and J. Matas, editors, Lecture Notes in Computer Science, Vol 3024: pp 25-36. Springer, Berlin, 2004

[95] G. Langs, P. Peloschek and H. Bischof. Optimal sub-shape models by minimum description length. Computer Vision and Pattern Recognition, vol 2(20): pp 310-315, 2005

[96] P. Dalai, B. C. Munsell, W. Song, T, Jijun, K. Oliver, H. Ninomiya, Z. Xiangrong and H, Fujita. A fast 3d correspondence method for statistical shape modelling. Computer Vision and Pattern Recognition, pp 1-8, 2007

[97] J. Karlsson and A, Ericsson. A ground truth correspondence measure for benchmarking. International Conference on Pattern Recognition, vol 2(1): pp 568-573, 2006

[98] M. Styner, K. Rajamani, L. P. Nolte, G. Zsemlye, G. Szekely, C. Taylor and R. Davies. Evaluation of 3d correspondence methods for model building. Information Processing in Medical Imaging Conference, pp 63-75, 2003

[99] M. G. Roberts, T. F. Cootes, E. M. Pacheco, and J. E. Adams. Quantitative vertebral fracture detection on DXA images using shape and appearance models. Academic Radiology, vol 14: pp 1166–1178, 2007

[100] V. Jain and H. Zhang. Robust 3D Shape Correspondence in the spectral domain. Proceeding of International Conference on Shape Modelling and Applications, pp 118-129, 2006

[101] T. Heimann, I. Oguz, I. Wolf, M. Styner and H. P. Meinzer. Implementing the automatic generation of 3D statistical shape models with ITK. Medical Image Computing and Computer Assisted Intervention Open Science Workshop 2006

[102] S. Xu, R. H. Taylor, G. Fichtinger and K. Cleary. Lung deformation estimation and four-dimensional CT lung reconstruction. Lecture Notes in Computer Science, vol 3750: pp 312-319, 2005

[103] A. Khamene, J. K. Warzelhan, S. Vogt, D. Elgort, C. Chefd'Hotel, J. L. Duerk, J. S. Lewin, F. K. Wacker and F. Sauer. Characterization of Internal Organ Motion Using Skin Marker Positions, pp 526-533, 2004

[104] M. Spiegel and L. Stephens. Schaum’s Outline of Statistics, Schaum, 1998.

Publication List

Z. Su, T. Lambrou, A. Todd-Pokropek, "Independent Component Analysis Based Active Shape Model with Spatial Relations for Finding Correspondence", Medical Image Understanding and Analysis, July, Manchester, UK, 2006

Z. Su, T. Lambrou, A. Todd-Pokropek, "A Minimum Entropy Approach for Automatic Statistical Model Building", IEEE 5th International Special Topic Conference on Information Technology in Biomedicine, 26-28 November, Ioannina, Greece, 2006

Z. Su, T. Lambrou, A. Todd-Pokropek, "Principal Geodesic Analysis for the study of Nonlinear Minimum Description Length Approach", Medical Imaging and Informatics, Beijing, China, 2007 (The same proceeding also appears on Lecture Notes on Computer Science 2007)

Z. Su, T. Lambrou, A. Todd-Pokropek, "Correspondence points identification by using a minimum Entropy approach", Medical Image Understanding and Analysis, Aberystwyth, UK, 2007

Z. Su, T. Lambrou, A. Todd-Pokropek, "3D Minimum Entropy Model in finding correspondence automatically", (Preparation in Submission to A Journal)

Publication Samples

Independent Component Analysis Based Active Shape Model with Spatial Relations for Finding Correspondence

Zihua Su, Tryphon Lambrou, Andrew Todd-Pokropek

Department of Medical Physics & Bioengineering, University College London,
Malet Place Engineering Building, Gower Street, London WC1E 6BT, U.K.

Abstract. Statistical shape models use Principal Component Analysis (PCA) to describe the shape variations. However, PCA has the restriction that the input data must be drawn from a Gaussian distribution, and is only able to describe global decomposition. In recent years, Independent Component Analysis (ICA) has become a popular alterative for shape decomposition. Due to the local variations that ICA represents, the final optimal result usually turns out to be an invalid shape. In this paper, we will investigate the details of the ICA-ASM. With the consideration of the influence from neighbourhood points by using Markov Random Field (MRF), we overcome this drawback introduced by ICA. Our initial results show that our proposed method offers a better rate in obtaining a valid shape. From this, we can conclude that the MRF based ICA model provides improved results to the Bayesian based ICA model currently used.

1 Introduction

Geometric shape information plays a key role in many computer vision and image processing applications, especially in medical image analysis where many anatomical structures and organs can be identified and classified in terms of their unique shapes. The correspondence is such a critical thing that usually comes before Procrustes alignment.

The advantage of Active Shape Model (ASM) is that it uses experiences from training data to judge the correspondence and shape outline. In recent years ICA has been introduced into ASM for its excellent performance on giving more accurate local variations and no restrictions on data set. Actually, ICA has become a more general data description method than PCA does. By using the Bayesian frame work [1], we can cast our problem into a Maximum a Posterior Probability (MAP) work. The final result shape is the optimal minimization. The problem is by using ICA your optimization method will face more local peaks than PCA does. Therefore, the result is more likely to be trapped in invalid shapes. We noted that some work has been done for eliminating this effect. By adding more artificial training set Wang et al [1] make their shape model more rigid and more global. To some extent, they solve the problem of invalid shape, but that makes it harder to capture local variations. By using MRF theory [2], our method successfully conquers this drawback. We also note that some other approach has been made by R.H. Davies et al [3].

2 Method

2.1 ICA-Based Active Shape Model

ICA is a more general description of data format than PCA, since PCA can only represent the orthogonal condition. An example is shown in Figure 1. PCA requires the data to be Gaussian distributed and in most of cases that does not hold [4]. In addition, ICA gives a very convenient advantage, which PCA can not easily have, that the joint probability of all the components is equal to the product of every component's probability.

$$P(X) = P(X_1)P(X_2)\ldots P(X_n) \tag{1}$$

In our proposed method we use similar ranking algorithm to that presented in [5]. After that, we select the first *t* components to cover 98% of all variations.

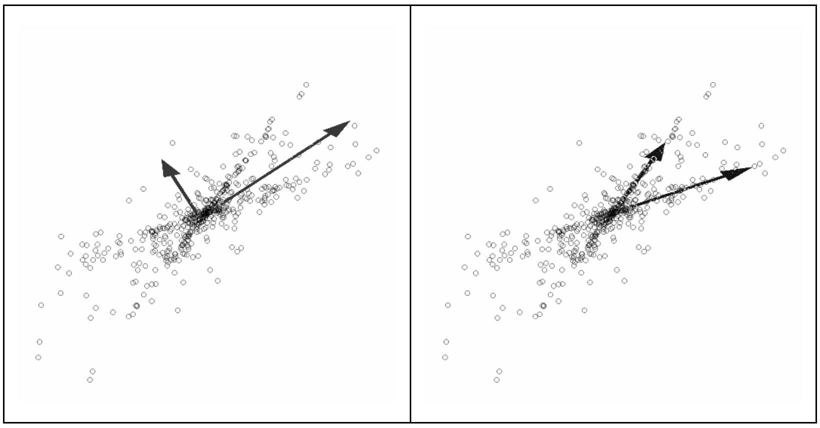
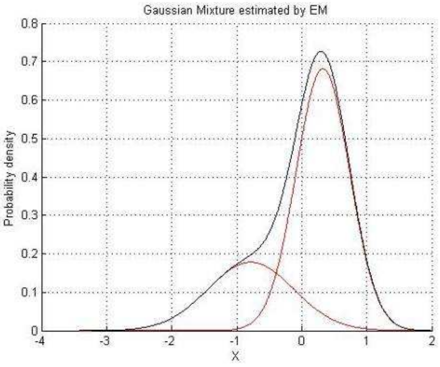


Figure 1. An example of PCA (left) and ICA (right), as we can see, ICA finds the real distribution format but PCA does not.

Since we know that finding the maximum negentropy direction is equivalent to finding a representation in which mutual information is minimized and negentropy is natural measure of non-Gaussianity. Here we propose a Gaussian mixture model for the ICA density expression. In this mixture model we are combing M parameterised densities and giving each one of them a weight. A frequently used algorithm for this optimization problem is the Expectation Maximization (EM) algorithm [4]. EM is an iterative method that finds the maximum by choosing a new guess to maximize the lower bound. Some of the optimization results from the real training sets are shown in Figures 2 and 3.



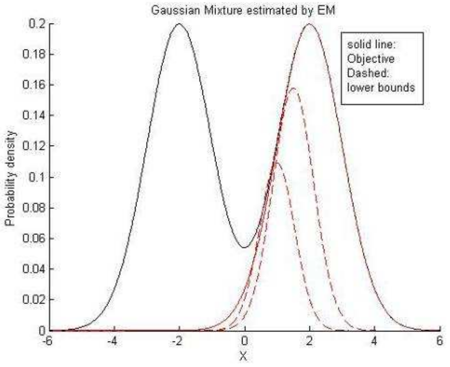


Figure 2. Gaussian Mixture estimated by EM.

Figure 3. Intermediate steps of the EM algorithm.

2.2 Markov Random Field Regularization

Since ASM was developed, research has been carried out on its parameter optimization. Different methods have been used in order to compose objective functions. Among these methods, Bayesian frame work is widely employed due to its usage of prior knowledge. However, lack of consideration in relations between parameters makes the final optimal shape often invalid. Using ICA to substitute PCA makes this even worse. Wang, et al [1] has tried to add some rigid artificial variations to the eigenvectors to make the shape changes more global and reasonable. This, to some extent, solves the problem, but makes the mode harder to capture local variations that often exist in practical cases. A natural way to incorporating spatial correlations into Bayesian process is to use MRF as a priori model. Thus, we follow the four successive stages of the Bayesian paradigm: 1) Construction of a prior probability distribution *p*(d) for the deformation field D matching the template shape S_t from training data to source shape S_s. 2) Formulation of an observation model *p*(y|d) that describes the distribution of the observed shaped Y given any particular realization of the prior distribution. 3) Using Bayes theorem to combine the prior and observation model into the posterior distribution. 4) Drawing inference based on the posterior distribution.

At this point we provide some definitions from MRF theory in order to describe the probability distribution on a spatial arrangement of points. Neighbourhood system and cliques are the most important definition in MRF theory. Given a graph of n connected sites $S = \{s_i\}_{i=1}^{n}$, a neighbourhood system $N = \{N_s, s \in S\}$ is any collection of subsets of S for which: i) $s \notin N_s$, and ii) $r \in N_s \Leftrightarrow s \in N_r$, then N_s are the neighbours of s. A clique c is a subset of sites S for which every pair of sites is neighbour. Let all cliques be denoted by C. For all $c \in C$ we assume that we have a family of potential function V_c.We may now define an energy function of any given configuration of *d* i.e. $U(d) = \sum_{c \in C} V_c$ .This leads to the definition of Gibbs measure. The Gibbs measure induced by energy function $U(d)$ is

$$p(d)=\frac{1}{z}e^{(-U(d)/T)} \qquad (2)$$

where, Z is the partition function and T is a parameter referred to as temperature. The Gibbs measure maximizes entropy (uncertainty) among all distributions with the same expected energy. The temperature controls the "peaking" of the density function; normally we define it as 1 for simplicity. The normalizing constant may be impossible to obtain due to the problem of dimensionality but often we need only ratios of probabilities and the constant omitted. A theoretical result called Hammersley-Clifford gives the relation between MRF and Gibbs random fields and states that D is a Markov random field with respect to *N* if $p(d)$ is a Gibbs distribution with respect to *N* [7][8]. So, we need to specify potentials that induce the Gibbs measure in order to encompass MRF properties of *D* on the graph. More details are given in [2].

2.3 Prior Distributions

We construct energy function based on differences between neighbouring sites. We put this in a multivariate case then we have the general expression of energy governing the site-priors

$$U_{site}(d)=\sum_{i\sim j}\left\|d_i-d_j\right\|_p \qquad (3)$$

where, $\|\cdot\|_p$ is the p-norm, *p=2* in the 2 D case, and d_i represents the multivariate displacement of the *i*th site.

With *p=2* the energy function induces a Gaussian prior on the deformation fields. Neglecting regions with strong surface dynamics the local optimization becomes concave and the maximum likelihood estimate of the displacement at the *i*th site is taken as the mean of the neighbouring displacements. Given the statistical models (ICA-ASM) and the shape parameters $a=(a_1a_2.....a_t)^T$, and pose parameters: scale s, rotation θ, transportation T_x,T_y . The combined pose and shape parameter vector to be determined is represented by the following equation.

$$P=(s,\theta,T_x,T_y,a_1,a_2,\ldots,a_t)^T \qquad (4)$$

2.4 Observation Models

The observation model *p*(y|d) describes the conditional distribution of the observed data Y. By specifying an observation model we may favour a mapping that makes correspondence between regions of similar boundary properties. We propose only using the edge information in the input image, which is denoted as E here. The edge image E is assumed to consist of one of the deformed templates, t_p corrupted by additive white zero mean Gaussian noise with standard deviation σ_n, i.e. E=t_p+n. This leads to (similar as in [9]):

$$U_{ob} = \sum_{n=1}^{N} \sum_{m=1}^{M} E(x(p,n), y(p,n)) k_m \frac{1}{\sigma_{n,m}^2} \qquad (5)$$

where, *k* the template magnitude at any point which is assumed to be a constant and is chosen to be the maximum boundary response. *N* is the number of marks on the boundary.

The posterior equation is given by

$$p(d \mid y) \propto \exp(-U_{total} / T) \qquad (6)$$

where, $U_{total} = aU_{site} + (1-a)U_{ob}$, in which $a \in [0:1]$ weights the sensitivities of different parts. When applying simulated annealing the posterior is linked to the prior and the observation model by

$$p(d \mid y) \propto (p(y \mid d)) p(d))^{1/T} \qquad (7)$$

3. Results and Discussion

In our experiment, a collection of 28 slices of different brain MRI datasets was used, in which the corpus callosum were labelled manually. On each image, 36 landmarks were labelled. Since we did not have a large data set, a leave-one-out experiment is performed, by repeatedly training the shape model on 27 of the images and testing it on the remaining image. The start position is selected by differing from the mean shape, either on X or Y coordinates by 10 pixels. A comparison is made between ICA based Bayesian model and our proposed MRF based method. In Figure 4, it can be seen that our result shows the new shape model finds the boundary with correspondences more reasonably accurately and the result is not sensitive to the start position.

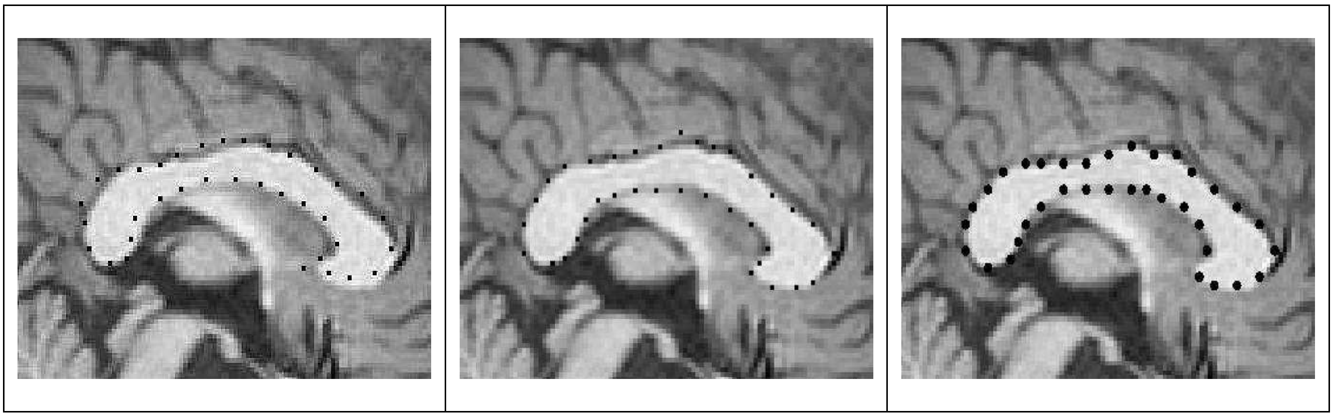


Figure 4. Three of the results: Bayesian ICA (left with small dot), Bayesian PCA （middle with small dot）and MRF (right with large dot). We can see that MRF achieves better performance.

In order to evaluate the two methods accurately, we calculate the overall displacement of searched shape to the manually labelled shape (Gold Standard) for each test image. The distance of two shapes is defined as the sum of all absolute distances between corresponding points. We calculate MRFASM (our proposed method) and BAYEASM (original ASM). Then we calculate the improvement (m)

$$m = \left(BAYEASM - MRFASM\right) / BAYEASM \qquad \textbf{(8)}$$

It is shown in Figure 5. The x- coordinate is the index of the test images, and the y-coordinate is its corresponding percentage improvement value. We can see from the figure that our method works better on all images. The points within the circle represent the cases where the Bayesian ASM converged. Comparing the equations provided in [1] and equation (6) of our methods, we can find that MRF introduces a stronger prior distribution by considering spatial relation between neighbour points. The use of this term is actually smoothing the cost function that can be the reason of the advantages.

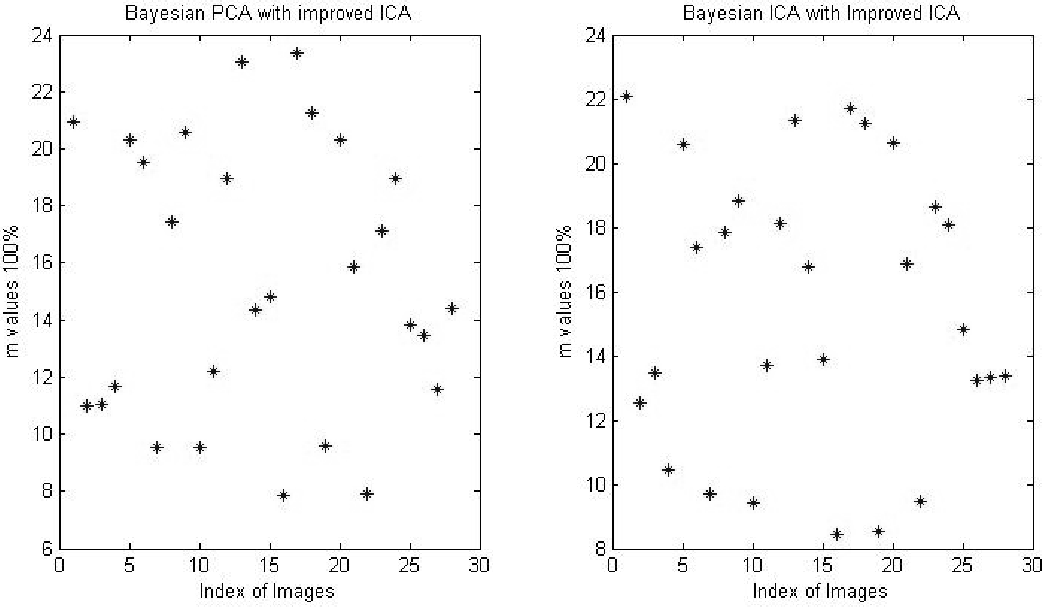


Figure 5. Shape to shape percentage distance difference, X direction is index of images; Y direction is percentage of improvement of ASM.

4. Conclusion

By using MRF, we give more restrictive relations between parameters to make our ICA-based model more easily converge onto a valid shape. Our initial results show that our proposed method offers a better rate in obtaining a valid and accurate shape. From this, we can conclude that the MRF based ICA model provides improved results to the Bayesian based ICA model currently used. Further work will include a larger dataset for the 2D case, as well as extending the technique into 3D, and different organs.

References

1. Y. Wang, L.H. Staib, "Boundary finding with correspondence using statistical shape models", Proceedings of IEEE, Conference on Computer Vision and Pattern Recognition, 338-345, 1998.
2. S.Z. Li, "Markov random field modelling in image analysis", Computer Science Workbench Series, 323, 2001.
3. R.H. Davies, C.J. Twining, T.F. Cootes, J.C. Waterton, C.J. Taylor, "A minimum Description Length Approach to Statistical Shape Modeling", IEEE Transactions on Medical Imaging, 21(5):525-537, 2002.
4. M. Uzumcu, A.F. Frangi, M. Sonka, J.H.C. Reiber, B.P.F. Lelieveldt, "ICA vs PCA Active Appearance Models: Application to cardiac MR Segmentation", Medical Image Computing And Computer-Assisted Intervention - MICCAI 2003, PT 1 2878: 451-458, 2003.
5. M. Uzumcu, A.F. Frangi, J.H.C. Reiber, B.P.F. Lelieveldt, "Independent component analysis in statistical shape models", Processing of the SPIE, Medical Imaging 2003: Image Processing, PTS 1-3 : 375-383 2003.
6. A. Dempster, N, Laird, D.Rubin, "Maximum likelihood for incomplete data via the EM algorithm", Journal of the Royal Statistical Society, 39:1-38, 1977.
7. J.M.Hammersley and P.Clifford, Markov field on finite graphs and lattices, 1971.
8. D. German, "Random fields and inverse problems in imaging", Lecture Notes in Mathematics, Springer-Verlag, 113-193, 1990.
9. L.H. Staib, J.S. Duncan, "Boundary finding with parametrically deformable models", IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(11):1061-1075, 1992.