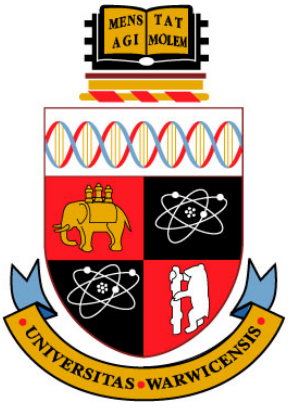


Statistical Shape Analysis for bio-structures: Local Shape Modelling, Techniques and Applications

by

Daniel Alejandro Valdés Amaro

A thesis submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Computer Science

University of Warwick, Department of Computer Science

October 2009

Dedicated to my family, specially to Carmen, Rosario, Blanca, Romulo, Rafael and Virgilio...

Summary

A Statistical Shape Model (SSM) is a statistical representation of a shape obtained from data to study variation in shapes. Work on shape modelling is constrained by many unsolved problems, for instance, difficulties in modelling local versus global variation. SSM have been successfully applied in medical image applications such as the analysis of brain anatomy. Since brain structure is so complex and varies across subjects, methods to identify morphological variability can be useful for diagnosis and treatment.

The main objective of this research is to generate and develop a statistical shape model to analyse local variation in shapes. Within this particular context, this work addresses the question of what are the local elements that need to be identified for effective shape analysis. Here, the proposed method is based on a Point Distribution Model and uses a combination of other well known techniques: Fractal analysis; Markov Chain Monte Carlo methods; and the Curvature Scale Space representation for the problem of contour localisation. Similarly, Diffusion Maps are employed as a spectral shape clustering tool to identify sets of local partitions useful in the shape analysis. Additionally, a novel Hierarchical Shape Analysis method based on the Gaussian and Laplacian pyramids is explained and used to compare the featured Local Shape Model.

Experimental results on a number of real contours such as animal, leaf and brain white matter outlines have been shown to demonstrate the effectiveness of the proposed model. These results show that local shape models are efficient in modelling the statistical variation of shape of biological structures. Particularly, the development of this model provides an approach to the analysis of brain images and brain morphometrics. Likewise, the model can be adapted to the problem of content based image retrieval, where global and local shape similarity needs to be measured.

Key words: Shape, Statistical Shape Modelling, Local Shape Models, Fractal Dimension, Markov Chain Monte Carlo, Metropolis-Hastings Algorithm, Hierarchical Shape Analysis, Curvature Scale Space, Diffusion Maps.

Contents

Acknowledgements

This work was possible through the support of the Consejo Nacional de Ciencia y Tecnologia (CONACyT), Mexico who granted me with a full PhD scholarship.

First, I would like to express my gratitude to my supervisor Dr. Abhir Bhalerao. Without his guidance and care, it would not have been possible to complete this research. He patiently answered questions and carefully explained everything to me. I am also grateful for his ideas and suggestions over the years, not to mention his support during the writing process. I would also like to thank Dr. Nasir Rajpoot for his advice and guidance through my PhD.

Special thanks go to Michal Komorowski, Hammad Qureshi and Thomas Popham for their contributions to the research process and sharing ideas. They patiently helped me when I needed some advice, explanation and even thesis corrections. I would like to thank as well Dr. Roger Packwood, Richard Cunningham and Rod Moore for helping me on technical and equipment problems.

All this would not have been possible without the support of my family in Mexico, to whom I am very grateful for all they did for me during my time in England, especially my parents Blanca and Virgilio, and my grandparents Carmen and Romulo, and my uncle Victor.

And of course I need to thank all my friends, who for four years were like a family for me: Daniel Claus, Michal Komorowski, Hammad Qureshi, Prem Fernando, Theodora Zarkadi, Yina Yuan, Agnieszka and Michal Rutkowski, Rosario Undurraga, Sarah Lim Choi Keung, Maurice Hendrix, Elisabetta Tondello, Melanie Reimann, Alberto Rebassa, Bernadette Martinez, Ritesh Krishna, Peter Krusche, Shanshan Yang, Nick Papanikolaou, Jop Vlaskamp, Pedro Nuno, Eugenio Rossi, Nicolas Blanco, Gabriel Penagos, Amanda Zamuner, Luis Hernandez, Chiara Soave, Jesus Cervantes, Soledad Betanzos, Miguel Valle, Emma Fernandez, Christian Voigt, Franciso Hernandez, Mariana Delgado, Juanjo Segui, Gabriel Tarriba, Emily Baker, Mario Tagle, Oliver Valiente, Hazel Mehmed Oztansel, Gustavo Inserra, Jorge Verissimo, Thomas Popham, Dveirel Kovalsky, Alejandro Garcia and many others. I would also like to thank my friends in Mexico for all the support provided from a distance: Erik Valdivieso, Juan Cruz and Pamela Enriquez.

Declaration

I declare, that except where acknowledged, the material contained in this thesis is my own work and that it has neither been previously published nor submitted elsewhere for the purpose of obtaining an academic degree.

Daniel Alejandro Valdés Amaro
October 2009

List of Figures

List of Tables

List of Algorithms

Acronyms and Notation

AAM	Active Appearance Model
AFDM	Adaptive Focus Deformable Model
ASM	Active Shape Model
aRI	adjusted Rand Index
CAT	Computed Axial Tomography
CCA	Canonical Correlation Analysis
CSF	Cerebrospinal fluid
CSS	Curvature Scale Space
DetCov	Determinant of the Covariance matrix
DM	Diffusion Maps
E-M	Expectation–Maximization
EEG	Electroencephalography
FASM	Fractal Active Shape Model
FD	Fourier Descriptor
$\mathcal{FD}$	Fractal Dimension
FEM	Finite Element Model
FFT	Fast Fourier Transform
fMRI	functional Magnetic Resonance Imaging
GPA	Generalised Procrustes Analysis
HSM	Hierarchical Shape Model
HVS	Human Visual System
ICA	Independent Components Analysis
ICP	Iterative Closest Point

IFS	Iterated Function Systems
kNN	k-nearest neighbour
LLE	Locally Linear Embedding
LOD	level-of-detail
LSM	Local Shape Model
MCMC	Markov Chain Monte Carlo
MDL	Minimum Description Length
MEG	Magnetoencephalography
M-H	Metropolis-Hastings
MLPV	Mislabeled Partial Volume Voxels
MRF	Markov Random Field
MRI	Magnetic Resonance Imaging
MSE	Mean Square Error
MSS	Manually initialised Subdivision Surface
NP-hard	Non-deterministic polynomial-time hard
PASM	Partitioned Active Shape Model
PCA	Principal Components Analysis
PDM	Point Distribution Model
PET	Positron Emission Tomography
PLSR	Partial Least Squares Regression
SM	Shape Model
SPHARM	Spherical Harmonics
SSA	Statistical Shape Analysis
SSM	Statistical Shape Model
SVD	Singular Value Decomposition

Symbols

a	State of a Markov chain.
$\mathbf{a}(n)$	Fourier coefficients.
$\mathfrak{a}$	Point in $\mathbb{R}^z$.
A	Random variable.
$\mathbf{A}$	Subset of a graph.
$\mathfrak{A}$	Set of points in $\mathbb{R}^z$.
b	State of a Markov chain.
$\mathfrak{b}$	Point in $\mathbb{R}^j$.
B	Subset of a graph.
$\mathfrak{B}$	Set of points in $\mathbb{R}^j$.
$\mathbb{C}$	The set of complex numbers.
$\mathbf{C}$	Covariance matrix.
$\mathcal{C}_i = (x_i, y_i)$	Contour, a set of n points in the plane.
CL_k	Cluster.
$card[\Theta_j]$	Cardinality of any partition Θ_j.
$cut(A, B)$	Measure of similarity between two clusters A and B.
d_{ii}	Element of a degree matrix $\mathbf{D}$.
$d\mathcal{S}_i$	Deviation from the mean.
$d_{\mathcal{M}}$	Geodesic distance on the manifold $\mathcal{M}$.
$\mathbf{D}$	Degree matrix.
$\mathcal{D}$	Dimension.
$\mathfrak{D}$	Denominator of the aRI performance measure.
$D(q_i, q_j)$	Minkowski distance.

$D^2_{\mathcal{G}}(\zeta_2, \zeta_1)$	Procrustes sum of squares.
$D_t(u, v)$	Diffusion distance at time t.
$dim_B F$	box-counting dimension or box dimension of F.
$dim_H F$	Hausdorff dimension for any set $F \subset \mathbb{R}^n$.
div	Divergence of the vector field.
e	Exponential function.
$\mathbf{e}$	Edge of a graph.
$\mathfrak{e}_j$	End point in a curve.
$E[\cdot]$	Expected value of a random variable.
$\mathbf{E}$	Set of edges.
$\mathbb{E}^n$	n-dimensional Euclidean space.
f	Function or mapping.
$f_\star(\mathcal{I})$	Space-filling curve.
$\mathbf{f}$	Feature vector.
$\|F_i\|$	*ith* Fourier coefficient.
$\|F_0\|$	The DC component.
$\mathbf{F}$	Nonempty subset of numbers in $\mathbb{R}^n$.
$g(i, \sigma)$	Gaussian function with standard deviation σ.
$\mathbf{G} = (V, \mathbf{E})$	Weighted graph.
$\mathcal{G}$	Group average configuration of the initial matrices.
$\mathcal{GP}$	Gaussian pyramid.
H	Nested structure partition of Q.
$\mathcal{H}^s(F)$	Hausdorff measure of F.
$\mathcal{I}$	Subset of $\mathbb{R}$.
i	Real number.
I	Identity matrix.
inf	Infimum, the largest value b in $\mathbb{R}^n$ such that for all a in a set F we have that $a \geq b$.
j	Real number.
J_n	Jordan content (area, volume) of a Jordan measurable subset of $\mathbb{E}^n$.

k	Real number.
$\mathbf{k}$	Kernel function.
$\mathbf{K}$	$n \times m$ matrix.
l	Real number.
$L(A\|B)$	Likelihood function.
$\mathbf{L}$ or $\mathbf{L}_{rw}$	Graph Laplacian (random walk).
$\mathcal{L}$	Laplace-Beltrami operator.
LB	Length of boundary of a curve.
$\mathcal{LP}$	Laplacian pyramid.
$\mathcal{LV}$	Laplacian vector.
m	Real number.
$\mathcal{M}$	A manifold.
$min\cdot$	The smallest value of a set, function, etc.
$max\cdot$	The largest value of a set, function, etc.
n	Real number.
n_{ij}	Number of observations in a group of a partition.
$\mathbf{N}$	Matching matrix for two partitions of the same data set.
$N(\delta)$ or N_δ	Smallest number of sets of diameter δ at most which can cover a subset.
o_{ji}	Feature (attribute, dimension, or variable).
$\mathbf{o}_i$	Input pattern.
$o(\cdot)$	Order of convergence.
$\mathbf{O} = \mathbf{o}_1, \ldots, \mathbf{o}_m$	Set of input patterns.
p	Entries of kernel P.
$\mathfrak{p}_m$	Posterior penalisation density.
$\mathbf{P}$	Markov matrix.
$P(\cdot)$ or $P(\cdot\|\cdot)$	Conditional probability of an event.
$\mathcal{P}$	Partition, a subset of ordered points along part of the shape.
$\hat{\mathcal{P}}_n$	Blended partition.
$\mathbb{P}(A)$	Prior distribution carried out to make an inference

	concerning A.
q	Density function.
$\mathfrak{q}_i$	d-dimensional object.
$\mathbf{Q}$	Eigenvectors matrix on the SVD.
$\mathcal{Q}$	Square in $\mathbb{E}^2$.
$r(i)$	Centroidal distance function.
$\mathbf{r}$	Distance vector.
r	Magnification factor.
$\mathbf{R}$	Rotation matrix.
$\mathbb{R}$	The set of real numbers.
$\mathcal{R}$	Regression parameter.
s	Real number.
$\mathbf{s}(\gamma, t)$	Diffusion distances accuracy term.
S	Procrustes problem equation.
$\mathcal{S}$	Shape, a subset of ordered points.
$\overline{\mathcal{S}}$	Mean shape from a set Ψ.
$\tilde{\mathcal{S}}$	Approximated shape after reconstruction.
sup	Supremum, the smallest value b in $\mathbb{R}^n$ such that for all a in a set F we have that $a \leq b$.
t	Real number.
$\mathfrak{t}$	Fractal dimension estimation.
$\mathbf{T}$	Transformation matrix.
T	$n \times 1$ Transpose vector.
u	Vertex of a graph.
$\mathbf{U}$	Matrix of eigenvectors.
v	Vertex of a graph.
V	Set of vertices.
$\mathbf{V}$	Matrix of eigenvectors.
w	Vertex of a graph.
$\mathbf{w}_{ij}$	Weight for the edge between vertex i and j.
$\mathbf{W}(\mathbf{u}, \mathbf{v})$	Weight matrix.

$\mathcal{W}$	Cube in $\mathbb{E}^3$.
x_i	Abscissa coordinate.
$\mathbf{x}$	Vector in $\mathbb{R}^n$.
$\mathfrak{x}$	Point in a manifold $\mathcal{M}$.
$X(i, \sigma)$	Convolved x-coordinates with function $g(i, \sigma)$.
$\dot{X}(i, \sigma)$	Convolved x-coordinates with the first derivative of the function $g(i, \sigma)$.
$\ddot{X}(i, \sigma)$	Convolved x-coordinates with the second derivative of the function $g(i, \sigma)$.
$\mathbf{X}$	$n \times m$ matrix.
$\mathcal{X}$	Random variable.
$\mathfrak{X}$	Data set.
y_i	Ordinate coordinate.
$\mathbf{y}$	Vector in $\mathbb{R}^n$.
$Y(i, \sigma)$	Convolved y-coordinates with function $g(i, \sigma)$.
$\dot{Y}(i, \sigma)$	Convolved y-coordinates with the first derivative of the function $g(i, \sigma)$.
$\ddot{Y}(i, \sigma)$	Convolved y-coordinates with the second derivative of the function $g(i, \sigma)$.
z	Real number.
$\mathbf{z}$	State of a Markov chain.
$\mathbf{z_i}$	Group from a given partition.
$\mathfrak{z}$	Point in a manifold $\mathcal{M}$.
$\mathcal{Z}_i$	Partition of $\mathbf{O}$.
$\circledast$	Convolution operator.
$\emptyset$	Empty set.
∞	Infinity.
∂	Partial derivative.
$\|\cdot\|$	Norm.
$\triangleq$	Equal to by definition.

$\alpha(a, \mathbf{z})$	Acceptance probability ratio in a Markov chain.
β	Location vector.
δ	Size of the self-similar structures needed to cover a fractal object in the calculation of the $\mathcal{FD}$.
Δ	Laplacian operator.
ϵ	Threshold, a number such that $\epsilon \in \mathbb{R}$.
ε	Complex k-vector of errors.
ε^*	Complex conjugate of the transpose of ε.
γ	Accuracy degree in the diffusion distances term.
κ	Curvature.
λ	Eigenvalue.
Λ	Matrix of Eigenvalues.
φ_k	Point on a contour.
μ	Mean vector.
∇f	Gradient.
ν	Translation vector.
$\omega(k)$	Window blending function.
Ω	Set of feature vectors.
π	Pi, constant value $3.1416\ldots$.
$\pi(A\|B)$	Posterior probability distribution of a random variable B that depends on a parameter A.
ϕ	Direction angle.
$\phi^*(l)$	The normalised net angular change in direction ϕ at each step around the perimeter (l) of a shape.
$\Phi = \mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_n$	Set of shapes.
ψ	Eigenvector.
Ψ	Matrix of eigenvectors.
$\mathbf{\Psi}$	A one-to-one map, or Homeomorphism.
ρ	Value that determines different distance measures that can be generated on the Minkowski distance.
$\varrho(u)$	Stationary distribution of the generated Markov chain.

σ	Width of the Gaussian function.
θ	Rotation angle.
Θ	Partition.
Υ	Design matrix.
ξ	Scale factor.
ζ	Point configuration.

Chapter 1

Introduction

1.1 Statistical shape analysis of biological and medical structures

Medicine has undergone a revolution with the use of new technologies, and it is the recent rapid technological developments which has increased the avenues for medical research. The role played by medical imaging has changed as well, as it is no longer used only as a simple way of visualisation of anatomic structures, but its role has expanded into finding a solution to the problem of extracting clinically useful information about structures. Today medical imaging is playing an increasingly important role in the diagnosis and treatment of diseases [82].

The study of the brain has many aspects. From the biological and computational, with the goal of understanding its function, to development of medical applications to unify knowledge of the brain across different areas. All this work is possible through exploration and experimentation in different ways. There are two main types of methods: invasive and indirect. Invasive methods such as autopsy allow research about the brain anatomy but not how it functions. Indirect measurements can be taken as well, using tests able to measure disease symptoms or mental performance. But it is relatively recent technological development which has increased the possibilities for neuro-scientific research, for instance, allowing

researchers to study the brain in vivo. Many non-invasive methods of brain observation are available, such as the EEG, MEG, MRI, fMRI, CAT or PET. (Figure 1.1).

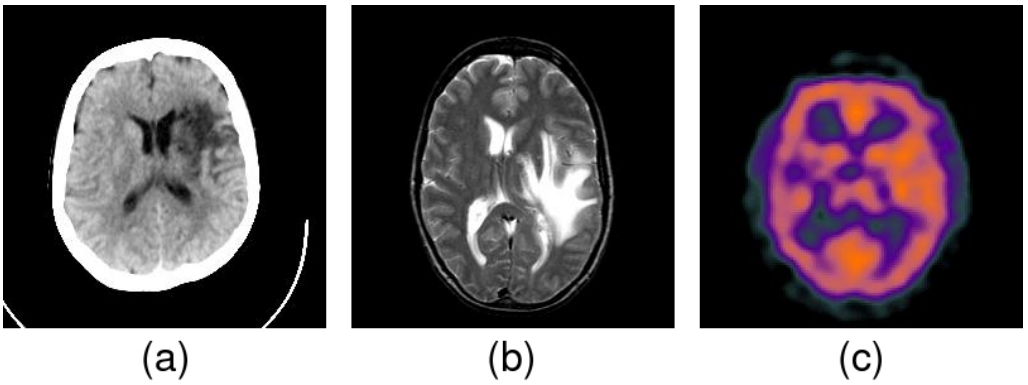


FIGURE 1.1: a) Example of CT (Roentgen-ray Computed Tomography), b) Example of MRI (Magnetic Resonance Imaging) and c) Example of SPECT/PET (Single Photon/Positron Emission Computed Tomography) [73]

In this way, brain imaging has empowered medicine and neuroscience with the ability to use images to identify structural brain changes associated to neurodegenerative diseases. Since brain structure is so complex and varies across subjects, methods to identify morphological variability can be useful for diagnosis, treatment and to assess the structural basis of normality and disease [8]. Methods that can capture the morphological variability of the human brain use mathematical models that are sensitive to changes in size, position, shape and tissue characteristics of brain structures affected by neurodegenerative diseases. Developments in Computer Science help as well to create new automated approaches and algorithms which avoid error-prone and labour-intensive manual measurements, offering precision in detecting brain differences. Such effort in developing algorithms and methods has been referred as Computational Anatomy.

Computational Anatomy emerged as a new discipline with the objective of creating algorithmic tools to help in the analysis of the human brain anatomy. But, as a new field in medical imaging, it does not cover all the fundamental features of the brain structure or function (in health and disease) due to brain complexity [63, 159]. One of the challenges of Computational Anatomy is the identification of

structural brain changes associated with different neuro-degenerative diseases, to bring valuable information in the diagnosis and treatment of various disorders [8]. A good example of this is the creation of *brain atlases* [82] to model the variability of neuro-anatomical structures across a population. Digital brain atlases transform the way of handling neuro-scientific information, making possible observations, such as, of how someone responds to a medication or how the brain changes with disease (Figure 1.2). Two main types of brain atlases are constructed: deformable and probabilistic. The idea behind the deformable brain atlas is that a digital brain atlas can be deformed to fit the anatomy of a certain subject. Transformations are used, so any segment is allowed to grow, shrink, rotate or twist in order to conform to the brain of the subject. As a by-product, the required transformations to warp an atlas into a new subject's brain provide an index of the anatomical shape differences between that subject's brain and the atlas [161]. Probabilistic brain atlases are created with information collected from imaging devices, and anatomical templates are generated. These type of atlases are able to capture how the brain varies with age, gender or disease. Likewise, they can be used for pathology detection in individuals or groups [161].

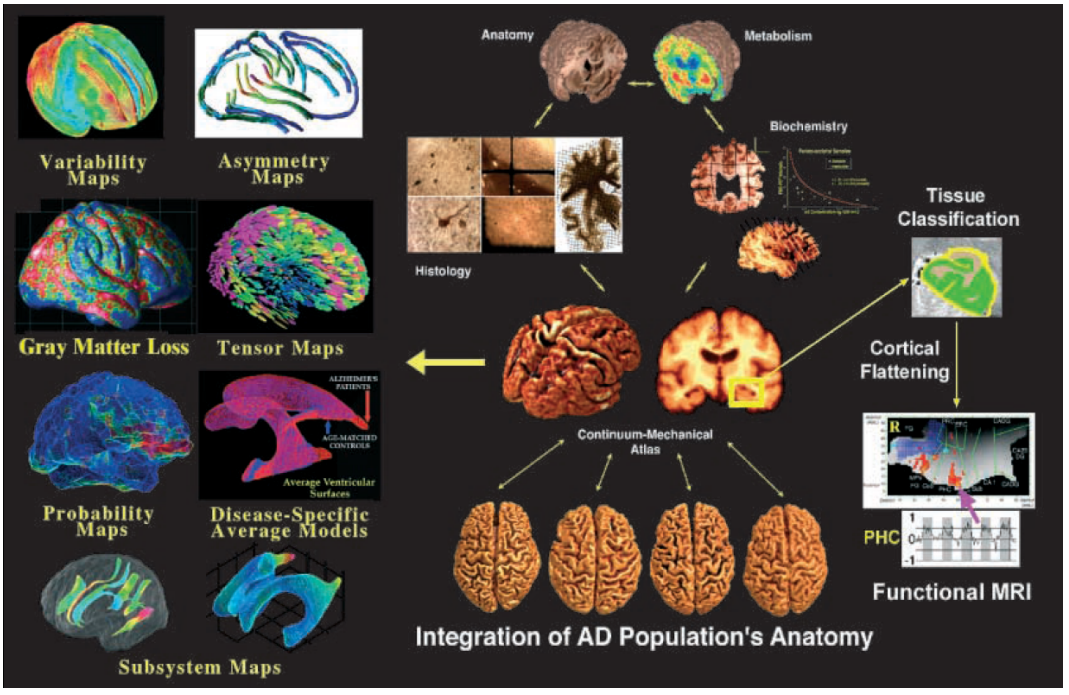


FIGURE 1.2: Elements of a disease-specific atlas: this schematic shows the types of maps and models contained in a disease-specific brain atlas [159].

Increasing evidence in neuroimaging research suggests that shape analysis of brain structures provides new information which is not accessible by the use of conventional volumetric measurements. This motivates development of novel morphometric analysis techniques in order to answer age old clinical research questions, but which have remained unanswered due to the lack of appropriate measurement tools. So for example, some of the challenges involved in the construction of these techniques are, the choice of biologically meaningful shape representations, robustness to noise and small perturbations, and the ability to capture the shape properties of populations that represent natural biological shape variation [58].

Statistical Shape Modelling can be applied to many tasks related to medical image analysis. They have proven that brain variation can be successfully captured. Worthy of mention is that many techniques have been created to model the surface of the brain such as [52] where a spherical topology mapping and topology correction are used to map accurately the cortex. [152] is another example that this can be assessed due to the effectiveness of capturing the variation of sets of contours (shapes). By this approach, a set of training shapes is used to build the model and then a given test-shape can be compared against the model. Another example of this can be found in [35], where fiber tracts extracted from Diffusion Tensor MRI are described and reconstructed using a shape template. Xue *et al.* [176] proposed an automatic segmentation algorithm for neonatal brain MRI using a knowledge based approach to identify and reduce mislabelled partial volume voxels (MLPV) in an Expectation-Maximisation Markov Random Field (EM-MRF) scheme. More recently, Rao *et al.* [127], have used Canonical Correlation Analysis (CCA) [102] and Partial Least Squares Regression (PLSR) to quantify and predict correlated behaviour in sub-cortical structures. In Hawkes *et al.* [65] a discussion and a detailed review of shape models in image-guided interventions can be found.

Deformable models are statistical shape models that combine geometry, physics and approximation theory. They have proven to be effective in segmenting, matching and tracking anatomic structures. These models can be interpreted as elastic bodies (templates) which react to applied forces and constraints [105]. The most

popular deformable model is known as 'Snakes' or *Active Contour Models* introduced by Kass et al. [75]. They impose physical constraints (using energy functionals) that control shape variations. The work of Cootes and Taylor [31] called *Active Shape Models* (ASM) or 'Smart Snakes' constrain the model deformation using statistics and are useful to track objects in a local image search [30]. Shen *et al.* [142], present a deformable model for segmentation and definition of point correspondences in brain images using an adaptive-focus deformable statistical model based on affine-invariant attribute vectors, minimisation of an energy function and Principal Components Analysis (PCA). Although, these type of models overcome many of the limitations of the traditional low-level image-processing techniques, there are still some problems that need to be solved. For instance, many of them are interactive models and need prior knowledge incorporated as initial conditions, or need human interaction with the model to guarantee good performance. A first review of shape methods can be found in [122], and more recently in [90]. Specifically, a survey for deformable methods can be found in [90]. Additionally, an overview of statistical methods based on the level-set theory and techniques is given in a survey by Cremers *et al.* [36]. In [105] a survey of deformable models in medical image analysis can be found. Finally, for a review and classification of statistical shape models for segmentation and other applications in medical image analysis refer to [68].

1.1.1 Fractal brain structure

Since Mandelbrot in his work *The Fractal Geometry of Nature* [98] proved that fractional dimensions were useful in the characterisation of natural phenomena, many studies have applied the framework of fractal geometry. So, too is the case in biology and medicine, where the complexity in the structures of living creatures resembles the principles of fractal geometry. For Weibel, an interesting conclusion on the importance of fractals in biological design is that it provides organisms with an important error tolerance, so the lack of a particular scale in biological evolution becomes an important advantage for fractal progeny [119]. But it is also plausible

that the fractal design provides the organisms with functional consequences as in [119] and [5].

Due to the complexity of human brain geometry it is plausible to assume that the brain possesses a self-similar structure [171]. To explain this, different studies have analysed brain processes and brain structures. For example, in [5] there is an analysis of the relationship of fractal patterns with dendrite and axon terminals, and they find that these 3D biostructures are fractal over at least one decade of length scales. In [183] there is a description of a method that detects age related white matter structural changes using fractal dimension, concluding that the human cerebellum is a highly fractal structure. A similar study in [53] performs a three-dimensional fractal analysis on the white matter surface to quantify its complexity. This study concludes that the white matter surface can be described by fractal dimension over a narrow and specific range of scales and that it measures the complexity of the surface. In [81] Kontos et al. provides a study of MRI activation patterns using space filling curves to relate brain structures with brain functions. The proposed approach is based on a mapping of the 3D space into 1D space using a transversal of the Hilbert space filling curve. Finally, in [79] a volumetric method based on the fast Fourier transform is used to address the question of whether the human cerebral cortex is self-similar, in a statistical sense. The analysis in this research was performed across a wide range of spatial scales, from the size of the whole cortex to the pixel. Results from six subjects confirm that the human cerebral cortex exhibits a fractal nature down to spatial scale of 3 mm.

Fractals and in particular space filling curves defy shape analysis because each example is, so to speak, its own exemplar and the relative complexity of such shapes: (1) make it hard to establish meaningful correspondences between shapes; and (2) require a much larger training size. Approaches that comprehend such ideas have been proposed, like in the aforementioned work by Vemuri and Radisavljevic [166] where fractal surfaces are generated. The importance of this relies on the fact that the generation of fractal surfaces of arbitrary order can be useful in the description of natural detail, such as one of their experiments with anatomical structures of

the brain (gyri and Hippocampus) from MRI data. As mentioned before, in [100] an approach called Fractal Active Shape Models (FASMs) is presented. Here the shape is represented by a fractal interpolation curve rather than a set of landmark points, but there is no use of fractal analysis in order to characterise the shape variability.

The fact that the brain seems to present a self similar structure at different scales is a key fact for this study since fractal analysis will be used here to analyse the shape structure of the brain and we will show how to use this in local shape modelling.

1.2 Local vs. Global

Work on shape modelling is constrained by many unsolved problems, for instance, difficulties in modelling *local* versus *global* variation. Global shape models result in highly non-linear shape spaces and it can be difficult to determine a compact set of modes of variation. It is important to point out that these models are successful in modelling large scale variations, but they struggle with the finer shape details. To cope with this problems hybrid shape models that combine local variation into the global scheme have been developed.

As an example of these type of models, in [166] a modelling scheme is presented based on multiresolution wavelet basis that endows it with the ability to continuously transform from local to global shape deformations. Another example of this is the work by Vemuri and Radisavljevic [166], where a deformable superquadric model that allows the generation of fractal surfaces is built. The deformable superquadric model is constructed in an orthonormal wavelet basis and empowers the model to continuously transform between local to global shape deformations. In [9] an approach called *deformation-based morphometry* is used to characterise differences in macroscopical anatomy among structural brain images. Active shape focusing [140] is another process where a multiscale active contour representation, in combination with an energy functional, is followed by an optimisation of the

model through a series of shape extractions, allows a study of shapes at different levels of detail. The method proposed by Yushkevich *et al.* [179] uses a multiscale medial representation and then two sets of features are computed: coarse features which describe relationships between neighbouring medial atoms and refinement features which measure geometrical differences between corresponding pairs of medial atoms. Then shapes are classified using Fisher linear discriminants. The method can analyse variability in the coarse-scale description of an entire object separately from the fine-scale variability in a part of the object. Shen *et al.* [143] presented a deformable model for segmentation and definition of point correspondences in brain images that incorporates geometric and statistical information in a hierarchical fashion. The important feature of this work is the incorporation of an attribute vector used to characterise the geometric structure of the model around a point of interest, from a local to a global scale. Neumann *et al.* [117] used a local method called negative symbols to compute a number of statistical and perceptual shape features for each connected component of an image and its background. It also incorporated a global method that uses a wavelet decomposition of the horizontal and vertical projections of the global image. In the end, their proposed method is based on the relative performances of these two methods, and it is applied to the problem of logo recognition.

In [78] a method for the analysis of 3D hippocampal shapes is presented. The method generates a hierarchical level-of-detail (LOD) using an octree-based scheme that allows it to discriminate the global shape difference and distinguish certain shape differences in specific local regions. In the work of Tsai *et al.* [163] a parametric model is derived by PCA to obtain a connection between multiple shapes within an image to capture co-variations among different shapes. In [22] a scale parameter is introduced in the active contour formalism using evolution equations (Polyakov functional) for active contours in scale spaces. The idea developed deals with the shape of objects at different scales of observation/resolution. Because at large scales the global shape of the object can be observed, since smaller shape features are suppressed, and at lower scales, finer characteristics appear in the

shape of the object, it seems natural to perform analysis not only at one scale but at several.

When using Active Shape Models two important issues normally arise. First, the requirement of homology, meaning the need for point-to-point correspondences. A key step in building shape models is the need to establish correspondences between shape boundaries over a reasonably large set of training images. It is important to establish correctly such correspondences, otherwise, an inefficient parameterisation of shape will be determined. The importance of finding the correct correspondences is even more evident in shape analysis, as new knowledge and understanding related to diseases and normal development is extracted based precisely on the establishment of correct correspondences [152]. In 2D, correspondence is often established using manually determined landmarks [19], but this is time-consuming and error-prone. Work has focused on this, like in [39] where a model for establishing automatically optimal correspondences between sets of shapes is described. This is done by casting the correspondence problem as a problem of finding an 'optimal' parameterisation of each shape in the training set, where the correspondences are found using the Minimum Description Length (MDL) principle. In [154] Styner *et al.*, a comparative study into three anatomical structures of four different correspondence establishing methods is presented. The studied methods include a Manually initialised Subdivision Surface (MSS) method and three automatic methods that optimise the object parameterisation: parametric surface description that uses spherical harmonics (SPHARM), Minimum Description Length (MDL) and the Determinant of the covariance matrix (DetCov) method. The results suggest that for modelling purposes the best of the studied correspondence methods are MDL and DetCov. More recent methods directly combine the search of correspondences and SSM, e.g. [186]. In this work the aim is to establish a pair-wise surface correspondence using a deformable model algorithm derived from the Adaptive Focus Deformable Model (AFDM). Lately, Hufnagel *et al.* [71] presented a work where in order to solve for inexact correspondences, a probabilistic approach based on an affine Expectation-Maximisation (EM) - Iterative Closest Point (ICP) registration algorithm is used. Here, the exact

correspondences are replaced by iteratively evolving correspondence probabilities which provide the basis for the computation of mean shape and variability model.

The second problem is the size of the training set and it happens that the homology problem makes it worse, since it is laborious and tiring to label manually homologous points for big samples. Moreover, the size of the training sample increases with the complexity of the shape, meaning with this that detail of shape variation will appear at higher modes, those associated with smaller eigenvalues of the PCA. This is not convenient since according to the model, usually most of the variation can be explained by a small number of modes, and additionally, this leads to another issue known as the flexibility of the SSMs. The power of the statistical model increases and falls with the amount of available training data. In the case of medical analysis the quantity is low as in practice there are rarely enough images to train the model. This leads to over-constrained models, meaning with this that the imposed restrictions on the deformations do not enable them to adapt accurately to new data [68]. But, the lack of training samples also implies that it is difficult to estimate a high-dimensional probability distribution of a shape from a relatively small number of samples. As a consequence, the subspace of "allowable shapes" spanned by the few available eigenvectors limits the capacity of the ASM to follow the finer details of the shape [38]. In order to overcome this problem in [33] there has been work that uses finite element methods to create vibrational modes of the shape training sets. These are used to generate a series of modified shape instances. Then a SSM is constructed including all the variants of the generated vibrational modes, which results in a model featuring original and synthetic variations. Depending on the given amount of training data, the model adapts the number of generated synthetic shapes. In a subsequent publication [29], the same authors present another technique for the same purpose, where the synthetic variation is added directly to the covariance matrix. A similar approach is presented in [169], to build flexible models where a mixed covariance matrix uses a combined prior of the smoothness and statistical variation modes. It is able to adapt gradually to use more statistical modes of variation as larger data sets

are available. In Lötjönen *et al.* [91] several different techniques to increase artificially the number of deformation modes were studied. They concluded that from their set, the best results are achieved by the use of a non-rigid movement strategy which deforms shapes randomly by local warping. In a more recent paper by Koikkalainen *et al.* [80], a comparison of different methods is performed and they conclude that results can be improved by introducing artificially generated deformation modes. In summary, the best techniques to introduce artificial variation were the nonrigid movement, and PCA and Finite Element Model techniques.

A different approach to increase model flexibility is to divide the SSM into independently modelled parts. The idea behind this is that smaller parts exhibit less variation, which can be easily captured using fewer training samples than the variation for the full shape [68]. In Davatzikos *et al.* [38] a hierarchical representation of shape in terms of its wavelet transform followed by a PCA on the wavelet coefficients is used. Here, the wavelet transform is used to organise their model into a hierarchy where the lower bands of the transform correspond to the global shape changes and higher bands to the local ones. Each band is modelled independently from the others. Another approach is given in [185]. In this approach, given a PDM, a mean mesh is partitioned into a group of small tiles. In order to constrain deformation of tiles, the statistical priors of tiles are estimated by applying PCA to each tile. Once again, each part or tile of the mesh is modelled separately, but the parameters for individual tiles are connected by curves in a combined shape space. More recently, Manousopoulos *et al.* [100] presented an approach called Fractal Active Shape Models (FASMs), an extension of the ASM using fractal interpolation. In the FASM framework, a shape is represented by a fractal interpolation curve rather than a set of landmark points. The proposed model has the advantage of requiring fewer landmark points, even for irregular shapes, and of using fewer variables for representing a shape.

In the context of shape analysis, a Local shape Model (LSM) refers to a mathematical construct that attempts to characterise the shape variation of parts from a shape, making emphasis on the local details. On the other hand, global models

deal with the whole structure with less relevance on the finer details of the shape in question.

A local SSM approach was proposed in [15] and it was demonstrated how local deviations from the fractal characteristics of space-filling curves could be analysed based on pose-alignment and an SSM. This work allows the pose variation between symmetrical parts of the curve to be eliminated and, in theory, allows the SSM to represent the interesting variation between parts. The model, namely *Contour Warplet*, cuts-up a given shape into a set of partitions and then each partition is modelled using Legendre polynomials. Next, a correspondence-free alignment is performed over the set of partitions into an affine shape space. Then, the Warplets are decomposed by PCA, and by retaining pose parameters, the entire contour can be reconstructed measuring the partition-to-partition variation or if necessary to drive a deformable template. An important conclusion of this work was that the localisation of the shape space to contour parts implies that periodic and repetitive structural variation can be modelled using a single training sample. *However, an open question of this work was what should be a good partition of a given contour into local parts and how can this be determined in a simple and efficient way.*

The method presented herein can be seen as complimentary to the work by Bhalerao and Wilson [15], which is similar to a PDM, except from the parameterisation in polar coordinates and the modelling of the partitions. The main objective of this research is to generate and develop a statistical shape model to analyse the local variation in shapes. It is important to mention that most of the applications of the PDMs have been useful for the creation of image segmentation methods. Here, a method for analysis of local variation based on PDM, i.e., the correlation of the shape in parts of certain structures, in this particular case natural shapes (leaves and squids) and brain structures is proposed. The model uses a combination of other well known techniques such as Curvature Scale Space (CSS) representation of shapes and Diffusion Maps for spectral shape clustering. Particularly, the aim here is to be able to partition the contour into a set of objects which are related by affine symmetry so the variations from these local symmetries can be identified. Within this particular context, this work addresses the question

of *What is local?* as well as helping to solve some of the problems of the SSMs such as the flexibility, that is directly related to the size of the training set problem.

1.3 Generating Local Shape Models

Here a method for local shape analysis is proposed, as sketched in figure 1.3. In general the method is based on the PDM, that will be described in Chapter 2, and works as follows: the input data consists of plane coordinates, so a source image (figure 1.3-(a)) is processed so the *contour* (2.16) (figure 1.3-(b)) can be drawn out of it, and if needed, it can be resampled to a desired number of points. Then, using the CSS process (figure 1.3-(c)) this contour is divided into local segments or *partitions* (figure 1.3-(d)). From here, it is possible to follow one of the two subsequent flows. One of them is to proceed with the *alignment* (figure 1.3-(e)) or **pose removal**, so that rotation and translation, namely rigid body transformations, are removed from each partition. To perform this, one of the segments should be selected as a reference one, and the others will be transformed with reference to this one. Then to perform the shape analysis a PCA analysis (figure 1.3-(f)) is used. The other option is to proceed to find a low-dimensional embedding of the sub-manifold using Diffusion Map clustering to identify a set of local shape models (figure 1.3-(α)) and then follow the aforementioned process to complete the shape analysis. All this is explained in more detail in the following sections.

1.3.1 Contour Localisation

By contour localisation we refer to the process of decomposing a whole contour into several parts. This is a common problem in areas such as machine vision and pattern recognition: breaking up (partitioning) information from the perceived world into coherent or meaningful parts. As most machine vision and pattern recognition systems involve some form of partitioning to simplify the analysis

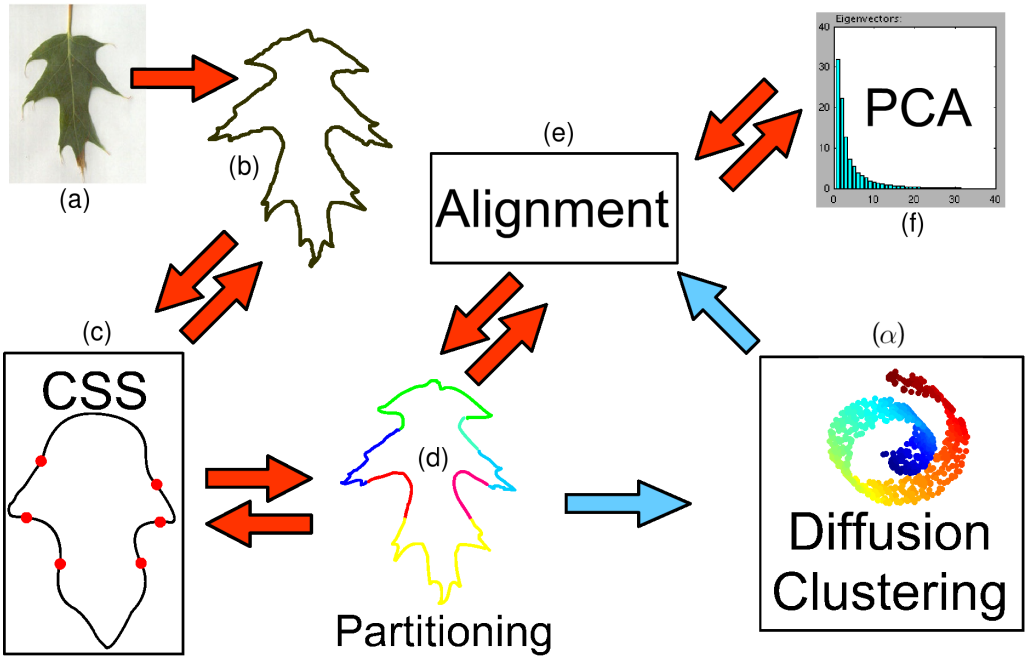


FIGURE 1.3: Process for our local shape model: **(a)** The source image is processed so the contour **(b)** can be drawn out of it. Then the CSS process **(c)** is applied in order to obtain the partitions of the contour **(d)**. From here there are two flows, first we can proceed with the alignment **(e)** and finish with the PCA analysis **(f)** or we can proceed to (α) and identify a set of local shape models.

process or minimise information storage, it is important to find a significant subset of the partitioning problem which may be solved using an algorithmic procedure. *Contour partitioning* can be defined as a computer process designed to decompose a contour or its interior into simpler parts [37].

Several methods for contour partitioning have been reported in the image analysis literature. One of the first research works was the Primal Sketch by Marr [103]. The Primal Sketch is a description of the grey-level changes present in an image. The description is expressed in a kind of vocabulary of intensity changes (edges, shades, lines, etc.). The modifying parameters are bound to the elements in the description, so they can specify elements like their position, orientation, or size. The importance of this descriptor relies on the fact that it can be regarded as a way to encrypt the information of the image, since the retained information corresponds very closely to the original image. So this means that in order to perform subsequent analysis, it is enough to read the primal sketch and not the

data from which it was computed, in this case the image [103]. Even though this is an image based technique its importance relies on the inspiration it has provided in the development of other methods.

Richards and Hoffman [129] developed shape descriptors known as Codons that are regarded as simple primitives to describe planar curves. Codons are basic representations based upon the singular points of curvature, in this case minima, maxima and zeros of curvature along the input curve. The aim of this descriptor is to segment a curve at concave (or minima of negative curvature) points in order to break the shape into parts. The study considers 2D objects having closed bounding contours (figure 1.4-(b)), but the presence of concavities in the silhouettes is used to infer boundary parts in 3D objects as well. A possible constraint of the method is that it needs to know the orientation of the boundary to determine the side of the figure (or conversely to know the figures' side to get the orientation), to determine the Codons. The determination of either side or orientation produces different results on the generated shape parts. It is not mentioned if the method is sensitive to noise or pose.

Asada and Brady [7] introduced a method based on a scale space approach to represent significant changes in curvature (figure 1.4-(e)). The approach proposes the creation of a set of curvature changes as primitives that are used to create a tree. Then the tree is parsed in order to sort the primitives according to the scale, providing a multi-scale interpretation of the contour. The representation is named Curvature Primal Sketch because the representation of the significant changes in curvature is analogous to the Marr's Primal Sketch representation of intensity changes in grey level images. The work presents sensitivity to noise, and efficiency is proven by using a set of tool contours (figure 1.4-(e)). No results are given on different sets such as the ones derived from natural shapes that might reflect a more complex construction. In addition, and as in other methods, there are no experiments regarding the sensitivity of the method to noise or pose.

Baruch and Loew [12] reported a method that uses the differential chain code of the edge to segment it into sections with constant curvature. Each section of

the boundary is assigned a code that represents the change in slope between the current and the previous section. To achieve this the first stage is the calculation of the discrete $\psi - S$ curve, denoted by $\psi(s)$. $\psi(s)$ is obtained by applying an integrating transformation to the chain code list derived from the boundary of a two-dimensional object. Then, the $\psi(s)$ curve is segmented by a recursive algorithm, that finds breakpoints that divides the curve into segments that are approximately linear. Results are presented on ideal shapes (circles and triangles) and a chromosome-like shape (figure 1.4-(a)). It is not clear by the results shown how the method would behave on other shapes, more complex or less linear. Even though the method exhibits low sensitivity to changes in orientation and noise presence, the performance on other data sets is unclear.

In [156] a method based on the detection of dominant points was introduced by Teh and Chin. The proposed algorithm is motivated by an angle detection procedure. Dominant points on a curve correspond to points of high curvature, so first an estimation of the curvature at each point on the curve is computed. Next, a two-stage procedure is applied to derive the dominant points. In the first stage a threshold is used to eliminate those points whose curvature is below such a threshold. In the second stage, a non-maxima suppression process is applied to the remaining points to eliminate those whose curvature estimates are not local maxima in a sufficiently large segment of the curve. The points remaining after the removal process are the dominant points. A comparison with other algorithms using different dominant point detection algorithms is presented, and results on standard shapes (such as figure 1.4-(a)) are given. The method seems to work better than the ones compared but no further mention of different applications or data sets is done, making this a limited approach. Emphasis is made of the robustness of the method under different scale conditions.

Furthermore, Wuescher and Boyer created a technique for partitioning contours into constant curvature segments [174]. The method is based on two ideas. The first is to filter some contour elements called "blips". Blips can be regarded as local single pixel displacements that represent a form of noise that are able to induce large local variations making it difficult to fit segments of constant curvature. The

second idea is to identify contiguous segments based on an overlapped voting and sequential extraction. Several result examples are provided with simple shapes, canonical shapes compared with other methods and stereo vision images (figure 1.4-(c)) to demonstrate the robustness of the method. Still, is not clear how this method might work with medical or biological image data sets.

More recently Cronin [37] proposed a method based on parsing concavity to partition digital contours into concave and convex sections. Such a method is based on the concavity code, that is derived from the chain code, and the algorithm consists of the implementation of rules to parse it. The set of six parsing rules are based on curvature and the location of the contour vertices. The result is a two-state boundary partitioning, where the points of high curvature are *promoted* as the points of interest to detect concave and convex sections of the portions of the contour remaining after the removal of concavities and convexities are regarded as the boundary residue. Even though the method was successfully tested on natural contours, such as fish contours (figure 1.4-(e)), it is not clear if the result of identifying convex and concave section on contours are enough to be applied to natural data, and likewise to other methods. No further description of its performance under noise conditions or pose variation is given.

Certain types of curves, e.g. fractals, can only be characterised by their local properties and are either repetitive arrangements of similar patterns or the result of applying a syntactical rule over a set of scales. Here, the aim is to decompose a contour into partitions according to this property to generate a proper set for local shape analysis (figure 1.5).

A *partition*, $\mathcal{P}_k$, of any given shape, $\mathcal{S}_i$ can be defined as a subset of ordered points along part of the shape

$$\mathcal{P}_k = \{(x,y)_{j+0}, (x,y)_{j+1}, \ldots, (x,y)_{j+m-1}\} \subset \mathcal{S}_i,\ 0 < m \leq n. \tag{1.1}$$

We assume that n is large enough to allow a reasonable piece-wise-linear approximation of the shape and that the partitions can capture local shape variation.

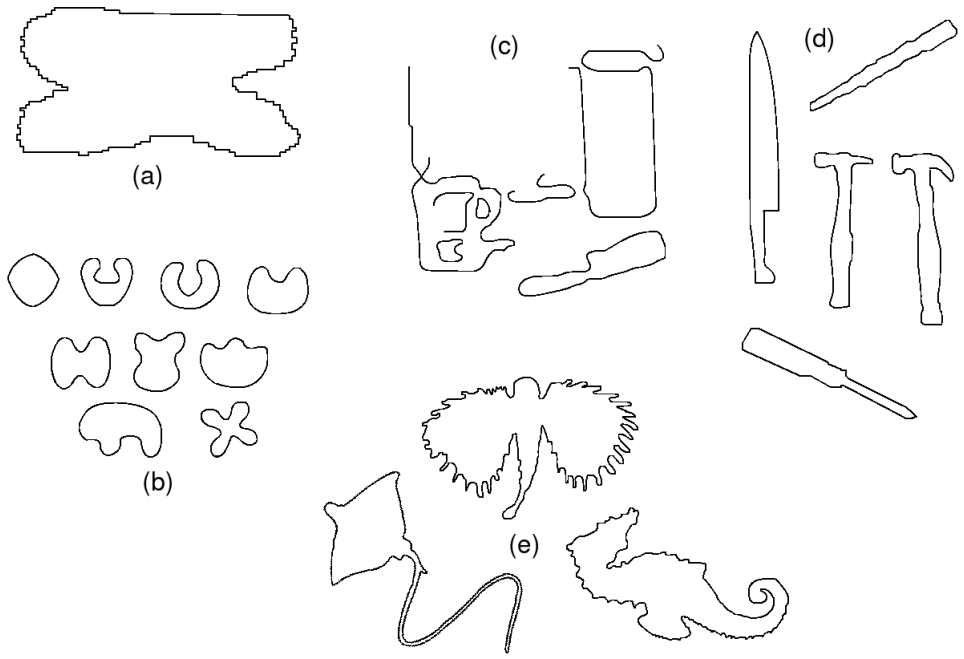


FIGURE 1.4: Some example of the figures used in previous approaches for contour partitioning: **(a)** Segmentation of two-dimensional boundaries using the chain code [12], **(b)** Codon constraints on closed 2-D shapes [129], **(c)** Robust contour decomposition using a constant curvature criterion [174], **(d)** Visualising concave and convex partitioning of 2D contours [37] and **(e)** Curvature Primal Sketch [7].

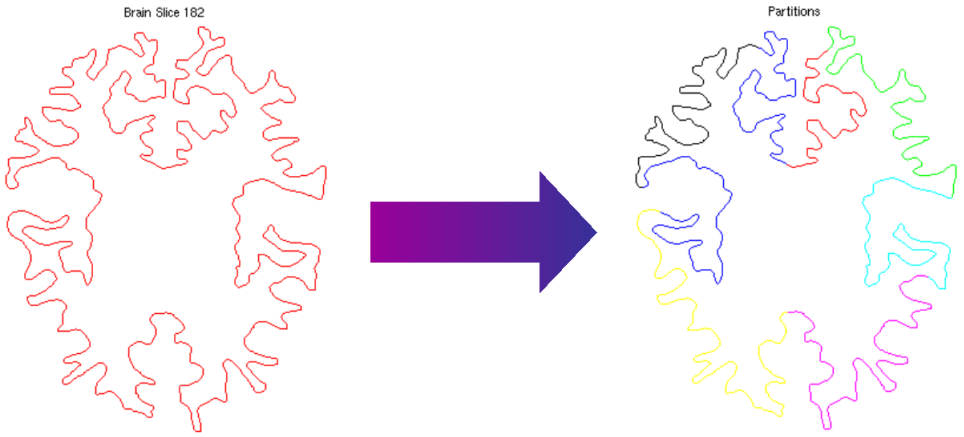


FIGURE 1.5: Decomposition of a brain contour into local segments according to specific properties.

Any partition is defined in $\mathcal{S}$ by picking its end points $\{\mathfrak{e}_{j+0}, \mathfrak{e}_{j+m-1}\}$ to be closest to two given points $\{\varphi_k, \varphi_{k+1}\}$. In this way, $\mathcal{P}_k$ is 'cut out' of a given $\mathcal{S}_i$.

1.3.2 Shape Clustering

Classification is a basic human conceptual activity. In general the problem of classification consists in assigning a class label to an object, a physical process or an event [165]. *Cluster analysis* is the generic name for a collection of procedures that can be used to create a classification. These form subsets of highly similar entities or *clusters* [4]. In each subset the resemblance between the objects is larger than with other objects in other subsets.

This introduces the problem of how to define the resemblance between the objects, and often the similarity or dissimilarity is assessed according to a proximity measure (or distance measure) between the objects [175]. First the *Minkowski distance* is introduced which depends on the value of ρ to generate different measures. Having two d-dimensional objects $\mathfrak{q}_i$ and $\mathfrak{q}_j$, it is denoted as:

$$D(\mathfrak{q}_i, \mathfrak{q}_j) = \left(\sum_{i=1}^{n} |\mathfrak{q}_{il} - \mathfrak{q}_{jl}|^{\rho} \right)^{1/\rho} \tag{1.2}$$

Then, when $\rho = 2$ the distance becomes the *Euclidean distance*:

$$D(\mathfrak{q}_i, \mathfrak{q}_j) = \left(\sum_{i=1}^{n} |\mathfrak{q}_{il} - \mathfrak{q}_{jl}|^{2} \right)^{1/2} \tag{1.3}$$

It is possible to obtain two other common special cases of the Minkowski distance: the city-block, also called *Manhattan distance* when $\rho = 1$,

$$D(\mathfrak{q}_i, \mathfrak{q}_j) = \sum_{i=1}^{n} |\mathfrak{q}_{il} - \mathfrak{q}_{jl}| \tag{1.4}$$

and the *sup distance* when $\rho \to \infty$:

$$D(\mathfrak{q}_i, \mathfrak{q}_j) = \max_{1 \leq l \leq n} |\mathfrak{q}_{il} - \mathfrak{q}_{jl}|. \tag{1.5}$$

The *Mahalanobis distance* is another metric and is defined as:

$$D(\mathfrak{q}_i, \mathfrak{q}_j) = (\mathfrak{q}_i - \mathfrak{q}_j)^T C^{-1} (\mathfrak{q}_i - \mathfrak{q}_j), \tag{1.6}$$

where $\mathbf{C}$ is the covariance matrix defined as $\mathbf{C} = E[(\mathfrak{q} - \mu)(\mathfrak{q} - \mu)^T]$, μ is the mean vector and $E[\cdot]$ calculates the expected value of a random variable.

In this work only five similarity measures have been presented, but examples and applications of these and others can be found in [175].

The problem of clustering arises when in many practical applications the training objects, or only a small fraction of them, are not labelled. In these cases, the structure of the data needs to be discovered without the help of labels. Clustering techniques are generally classified as hard partitional and hierarchical [175]. Hard partitional clustering attempts to divide data points into some prespecified number of clusters without any hierarchical structure. On the other hand, hierarchical clustering groups data with a sequence of nested partitions from single clusters to a cluster that includes all individuals. The following is a simple mathematical description. Given a set of input patterns $\mathbf{O} = \mathbf{o}_1, \ldots, \mathbf{o}_j, \ldots, \mathbf{o}_m$, where $\mathbf{o}_j = o_{j1}, o_{j2}, \ldots, o_{jn} \in \mathbb{R}^n$, with each measure o_{ji} called a feature (attribute, dimension, or variable):

1. Hard Partitional clustering attempts to search a K-partition of $\mathbf{O}$, $\mathcal{Z} = \mathcal{Z}_1, \ldots, \mathcal{Z}_k$ with $(k \leq n)$ such that:
   - $\mathcal{Z}_i \neq \emptyset, i = 1, \ldots, k$;
   - $\bigcup_{i=1}^{k} \mathcal{Z}_i = \mathbf{O}$;
   - $\mathcal{Z}_i \cap \mathcal{Z}_j = \emptyset, i, j = 1, \ldots, k$ and $i \neq j$
2. Hierarchical clustering attempts to construct a tree-like, nested structure partition of $\mathbf{O}$, $H = H_1, \ldots, H_l (l \leq n)$, such that $\mathcal{Z}_i \in H_s$,$\mathcal{Z}_j \in H_z$, and $s \geq z$ implies that $\mathcal{Z}_i \subset \mathcal{Z}_j$ or $\mathcal{Z}_i \cap \mathcal{Z}_j = \emptyset$ for all $i, j \neq i, s, z = 1, \ldots, l$.

Here the two classical types of clustering have been briefly introduced, nevertheless, there are more approaches that consider alternative ways to perform clustering

such as neural network-based, kernel-based, sequential data, large-scale data, and high-dimensional data clustering. Data observations with thousands of features implies working with high dimensions, as a consequence, applications require clustering algorithms that are able to process the data with more features than the number of observations, hence high-dimensional data clustering. This approach requires diverse algorithms to achieve the clustering. According to [175] the methods can be linear, non-linear, projected and subspace. The method used in this work belongs to the category of non-linear projection or non-linear dimensionality reduction algorithms. For a comprehensive review of these methods refer to [175].

Spectral clustering has become one of the most popular modern clustering algorithms. They are simple to implement, can be solved efficiently by simple linear algebra, and very often outperform traditional clustering algorithms. For example, K-means and learning a mixture-model using EM are methods based on estimating explicit models of the data, that provide high quality results when the data is organised according to the assumed models. However, when data is arranged in more complex way and there are unknown shapes, these methods tend to fail. Spectral clustering is shown to handle such structured data well since it does not require estimating an explicit model of a data distribution, rather it uses a spectral analysis of the matrix of point-to-point similarities [180].

Spectral methods for clustering use eigenvectors corresponding to the highest eigenvalues of a matrix derived from the distance between points. They are closely related to spectral graph partitioning in which the second eigenvector of a graph's Laplacian is used to define cuts over the graph. But this analysis can be extended to perform the clustering by building a weighted graph in which the nodes correspond to data points and the edges are related to the distance between the points[118].

Here, the idea is to use Diffusion Maps for non-linear, spectral clustering to build a set of linear shape spaces for statistical shape analysis (figure 1.6). The ideas, mathematical foundations, algorithms, application and examples will be given in Chapter 5.

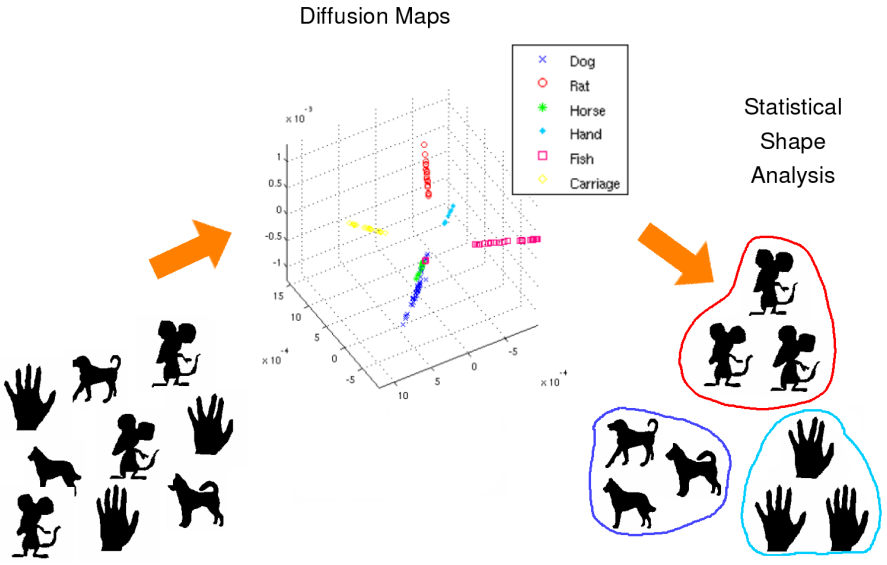


FIGURE 1.6: Diffusion Maps clustering is applied to find sets of linear shape spaces useful in the proposed local shape analysis.

1.4 Contributions

As described in section 1.2 the main objective of this research is to generate and develop a statistical shape model to characterise the shape variation of parts from a shape, making emphasis on the local details. The adoption of the PDM seems natural due to the fact that even though in image interpretation the use of rigid models is well established, in many practical situations objects that belong to the same class are not identical, so rigid models are inappropriate. This is particularly true in medical and industrial applications where the appearance of the objects can vary. In such cases flexible models or deformable templates can be used to analyse the degree of variability in the shape of the object in question.

In general, one can argue that the resulting model can be considered as an extension of the PDM adapted for local shape modelling. Representation and matching of the training set relies on the definition of a common origin from which correspondences can be established. Pose parameters are estimated by an affine alignment over a set of shapes, which is a form of General Procrustes Analysis (GPA) [60]. Then, PCA is subsequently applied to characterise the statistical shape variation. Finally, any shape of the set can be approximated using the mean shape and a

sum of the modes. Here, the proposed model cannot be considered by itself a contribution, but its extension that offers ways to cut-up any given input shape into a set of partitions and classify these according to spectral clustering to form meaningful local sets of shapes for analysis, can be regarded as one (figure 1.7).

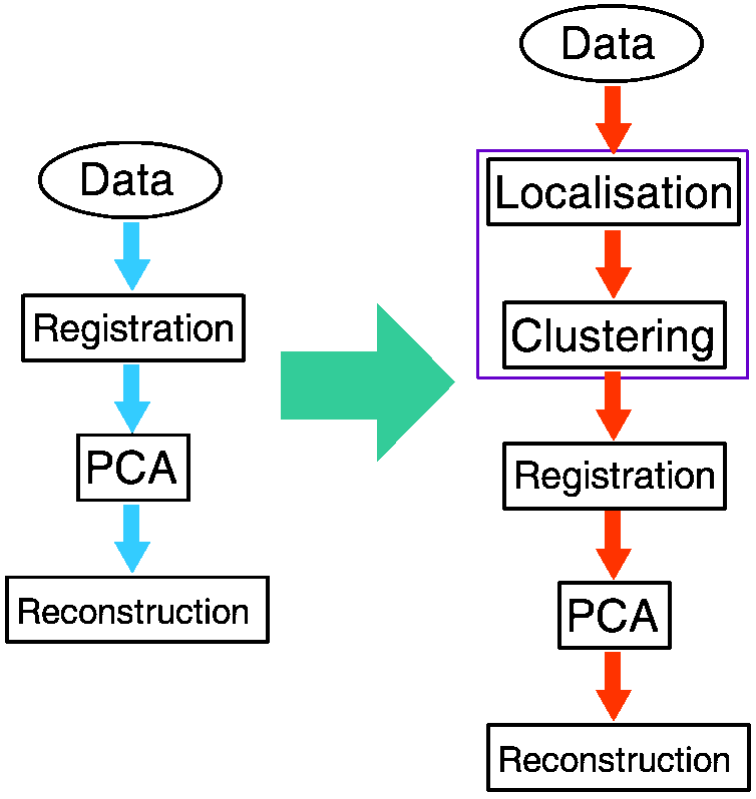


FIGURE 1.7: The proposed LSM (right side) uses a PDM (left side) to characterise the statistical shape variation and at the same time offers ways to cut-up any given input shape into a set of partitions. A classification method of these is performed according to the Diffusion Maps spectral clustering technique to form meaningful local sets of shapes for analysis. The additional features of the method are highlighted in the purple rectangle on the right.

Thus, based on the latter, the main contributions of this work are focused on the contour localisation and the clustering of those shapes generated by such methods. Hence, the **first contribution** is the creation of a method cast in a *Bayesian framework* combined with fractal analysis to estimate a set of contour partitions. In detail, a *Markov Chain Monte Carlo method* is used along with box counting dimension to characterise the *fractal dimension*. The idea behind this was to innovate and create a different method than previous endeavours (section 1.3.1)

by using the fractal dimension as the driving element for the algorithm, taking advantage of the *fractal nature of the brain* (section 1.1.1).

The **second contribution** is the creation of an alternative method for contour partitioning based on the utilisation of the *Curvature Scale Space descriptor*. No modifications were made to the original method, here just the evolution of curves is used to obtain a set of *zero-crossing points* that are used to *create consistent partitions* in contours. Despite other methods have used the CSS descriptor (section 1.3.1), this one is simpler yet effective in performing the task of obtaining unique solutions.

A method for *shape clustering* useful in shape analysis is presented as the **third contribution**. Such a model uses Spectral clustering (section 1.3.2) to create clusters of closed and non-closed contours in an unsupervised way. The selected spectral clustering method used was Diffusion Maps, and it outperforms the supervised clustering method devised in Chapter 4. Moreover, it is useful in solving the problem of choosing the prototypical shape used for the shape analysis (section 2.4.4).

Additionally, a novel *Hierarchical Shape Analysis* method based on *Gaussian and Laplacian pyramids* (Chapter 6) is presented as the **last main contribution**. The method is able to extract in a hierarchical fashion, the relevant shape variation and is used to compare the performance of the LSM.

1.5 Published Work

The work corresponding to chapters 3, 4 and 5 of this thesis has been presented in 3 peer-reviewed publications. A list of these publications may be found in Appendix A.

1.6 Equipment

All the experiments described in this thesis were run on an IBM IntelliStation Z Pro, with two Dual-Core Intel Xeon Processors at 2.1 GHz and with 2 GB of RAM. The operating system was Red Hat Enterprise Linux and for the developing of the algorithms MATLAB® v7.3.0.298 (R2006b) was used.

1.7 Outline of the thesis

Chapter 2 reviews basic concepts for this study, among them the concept of shape, shape spaces and shape analysis . The chapter focuses in particular in the concepts involved in statistical shape analysis, such as Procrustes methods or PCA, and describes the details of the construction of the PDM.

Chapter 3 presents a method for modelling fractal curves, and partitioning such curves into segments. Note that this chapter will present details on the utilisation of the properties that regard the brain as a fractal object (see chapter 1). The rest of the chapter is devoted to reviewing Bayesian statistics, with particular reference to Monte Carlo methods, Markov Chains and the Metropolis Hastings algorithm. Details of the implementation are given along with a discussion of the use for building local statistical shape models.

In Chapter 4, the first part states the principles of the CSS method, devised as a useful and natural way to set partitions on a contour so the rest of the chapter explains how the CSS representation is used in different ways in this research and an interactive tool was built for its use.

Next, Chapter 5 explores the use of spectral clustering concepts to develop a way of improving the clustering of data sets for use in shape analysis. The first part of this chapter describes how Fourier descriptors can be used to represent a given shape contour. Then it continues with the definition of manifolds, and with explanations of diffusion maps and how they can be used to build a shape model together with

the use of extrema points from a CSS. Results on different shapes are given along with a discussion of them.

In Chapter 6 a new Shape Analysis method is introduced. The method is based on Gaussian and Laplacian pyramids, concepts that are explained in the chapter, along with an explanation of the method to derive shape information out of Laplacian pyramids. Since the presented technique extracts and quantifies correlated behaviour between the different levels of the Laplacian pyramid it is named Hierarchical Shape Analysis. Additionally, this provides a useful method to compare the LSM introduced in this thesis by experiments on the compactness of the eigenvalues. Results are presented along with a discussion of how to extend this method.

Finally, in Chapter 7 conclusions are presented and at the same time the limitations of this work and further research opportunities are also discussed.

Chapter 2

Shape, Shape Analysis and Statistical Shape Analysis

2.1 Introduction

Statistical shape models have proven to be useful to study variation in anatomical shapes [30]. The purpose of this chapter is to introduce the background useful to understand the proposed shape model. The chapter focuses mainly on the concepts related to shape and its analysis. The first part defines important concepts of: shape, shape descriptors, shape spaces, landmarks and shape signatures. Next, two similar concepts are presented, Morphometry and Shape analysis but, since Morphometry can be regarded as an antecedent to Shape Analysis, it is worth mentioning in order to understand the differences. Statistical Shape Analysis is introduced, along with two important aspects of it, Procrustes methods and PCA. After this, Eigenshape analysis is presented as it exhibits similarities with the PDM, which is the last part of chapter. PDMs represent the core of this work since the proposed model is based on this. It is important to note that in this last section, descriptions of how the contours are extracted from images, how registration analysis is carried out, how the statistical shape variation is captured,

and how the shapes are reconstructed, is given. Finally, the chapter concludes with a summary.

2.2 Shape, Shape Descriptors and Shape Spaces

Shape is an important feature of the human visual system (HVS) as it is one of the basic features used to describe perceived objects, both natural and man-made. Technology has led to a collection of a great amount of visual information where there is a need to identify such objects in a wide variety of disciplines, commonly done by the use of the notion of shape. However, shape tasks such as recognition, representation and description are difficult to achieve. In the context of image processing, shape information extracted from the input data only represents the projected object in a partial way. Additionally, to make the problem more complex, shape is often corrupted with noise, distortion and occlusion.

One of the first studies to analyse mathematically the variation of shape was *On Growth and Form* by D'Arcy Thompson [158]. In this work, it is possible to see that the author was concerned with the explanation of biological growth and form in terms of mathematics and physics. The book covers a wide range of morphological studies and constitutes a study of the laws governing the dimension of organisms and their growth, the statics and dynamics in cells and tissues including the phenomena of geometrical packing, membranes under tension, symmetries, and cell division. As well, it deals with the engineering and morphology of skeletons in simple organisms. Thompson conceived the form as a product of dynamic forces that are shaped by flows of energy and stages of growth. The last chapter on transformations is perhaps the most remarkable contribution to the shape analysis field. It shows diagrams of how differences in the forms of, for example, species of fish or different bones from animals, can be construed in terms of distortions of the co-ordinate systems onto which they are mapped. It is important to remark that most of the work in shape analysis is heavily influenced by D'Arcy's seminal work.

Dryden and Mardia discuss shape [45] as the geometrical information that stays when location, scale and rotational effects are filtered out from a particular object. Small complements this idea by further adding that the remaining information should be invariant to transformations [145] (Figure 2.1).

In general, an *affine transformation* is a composition of rotations, translations, dilations, and shears. An affine transformation is any transformation that preserves collinearity and ratios of distances. Let $\mathbf{A}$ be a $n \times m$ matrix and $\mathbf{x}$ and $\mathbf{y}$ are $n \times 1$ column vectors, then an affine transformation has the form [145]:

$$\mathbf{x} \mapsto \mathbf{A}\mathbf{x} + \mathbf{y} \tag{2.1}$$

meaning that it consists of a linear transformation followed by a translation.

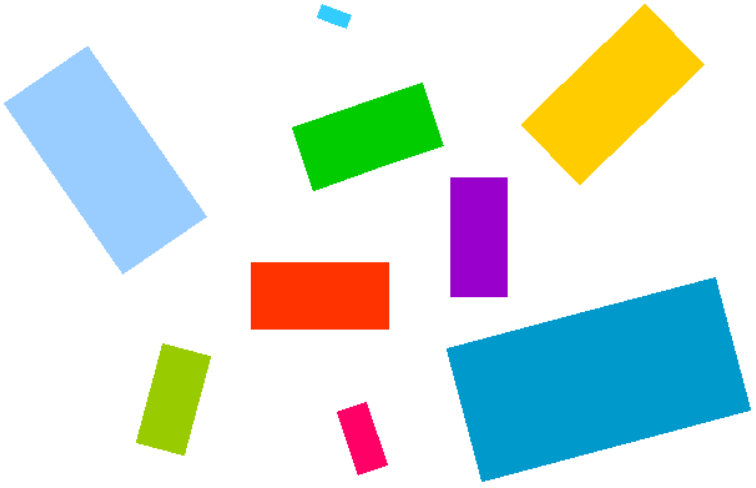


FIGURE 2.1: The concept of shape is invariant to transformations so for example, a rectangle's shape can be identified even if it is in different positions or present different sizes. Here rectangles are related by similarity transformations.

Therefore, in order to be able to analyse the objects from a scene this needs to be represented. Shape representations in general look for effective and perceptually important shape features that are based on either shape boundary information or interior content of the boundary [184]. At the same time, shape representation methods result in a non-numeric representation of the original shape so the important features or characteristics of the shape remain [90]. Another important concept is *shape descriptor* that, in general, can be regarded as a set of numbers

produced to represent a given shape feature. Such descriptors attempt to quantify the shape in ways that agree with human concepts of shape or with task-specific requirements [107]. Some of the basic descriptors are area, perimeter, compactness, eccentricity, elongation, rectangularly, orientation, variance or principal axes. Nevertheless, there are several ways to classify them; reviews, surveys and complete details of the descriptors can be found in [123, 90, 184, 107].

Shapes can be described by a series of finite points locating points on the sample to be analysed, called *landmarks*. According to Bookstein [19], the points at which one's explanations of biological processes are grounded are so called landmarks. A less intuitive definition [45] defines a landmark as a point of correspondence on each object that matches between and within populations (Figure 2.2). Dryden identifies three basic types of landmarks. An *anatomical landmark* is a point that corresponds between organisms in some biologically meaningful way. These are assigned by an expert. *Mathematical landmarks* are points located on an object according to some mathematical or geometrical property of the figure. *Pseudo-landmarks* are points constructed on an organism, located either around the outline or in between other landmarks, mathematical or anatomical.

Traditionally, the process of identifying landmarks is a labour-intensive part of shape analysis techniques since presently this is usually done manually. Consequently, attempts to develop completely automated and reliable systems to identify landmarks in digital images have been an important goal of research, as in [69, 41, 72, 135, 121, 67].

Bookstein defines the *shape space* as the set of all possible shapes in the orbit space $\sum_k^m$ of the non-coincident k point set configurations in $\mathbb{R}^m$ that is under the action of the Euclidean similarity transformations [19]. For example, when there is a set of k labelled points in the Euclidean space $\mathbb{R}^m$, the centroid of the points can be considered as the origin. Then the described basic object will be called a *pre-shape* and any two configurations of the labelled points will be considered as having the same shape if either of their pre-shapes can be transformed into another by a rotation about the shared centroid. Therefore the shape space can be considered

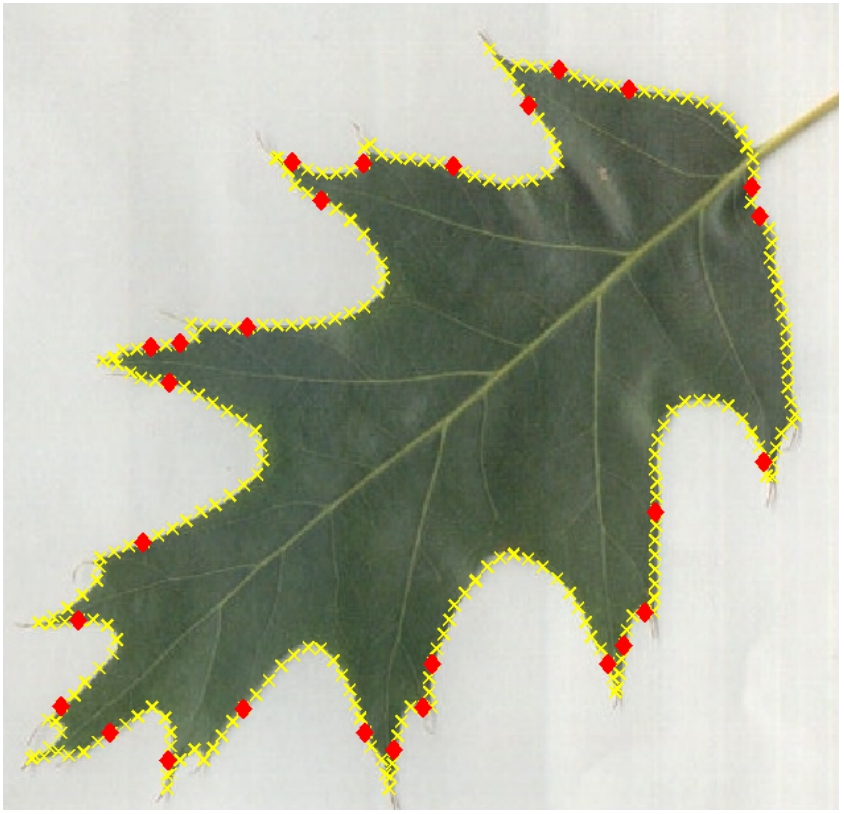


FIGURE 2.2: A leaf with mathematical landmarks in red and pseudo-landmarks in yellow.

as the collection of all possible shapes [77]. The fundamental space for this work is Kendall's shape space, since it provides a complete geometric setting useful for the analysis of Procrustes distances among sets of landmarks [44]. Each point in this shape space represents the shape of a configuration of points in some Euclidean space, irrespective of size, position, and orientation. In this shape space, scatters of points correspond to scatters of entire landmark configurations, see [144]. The importance of this space relies on the fact that most of the multivariate methods of geometric morphometrics are linearisations of statistical analyses of distances and directions in this space. It is relevant for this work since the analysis of shape variation for contour or contour partitions requires them to be registered to the common coordinate frame: Kendall's shape space.

A *shape signature* can be defined as any 1D function that represents 2D areas or boundaries [181]. Shape signatures are able to capture the perceptual feature of the shapes, and therefore are often used as a preprocessing to other feature

extraction algorithms, such as Fourier descriptors. [107]. Complex coordinates, Centroid distance function, Tangent angle (Turning angles), Curvature function, Area function, Triangle-area representation and Chord length function are the commonly used shape signatures. For a complete description and use of these refer to [120, 76, 182, 181, 107].

2.3 Shape Analysis and Morphometrics

2.3.1 Morphometry

Morphometry can be defined as the measurement of shape. Morphometry allows the transformation of visual information into its mathematical representation. For Bookstein [19] morphometrics is the study of biological shape and shape change through statistics.

Morphometrics is interested in the degree of relationship of the variables that explain the shape variation in a sample or in a population, and not on the identification of the objects as pattern recognition does.

Morphometric methods can be used to describe and compare shapes of whole organisms or particular structures of them. The principal characteristic of morphometry is the use of multivariate statistical methods over sets of variables. The variables are frequently measures taken from organisms. These measures usually correspond to lengths or widths of an object and to distances between reference points (Figure 2.3). Results are commonly expressed as graphs or numbers in terms of linear combinations of variables. PCA, canonical correlation analysis, discriminant functions or generalised distances are examples of the techniques used [131]. Eigenshape analysis is another technique that will be explained later with more detail.

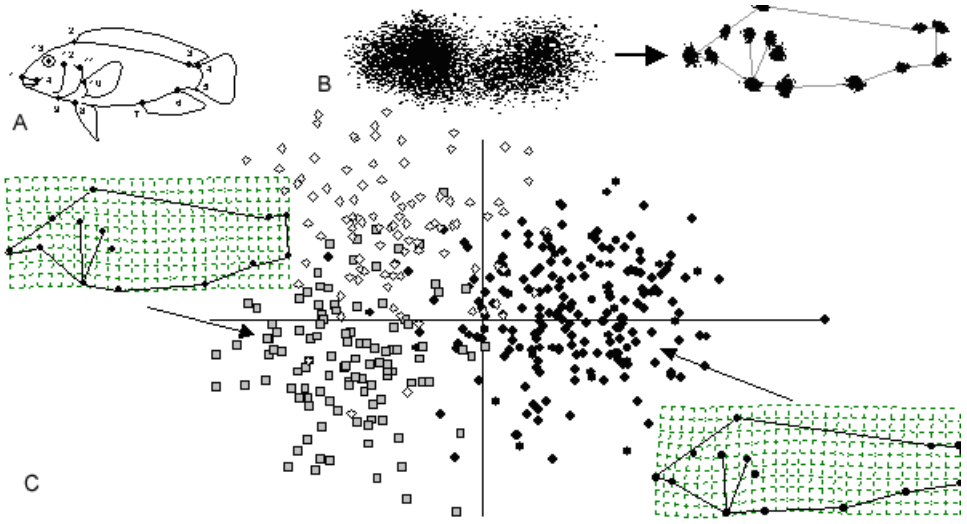


FIGURE 2.3: Graphical representation of the four-step morphometric protocol. **A** Landmarks recorded on the body of a child fish are quantified. **B** remove non-shape variation from landmarks of 412 specimens before and after Generalised Procrustes Analysis (GPA). **C** statistical analysis and graphical presentation of results. Deformation grids (left and right) are presented for two different species. Adapted from [2]

Morphometry can be regarded as an antecedent to the use of shape to analyse the variation and change in the form (size and shape) of organisms, hence many concepts are needed and used in Shape Analysis and Statistical Shape Analysis.

2.3.2 Shape Analysis

Measurement is a very important activity in many disciplines, not only to describe objects, but also to compare the shape of different objects. Therefore, the field of *Shape Analysis* implies the study of the relationship of measures with shape theory and its methods. The analysis of shape is related to the study of the geometrical properties of objects. The study of the shape of objects in mathematical terms, is necessary for the description and comparison of shapes. Hence, shape analysis is of great interest in a wide variety of fields of study and can find applications in a diverse variety of disciplines such as Biology, Medicine, Genetics, Archaeology, Geography, Geology and Agriculture [45].

The use of landmarks is fundamental to summarise and work with the objects. Shape Analysis involves methods for the geometrical study of objects where location, rotation and scale information can be removed [45], elements of rigid variation that are not of particular interest in Shape Analysis. In the same way, methods for quantifying the shape components and deriving information from it need to be considered.

The study of shape analysis has been achieved through different approaches. One of them is multivariate morphometrics [128] and the other major area of work is Allometry [139] (Figure 2.4).

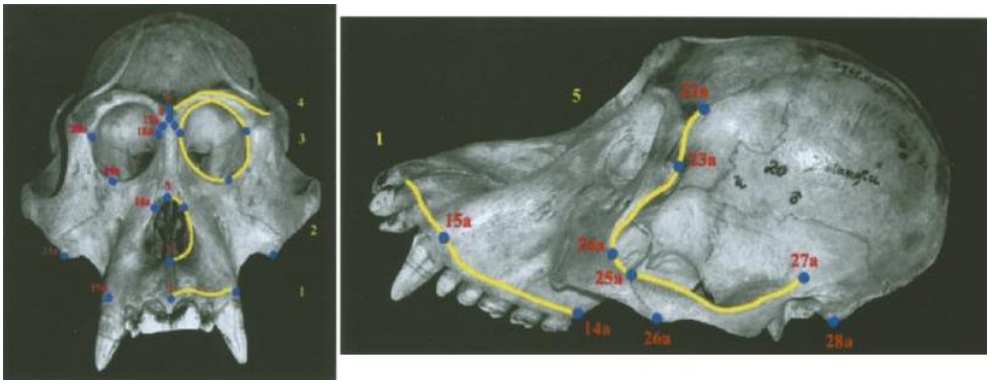


FIGURE 2.4: Examples of landmarks in an allometry study. Adapted from [139]

2.4 Statistical Shape Analysis

As was previously stated, shape analysis involves methods for studying the shapes of objects where location, rotation and scale information can be removed. This is important because the shape of an object is not changed when any of these transformations are applied. *Statistical Shape Analysis* (SSA) is the geometrical analysis from a set of shapes in which statistics describe the geometrical properties from similar shapes [45]. The field of SSA involves methods for quantifying the shape components of visual data and deriving information from them.

In the context of shape analysis, a *statistical model* is a term used to describe the mathematical construct that attempts to characterise how individuals within a

population vary [85]. The selection of the appropriate model is important because the adopted model influences the results. The choice of a suitable model depends on factors like the previous experience of the investigator, knowledge and familiarity with the data or mathematical convenience. All the available models can be regarded as approximations, so unfortunately, there is no correct model [85].

Methods in 2D shape analysis can typically be divided into three high level steps. The first step is to establish a geometric representation of the shapes. The second, involves the derivation of a set of features from the representation. These features must be invariant under the similarity transform. And finally in the third step, a statistical analysis method is chosen and applied to the features [179].

A *Statistical Shape Model* (SSM) is a statistical representation of a shape obtained from data. Statistical shape models study variation in shapes. SSMs have been applied to image segmentation especially in medical image applications, where expert knowledge is necessary [143].

The basic idea in statistical shape model building is to establish from a training set, the pattern of legal variation in the shapes and as well, the spatial relationships of structures for a given class of images. Statistical analysis is used to give a parameterisation of this variability, providing an appropriate representation of shape and allowing shape constraints to be applied [34].

The construction of a SSM consists of extracting the mean shape and deriving a series of modes of variation from a collection of training samples. The methods employed on this construction process strongly depend on the chosen shape representation. Here this work utilises a PDM, which will be explained with more detail in subsequent sections.

2.4.1 Procrustes Methods in the Statistical Analysis of Shape

Procrustes was a character from Greek mythology that claimed to have a magic all-fitting bed. The victims were fitted to the bed perfectly by either stretching their limbs or cutting them off [62]. Following the Procrustes story, there are three elements: the unfortunate traveller, say $\mathbf{K}_1$; the Procrustean bed $\mathbf{K}_2$; and the treatment to make the traveller fit in the bed, regarded as $\mathbf{T}$. The simplest algebraic statement of a Procrustes problem seeks a matrix $\mathbf{T}$ that minimises:

$$S = \| \mathbf{K}_1\mathbf{T} - \mathbf{K}_2 \| \tag{2.2}$$

This means that the minimisation problem 2.2 is called a Procrustes problem because $\mathbf{K}_1$ is transformed by $\mathbf{T}$ to fit the 'bed' $\mathbf{K}_2$. This can be regarded as a multivariate multiple regression problem, estimating $\mathbf{T}$ by:

$$\mathbf{T} = (\mathbf{K}_1'\mathbf{K}_1)^{-1}\mathbf{K}_1'\mathbf{K}_2. \tag{2.3}$$

In (2.2) $\mathbf{K}_1$ is matched to $\mathbf{K}_2$, the target matrix. Sometimes it is more appropriate to consider $\mathbf{K}_1$ and $\mathbf{K}_2$ symmetrically, so equation 2.2 transforms into

$$S = \| \mathbf{K}_1\mathbf{T}_1 - \mathbf{K}_2\mathbf{T}_2 \| . \tag{2.4}$$

This two-sided variant of (2.2) seeks to find transformation matrices $\mathbf{T}_1$ and $\mathbf{T}_2$ such that $\mathbf{K}_1\mathbf{T}_1$ best matches $\mathbf{K}_2\mathbf{T}_2$.

Then, according to the SSA, two geometrical figures or shapes $\mathbf{K}_1$ and $\mathbf{K}_2$ in $\mathbb{R}^k$, each consisting of n landmark points have the same shape if they differ by rotation, translation and scaling [60]. Procrustes analysis involves matching these configurations with transformations so they be as close as possible by Euclidean distance using least square techniques [45]. Procrustes methods are also useful in estimating an average shape to explore the structure of shape variability in a

data set. In order to compare such configurations in shape a measure of distance between the two configurations needs to be established.

Considering two centred configurations ζ_1 and ζ_2, both in $\mathbb{C}^k$, a suitable procedure to find such a distance, is to match them using similarity transformations. The differences between the fitted and the observed ζ_2 indicate the magnitude of the difference between ζ_1 and ζ_2. Considering the complex regression equation:

$$\zeta_2 = \xi \mathbf{T} e^{i\theta} + 1_k \beta + \varepsilon = \Upsilon \mathcal{R} + \varepsilon \tag{2.5}$$

where $\beta \in \mathbb{C}$ is the location, $\xi \in \mathbb{R}^+$ is the scale, θ is the angle of rotation, $\Upsilon = [1_k, \mathbf{T}]$ is the design matrix, $\mathcal{R} = [\beta, \xi e^{i\theta}]^{\mathbf{T}} \in \mathbb{C}^2$ are the regression parameters and ε is a complex k-vector of errors. The most straightforward approach to estimate the regression parameter β is to perform a least squares approach, this means to minimise $\varepsilon^* \varepsilon$ where ε^* is the complex conjugate of the transpose of ε. The solution is given by [44]:

$$\hat{\mathcal{R}} = (\Upsilon^* \Upsilon)^{-1} \Upsilon^* \zeta_2 \tag{2.6}$$

A measure of residual discrepancy between ζ_1 and ζ_2 is:

$$D_G(\zeta_2, \zeta_1) = \|\zeta_2 - \zeta_1 \hat{\mathcal{R}}\| = \|(I_k - \Upsilon(\Upsilon^* \Upsilon)^{-1} \Upsilon^*)\zeta_2\|, \tag{2.7}$$

and $D^2_{\mathcal{G}}(\zeta_2, \zeta_1)$ is known as the Procrustes sum of squares [60]. So, as is well known, the full and partial Procrustes distances in Kendall's shape space for two dimensional points can be obtained as the solution to complex linear regression problems.

But, Procrustes analysis is limited in its application since it is the comparison of only two sets of configurations (shapes). Therefore, there is the need to use a method of evaluation for k sets of configurations. With Generalised orthogonal Procrustes analysis (GPA) [61] k sets can be aligned to one target shape or aligned to each other.

With k sets, the natural form of the norm to be minimised is:

$$\sum_{i<j}^{k} \parallel \mathbf{K}_i\mathbf{T}_i - \mathbf{K}_j\mathbf{T}_j \parallel . \tag{2.8}$$

Having the following identity is useful:

$$\sum_{i<j}^{k} \parallel \mathbf{K}_i\mathbf{T}_i - \mathbf{K}_j\mathbf{T}_j \parallel = k \sum_{m=1}^{k} \parallel \mathbf{K}_m\mathbf{T}_m - \mathcal{G} \parallel . \tag{2.9}$$

where

$$\mathcal{G} = k^{-1} \sum_{m=1}^{k} (\mathbf{K}_m\mathbf{T}_m) \tag{2.10}$$

The matrix $\mathcal{G}$ is known as the group average configuration of the initial matrices $\mathbf{K}_1, \mathbf{K}_2, \ldots, \mathbf{K}_k$ after being transformed by $\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_k$. Its relevance relies on the fact that it provides a basis for algorithms and shows that the sum of the pair wise Procrustes solution may be expressed in terms of deviations from a single matrix, in this case $\mathcal{G}$.

An algorithm for generalised orthogonal Procrustes analysis can be [132]:

1. Select one shape to be the approximate mean shape, i.e. the first shape in the set.
2. Align the shapes to the approximate mean shape.
   a) Calculate the centroid of each shape.
   b) Align all shapes centroid to the origin.
   c) Normalise each shape's centroid.
   d) Rotate each shape to align with the newest approximate mean.
3. Calculate the new approximate mean from the aligned shapes.
4. If the approximate mean from steps 2 and 3 are different then return to step 2, otherwise the true mean shape of the set has been found.

For more details about GPA and its algorithms refer to [45, 61, 62].

2.4.2 Principal Components Analysis

Principal Components Analysis is a multivariate statistical technique that provides means to identify patterns in data, and to highlight similarities and differences in them, see [146]. Moreover, as noted by Jolliffe [74] the idea of PCA is to simplify a data set of interrelated variables, by reducing the number of dimensions without meaningful loss of information about the variation in the data. Another outstanding feature of the technique is that it has no underlying statistical model of the observed variables so it concentrates on the explanation of the total variation in the variables based on the maximum variance properties of principal components [46].

PCA is a method to transform the original variables into uncorrelated ones. The new variables are called principal components, and each of these is a linear combination of the original variables. Variance is a measure of the amount of information reflected by each principal component [3], but since the aim of statistical analysis is usually to analyse how much dimensions vary from the mean with respect to each other, covariance is used [146].

Let $\mathbf{C}$ be a covariance matrix. The *jth* principal component of the sample of k-variate observations is the linear compound [116]:

$$\mathbf{y}_j = \psi_{j1}\mathbf{x}_1 + \cdots + \psi_{jp}\mathbf{x}_k = \sum_{i=1}^{k} \psi_{ji}\mathbf{x}_i \tag{2.11}$$

whose coefficients are the elements of the characteristic vector of the sample covariance matrix $\mathbf{C}$ corresponding to the *jth* largest characteristic root. PCA finds the optimal weight vector $\psi_j = (\psi_{j1}\mathbf{x}_1 + \cdots + \psi_{jk}\mathbf{x}_k)$ and the associated variance of $\mathbf{y}_j$ usually denoted by λ_j, therefore these are the main statistical results from a PCA analysis [46]. These lead to the matrix equation:

$$\mathbf{C}\psi = \lambda\psi \tag{2.12}$$

where λ is the *eigenvalue* of the covariance matrix $\mathbf{C}$ and ψ is its associated *eigenvector* [87]. Hence, there are j linear transformations or principal components of the original j variables and equation 2.11 can be expressed as:

$$\mathbf{y} = \Psi^T \mathbf{x} \tag{2.13}$$

where $\mathbf{y}$ is a j element vector of principal component scores, Ψ is a $j \times j$ matrix of eigenvectors:

$$\Psi = (\psi_1 \ \psi_2 \ \ldots \ \psi_t) \tag{2.14}$$

where the *ith* row corresponds to the elements of the eigenvector associated with the *ith* eigenvalue and $\mathbf{x}$ is a j element column vector of the original variables [46].

PCA implies a linear transformation that transforms the data to a new coordinate system. PCA is equivalent to Karhunen-Loève and Singular Value Decomposition transforms used in the context of multivariate statistical analysis [57]. It is also known as the Hotelling transform [6].

The steps to perform PCA are as follows. The data is arranged in an m by n matrix, with a column for each dimension. Then the mean is calculated for each dimension and subtracted. The next step is to calculate the covariance matrix, then the eigenvectors and eigenvalues are obtained performing an SVD transform.

Formally singular value decomposition (SVD) can be defined as [151]: Any m by n matrix $\mathbf{K}$ can be factored into

$$\mathbf{K} = \mathbf{Q}_1 \Sigma \mathbf{Q}_2^T = (orthogonal)(diagonal)(orthogonal) \tag{2.15}$$

The columns of $\mathbf{Q}_1$ (m by m) are eigenvectors of $\mathbf{K}\ \mathbf{K}^T$, and the columns of $\mathbf{Q}_2$ (n by n) are eigenvectors of $\mathbf{K}^T\ \mathbf{K}$. The n singular values on the diagonal of Σ (m by n) are the square roots of the nonzero eigenvalues of both $\mathbf{K}\ \mathbf{K}^T$ and $\mathbf{K}^T\ \mathbf{K}$.

The number of principal components retained must be determined by the rule that asserts that the number should be enough to explain a certain percentage of the total variance. A common cutoff point is 80%, but some researchers have different

ways to select the number of components [3]. There are other strategies to select the components, for instance, the Kaiser criterion, Scree plot, Parallel Analysis, Minimum average partial criterion, Variance explained criteria, Joliffe criterion, Mean eigenvalue, or Comprehensibility, among others. More details are given in [55] and [46].

2.4.3 Eigenshape Analysis

Eigenshape analysis is a PCA technique developed by Lohmann in 1983. Its aim is to represent a collection of shapes by the fewest number of orthogonal functions necessary to represent the greatest proportion of the shape's variance [89].

Eigenshape analysis derives a set of empirical orthogonal shape functions by an eigenfunction or PCA of a matrix of correlations between shapes, so it is this derivation that is called eigenshape analysis. In his paper [89], Lohmann describes a procedure that in general is as follows:

1. Acquire shape's outline data. This refers to the x, y Cartesian coordinates of the outline from the specimen. In [95], MacLeod quotes that the number of points to be digitised to represent any curve depends on the complexity of the curve. Since the specimen's orientation is arbitrary, rotation does not affect the shape's outline in the x, y plane. But on the other hand, rotation on the z direction can affect the silhouette of the specimen.

2. Represent the shape as the shape function $\phi^*(l)$, the normalised net angular change in direction ϕ at each step around the perimeter (l) of the shape. $\phi^*(l)$ is an exact description of how a shape differs from a circle, and it retains the necessary information (except size) to reconstruct precisely the outline (Figure 2.5). In other words, the resulting function is like a road map around the shape and represents mathematically the precise way to say "take one step, turn 20 to the right, take another step, turn 10 to the left,..." [95].

3. Homoligise shapes. There are cases where no homologous points can be determined, so in those, shapes are defined to match homologously once their shape function are mutually rotated to positions of maximum correlation. In practice this is equivalent to selecting a single representative specimen as a reference shape, and locate the other shapes rotated with respect to it. An interpolation of the perimeter to the same number of equal-length segments, between the data digitisation and the computation of $\phi^*(l)$, must be performed. This is to ensure the possibility of a point to point comparison between shapes in cases without identifiable landmarks.

4. Compute the eigenshapes. Through PCA, eigenvalues are obtained, and these reflect the proportion of observed shape variation that each eigenshape represents. The $\phi^*(l)$ shape functions represent the observed shapes, matched point by point. The variation observed among them can be represented by linear combinations of a few of these uncorrelated shape functions. The fewest is the set of empirical orthogonal shape functions derived by an eigenfunction analysis of correlations between the shapes. These eigenshape functions represent the fewest shapes needed to represent most of the original, observed shape variation.

One of the main advantages over other morphometric procedures is that the original shape, no matter how complicated, can be reconstructed from its eigenshape representation. The latter is only as complex as necessary to describe the observed differences between the shapes [89]. Another important aspect of the eigenshape method is that it allows a graphical representation of the results of any analysis. This is important because, according to MacLeod [95], this is powerful and underutilised aspect, since it can provide a qualitative way to communicate research results in an understandable way to morphologists.

Comparisons between eigenshape analysis and other morphometric studies can be found in the works by Lohmann [89], MacLeod [95] and [96], Macleod and Rose [97], Cadrin [24], and Ubukata [164].

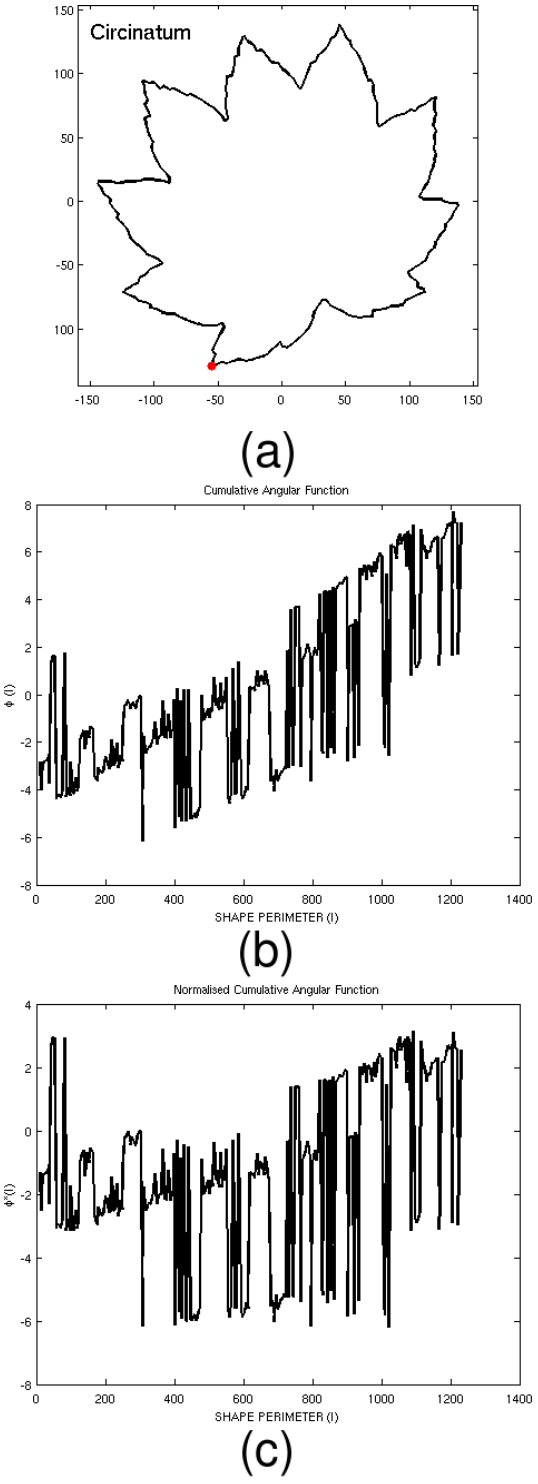


FIGURE 2.5: Graphical representation of the shape of a Circinatum leaf in x,y Cartesian coordinates (a), the shape function $\phi(l)$ (b) and $\phi^*(l)$ (c).

2.4.4 Point Distribution Models

A *Point Distribution Model* (or PDM) is a way to represent shapes through a model that reflects the position of labelled points [32]. The model also represents the mean geometry of a shape and derives statistical modes of geometric variation inferred from a training set of shapes. It has become a standard in computer vision for statistical shape analysis and especially for segmentation purposes on medical images. Therefore this had led to the creation of two important techniques: Active Shape models (ASM) [31] and Active Appearance Models (AAM) [29]. A comparison between both models can be found in [28].

PDMs rely on landmark points, defined in section 2.2 of this chapter. These points represent the boundary of an object. The method can be generalised as follows:

(a) Obtain a training set of outlines with enough landmarks so they are able to sufficiently approximate the geometry of the original shapes (Figure 2.6).

(b) Align the sets of landmarks using the Generalised Procrustes Analysis. The idea behind this is that the shape information is not related to affine pose parameters, so they need to be removed. Having this, a mean shape can now be computed by averaging the aligned landmark positions (Figure 2.7).

(c) PCA is a relevant tool to study the correlations of shape between groups of landmarks among the training set population. PCA computes the eigenvectors and eigenvalues of the training set using a covariance matrix. Each eigenvector describes a principal mode of variation along the set, the corresponding eigenvalue indicating the importance of this mode in the shape space scattering (Figure 2.8). If correlation is found between landmarks, then the total variation of the space is concentrated on the first few eigenvectors, that present a very rapid decrease in their corresponding eigenvalues. Otherwise, if no correlation is found, that might suggest that the training set has no variation or that the pose of the landmarks has not been properly removed.

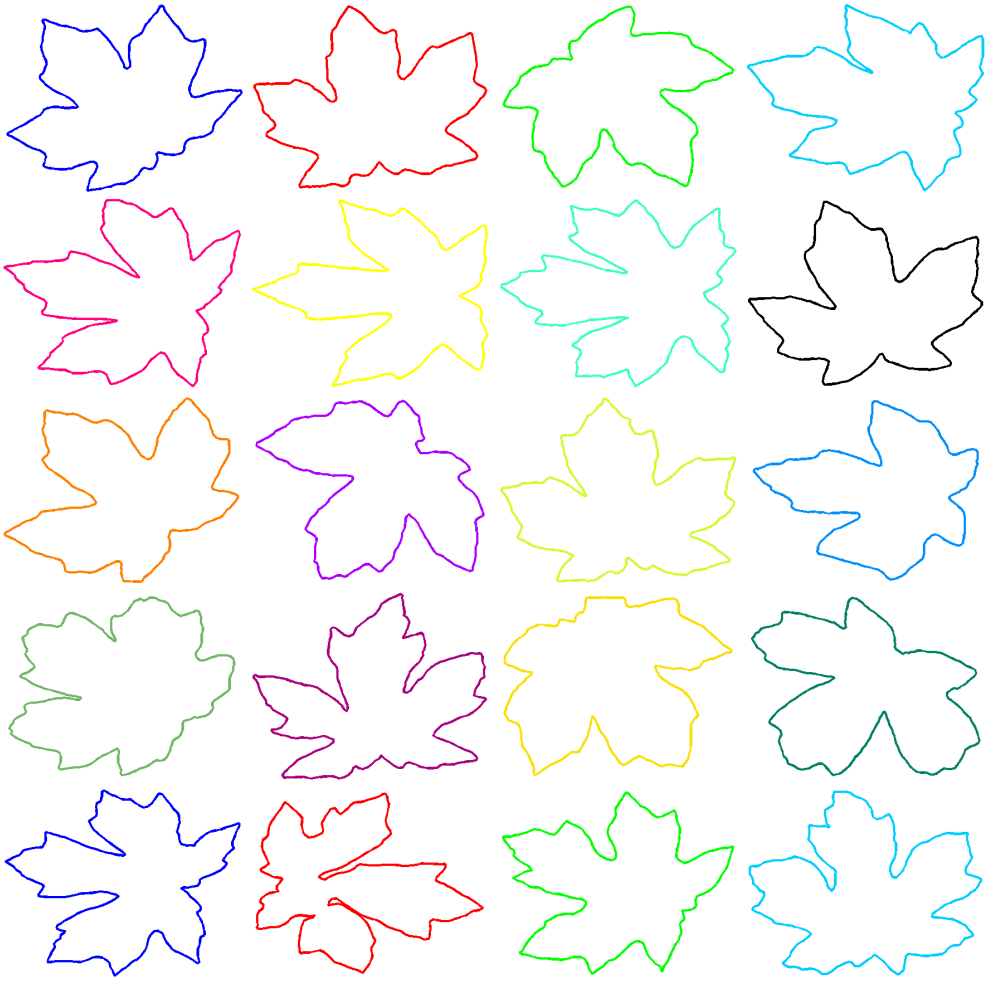


FIGURE 2.6: Training set of 20 leaves of the type Macrophyllum.

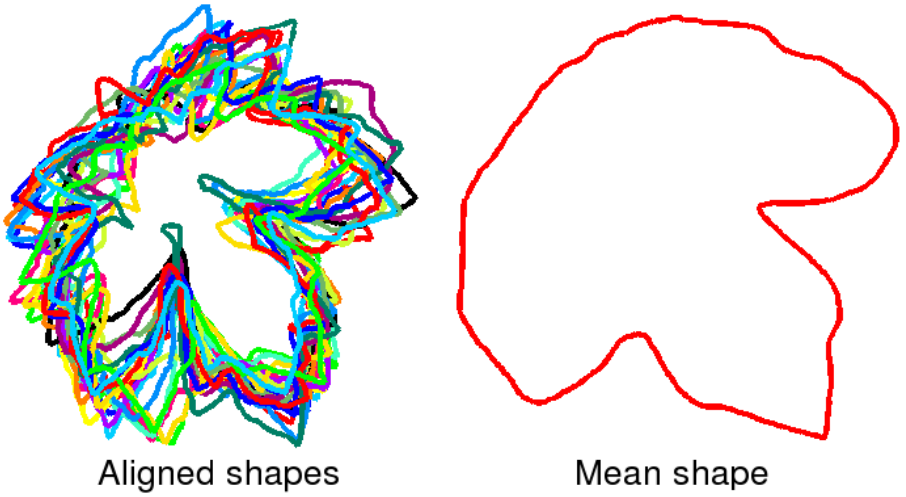


FIGURE 2.7: Shape alignment and Mean shape.

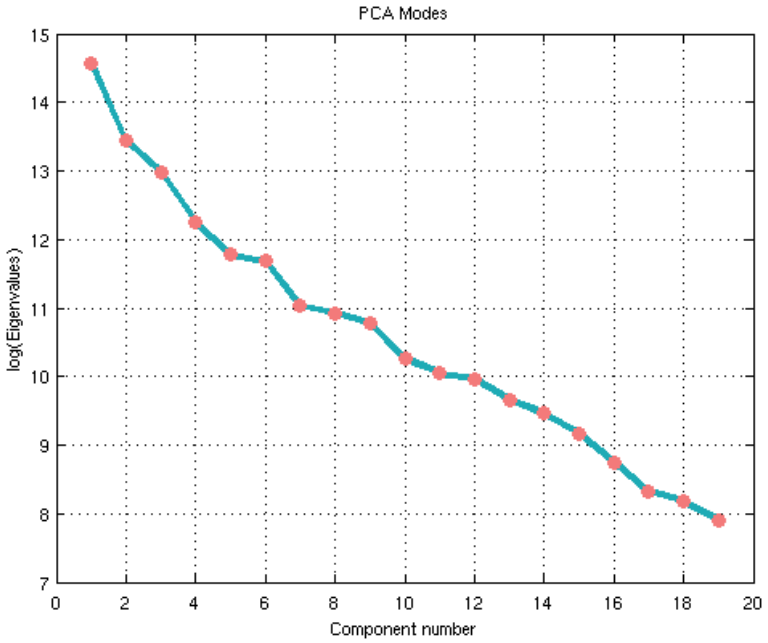


FIGURE 2.8: PCA modes log plot.

(d) Using the set of generated eigenvectors and eigenvalues, any shape of the training set can be approximated using the mean shape and a weighted sum of deviations obtained from the modes (Figure 2.9).

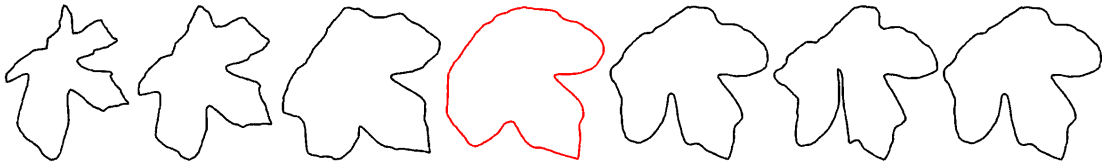


FIGURE 2.9: The figure in red is the mean shape, and here the effects of varying the first parameter of the leaves model ψ_1 in the interval $-3\sqrt{\lambda_1}$ (left) to $+3\sqrt{\lambda_1}$ (right) are shown.

The generated eigenvectors, can be seen as a sequence of vectors associated to corresponding landmarks, where each comprise a mixture of shape variation for the whole shape. Here, the model consists of the mean positions of these points and the main modes of variation, which describe how the points tend to move from the mean.

The most important idea behind the PDMs is that eigenvectors can be linearly combined to create new shape instances that will be similar to any in the training set [32].

The remaining subsections corresponding to the steps (a) to (d) of the aforementioned PDM, provide a description of how the steps of the proposed model perform the tasks of contour tracing, how the shapes are aligned by Registration, how the Statistical Variation of an aligned set of shapes is captured and finally, how the shapes can be reconstructed.

2.4.4.1 Contour Tracing

The input data of our model are plane coordinates that are extracted from images (see figure 2.11). The process to extract the boundary information or the coordinates of the boundary from the images can be summarised in the following figure 2.10:

The assumption here is that the input images contain the shapes of interest on plain backgrounds. The first step is to binarise the image, hence a thresholding can be applied to convert the image or alternatively can be saved in such format. Often images are corrupted with noise, therefore a denoising process is applied so that isolated pixels can be eliminated. Then, the shape is traced and the boundary coordinates are obtained. In practice, preprocessing is done using standard Matlab functions from the Image Processing Data Toolbox. Different functions were used, for instance, *im2bw* to convert the image to a binary one and filter the noise, then to find the remaining shape the function *find* was used to get the indices and values of nonzero elements in the image. And finally, *bwboundaries* was used to trace the region boundaries in the filtered binary image and obtain the contour coordinates. The final step is to resample the resulting contour to a desired number of coordinates using cubic spline data interpolation functions. The created contours are always closed.

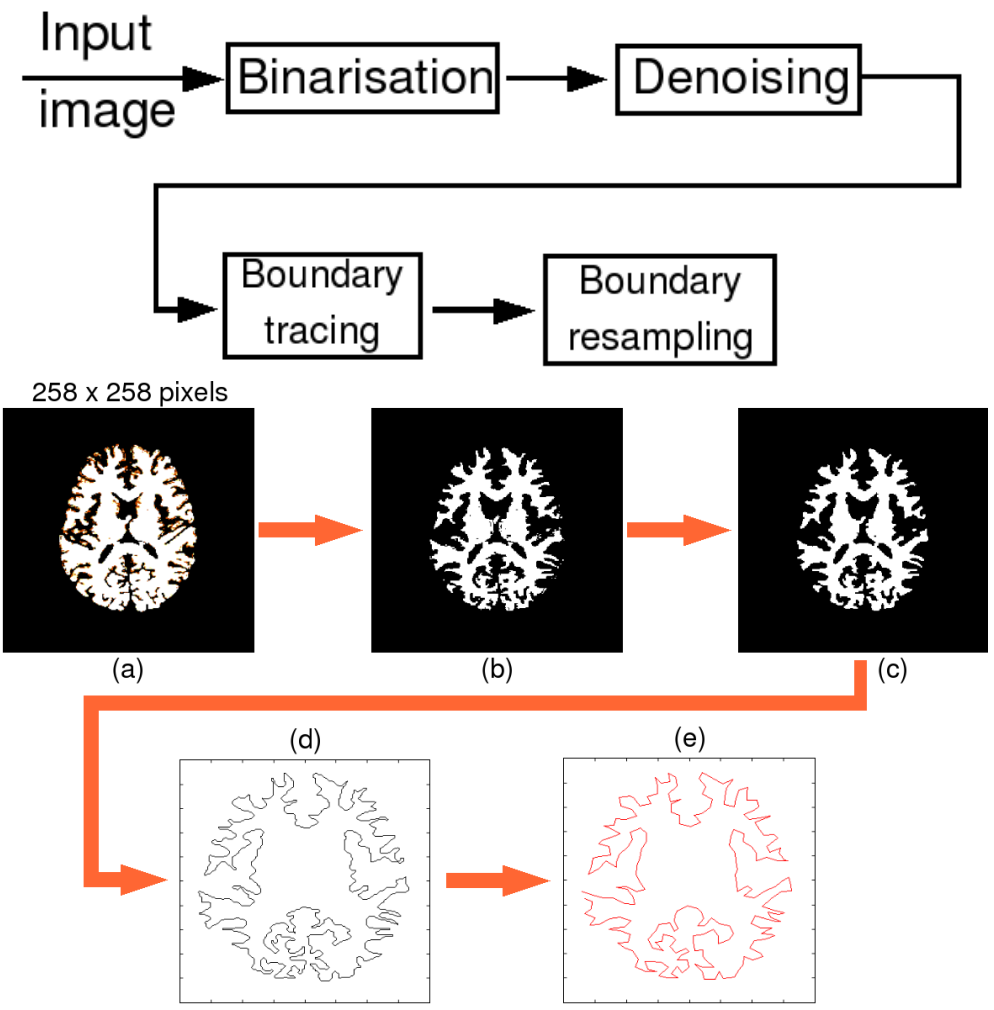


FIGURE 2.10: Processing to extract the boundary coordinates of an image: **(a)** Original image, **(b)** binarisation of the image, **(c)** denoising process, **(d)** boundary coordinates and **(e)** resampled contour.

Any *Contour* can be defined as a set of n points in the plane, $\mathcal{C}_i = (x_i, y_i)$ with $1 \leq i \leq n$, denoted by:

$$\mathcal{C} = (x_1, y_1, x_2, y_2, \ldots, x_n, y_n)^T \tag{2.16}$$

2.4.4.2 Registration Analysis

To describe geometrically any object it is possible to decompose it into registration and shape information. For example, the location, rotation and size of a shape can be the registration information and the geometrical information that remains can

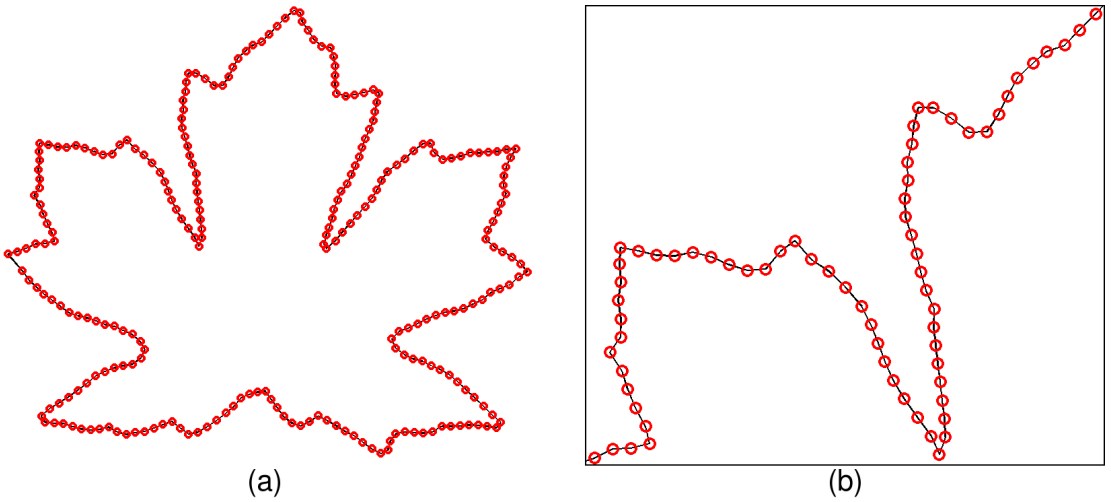


FIGURE 2.11: 256 points contour from the boundary of a leaf of the type Macrophyllum (a) and zoom into an upper section (b).

be regarded as the shape of the object. Then, the shape of an object is invariant under registration transformations and two objects have the same shape if they can be registered to match exactly [44]. The implications of this are fundamental, because if the variability in shape in a population of objects needs to be estimated, then the registration information is not relevant and can be neglected. Hence, by registration, the objects to be analysed are corrected for position, rotation and scale variation.

Having sets of clustered partitions, an important step in the proposed local shape model is **registration analysis** or **pose removal**. Given a set of shapes $\Phi = \mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_n$, the registration analysis consists of estimating an affine warp by transforming any shape $\mathcal{S}_j$ to some prototype shape $\mathcal{S}_i$. The warps $\mathcal{S}_j \rightarrow \mathcal{S}_i$ are searched by determining rigid body transformation parameters for each $\mathcal{S}_j$, and the parameters are kept to reconstruct the shapes accordingly. It is important to mention here that the shapes are just corrected in rotation and translation since the data used is considered to have the same scale, so the general formulation of this problem can be written as:

$$\mathcal{S}_i = \mathbf{R}\mathcal{S}_j + \nu \tag{2.17}$$

Where $\mathbf{R}$ is the rotation matrix and ν is a translation vector. Later, the method

to obtain these is explained, but first Procrustes analysis is discussed. The basic idea in Procrustean analysis is to find the required similarity transformations through objective function minimisation. This objective function can be defined by choosing an appropriate shape distance measurement and an appropriate reference shape, with respect to which the distance measure is evaluated.

The process to determine the rigid body transformations parameters (rotation and translation) that describe the transformation of points from one reference frame to another is now explained. To perform this, one of the segments should be selected as the reference one, and the others will be transformed in reference to this one. This relationship can be expressed as follows:

$$y_i = \mathbf{R}x_i + \nu \tag{2.18}$$

where y_i is the *ith* point measured in the reference frame $\mathcal{S}_k$, $\mathbf{R}$ is the rotation matrix, x_i is the *ith* point measured in the reference frame $\mathcal{S}_j$ and ν is a translation vector of the origin of reference frame $\mathcal{S}_j$ measured in reference frame $\mathcal{S}_k$.

To determine both the rotation matrix $\mathbf{R}$ and the translation vector ν, least squares is used to minimise:

$$\frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{R}x_i + \nu - y_i\right)^T\left(\mathbf{R}x_i + \nu - y_i\right) \tag{2.19}$$

Then the mean vectors are computed ($\overline{x}$ and $\overline{y}$) so the vector ν can be determined using them as:

$$\nu = \overline{y} - \mathbf{R}\overline{x}. \tag{2.20}$$

Next, the cross dispersion matrix $\mathbf{C}$ is computed from:

$$\mathbf{C} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \overline{y}\right)\left(x_i - \overline{x}\right)^T. \tag{2.21}$$

Then the SVD of $\mathbf{C}$ is computed by:

$$\mathbf{C} = \mathbf{U}\Lambda\mathbf{V}^T \tag{2.22}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices with the eigenvectors, and Λ contains the eigenvalues of $\mathbf{C}$. Now using $\mathbf{U}$ and $\mathbf{V}$, it is possible to calculate $\mathbf{R}$ as:

$$\mathbf{R} = \mathbf{U}\mathbf{V}^T \tag{2.23}$$

With $\mathbf{R}$ and ν we use (2.18) to complete the registration step. The method used in this work to determine such parameters is described in more detail in [25].

2.4.4.3 Capturing the Statistical Variation of a Set of Aligned Shapes

The following is an account of how the statistical variation of a set of aligned shapes is captured using the PCA. Under correspondence free conditions, meaning with this after all the shapes are registered, our set is ready for us to capture the statistics of the set of aligned shapes [32].

Let $\Phi = \mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_n$ be a set of partitions or shapes, and $n_{\mathcal{S}}$ the cardinality of the set. Then for each shape of the form 2.16, $\mathcal{S}_i, i = 1, \ldots, n_{\mathcal{S}}$ we obtain the mean:

$$\overline{\mathcal{S}_i} = \frac{1}{n_{\mathcal{S}}} \sum_{i=1}^{n_{\mathcal{S}}} \mathcal{S}_i \tag{2.24}$$

The modes of variation, or the ways in which the points of the shape tend to move together, can be found applying PCA to the deviations from the mean as follows.

First, for each shape in the training set the deviation from the mean is calculated as:

$$d\mathcal{S}_i = \mathcal{S}_i - \overline{\mathcal{S}} \tag{2.25}$$

Now, the $2n \times 2n$ covariance matrix $\mathbf{C}$ is calculated using:

$$\mathbf{C} = \frac{1}{n_{\mathcal{S}}} \sum_{i=1}^{n_{\mathcal{S}}} d\mathcal{S}_i d\mathcal{S}_i^T \tag{2.26}$$

The modes of variation of the points of each shape are described by the unit eigenvectors of $\mathbf{C}$, ψ_i $(i = 1, \ldots, 2n)$ (see equation 2.12) such that:

$$\mathbf{C}\psi_i = \lambda_i \psi_i \tag{2.27}$$

where λ_i is the i'th eigenvalue of $\mathbf{C}$, with $\lambda_i \geq \lambda_{i+1}$.

In practice, to derive the eigenvectors and their corresponding eigenvalues, an SVD (equation 2.15) is performed over the covariance matrix $\mathbf{C}$, which is an already implemented function in MATLAB. In general, PCA derives modes that influence all variables simultaneously, meaning that varying one mode will affect all landmarks of the shape model.

It can be demonstrated that the eigenvectors that correspond to the largest eigenvalues of a covariance matrix describe the most significant modes of variation [54]. Additionally, the modes of variation can be used in an alternative way to evaluate its importance. In this case, the sizes of the principal non-zero eigenvalues are plotted against the number of principal mode (see figure 2.8). So it is possible to use this plot to compare the performance of the methods based on eigenshape analysis in the following way. Having the eigenvalues plots for different data sets or methods, it is likely to observe how compact is the shape space, meaning with this that the most compact method will be reflected on the plot that gives the least variation from the mean [15].

2.4.4.4 Shape reconstruction

Finally, an important feature of this analysis and of this work is the way the reconstructions are presented. Any shape of the set can be approximated using

the mean shape and a sum of the modes: $\tilde{\mathcal{S}} = \overline{\mathcal{S}} + \Psi$, where Ψ is the matrix of the first k eigenvectors as equation 2.14.

Here an alternative way to visualise results has been created, and it consists of presenting the reconstructed partitions *blended back* into the smoothed version of the contour to which they belong (figure 2.12). The blending is achieved by performing a window function using a squared cosine. Suppose that we have 2 partitions to blend, namely $\mathcal{P}_i$ and $\mathcal{P}_j$ of the same size m, then we use a window $\omega(k)$ of size m as well such that each new blended partition, $\widehat{\mathcal{P}}_n$, is given by

$$\widehat{\mathcal{P}}_n = \omega(k)\mathcal{P}_i + \omega(1-k)\mathcal{P}_j \tag{2.28}$$

with

$$\omega(k) = cos^2(k), -\pi/2 \leq k \leq \pi/2$$

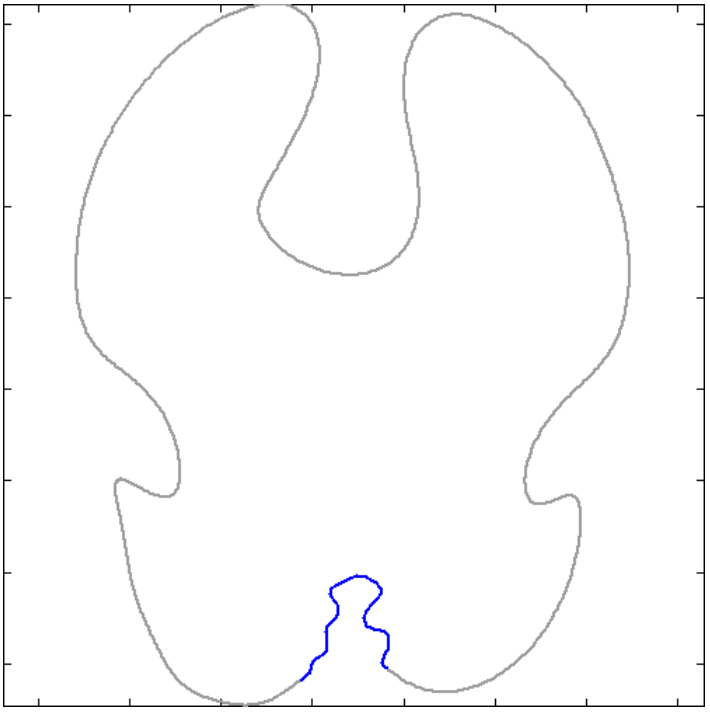


FIGURE 2.12: Example of a reconstructed partition blended back into the smoothed version of the contour it belongs to.

Later, in Chapter 4 more examples of this reconstructions will be given.

2.5 Summary

This chapter has described in overview the background useful in order to understand the proposed method and it has focused mainly on the concepts related to shape and its analysis.

Shape is an important feature of the human visual system (HVS) as it is one of the basic features used to describe perceived objects, both natural and man-made. Dryden and Mardia discuss shape [45] as the geometrical information that stays when location, scale and rotational effects are filtered out from a particular object. Small complements this idea by further adding that the remaining information should be invariant to transformations [145]. Another important concept is *shape descriptor* that, in general, can be regarded as a set of numbers produced to represent a given shape feature. Such descriptors attempt to quantify the shape in ways that agree with human concepts of shape or with task-specific requirements [107]. Shapes can be described by a series of finite points locating points on the sample to be analysed, called *landmarks*. *Shape space* can be considered as the collection of all possible shapes [77]. The fundamental space for this work is Kendall's shape space, since it provides a complete geometric setting useful for the analysis of Procrustes distances among sets of landmarks [44]. Another impotant concept is *shape signature*, that can be defined as any 1D function that represents 2D areas or boundaries [181]. Shape signatures are able to capture the perceptual feature of the shapes, and therefore are often used as a preprocessing to other feature extraction algorithms, such as Fourier descriptors.

Morphometry can be defined as the measurement of shape. Morphometry allows the transformation of visual information into its mathematical representation. As previously stated, Morphometry and Shape analysis but, since Morphometry can be regarded as an antecedent to Shape Analysis, the difference relies on the fact that Shape Analysis is related to the study of the geometrical properties of objects. *Statistical Shape Analysis* (SSA) is a geometrical analysis from a set of shapes in which statistics describe the geometrical properties from similar shapes [45]. Two important aspects of SSA are introduced as well, Procrustes methods

and PCA. Procrustes analysis is a form of statistical shape analysis used to analyse the distribution of a set of shapes. The name Procrustes refers to a bandit from Greek mythology who made his victims fit his bed either by stretching their limbs or cutting them off. *Principal Components Analysis* is a multivariate statistical technique that provides means to identify patterns in data, and to highlight similarities and differences in them [146]. The next section of the chapter was dedicated to *Point Distribution Models* that are models for representing the mean geometry of a shape and some statistical modes of geometric variation inferred from a training set of shapes. In this section as well, descriptions of how the contours are extracted from images, how registration analysis is carried out, how the statistical shape variation is captured, and how the shapes are reconstructed, was given. To obtain the shape contours, first the image they come from is binarised and denoised, then the shape is traced and the boundary coordinates are obtained. In practice, preprocessing is done using standard Matlab functions from the Image Processing Data Toolbox. Given a set of shapes, the registration analysis consists of estimating an affine warp by transforming any shape to some prototype shape. The warps are searched by determining rigid body transformation parameters for each shape from the set, and the parameters are kept to reconstruct the shapes accordingly. Under correspondence free conditions, after all the shapes are registered, the set is ready to capture the statistics of the set [32]. The modes of variation, or the ways in which the points of the shape tend to move together, can be found applying PCA to the deviations from the mean. Finally, any shape of the set can be approximated using the mean shape and a sum of the modes.

The next two chapters are devoted to the study of the best way to find partitions out of contours. Chapter three illustrates the first attempt to generate partitions through the use of Fractal Dimension in a Bayesian framework (MCMC). Chapter four presents a more stable solution to the problems presented by the Markov Chain Monte Carlo algorithm and shows an alternative solution using the Curvature Scale Space method.

Chapter 3

Fractal Analysis and Markov Chain Monte Carlo Simulation for Contour Localisation

3.1 Introduction

Being able to partition a contour into a set of meaningful local parts has many applications in machine vision, image retrieval, terrain classification or handwriting recognition. As presented in Chapter 1, several methods for contour partitioning have been developed but none of them take into account the fractal nature of the objects such as brain contours, since none of them consider types of these curves. This section presents a method of modelling fractal curves, such as the boundary of brain white matter, and partitioning them into segments having equal fractal dimension. This might lead to a better set of partitions for SSA, one of the aims of this work (refer to section 1.4). Since the solution space, for a given number of contour points and a required set of partitions is very large, a Bayesian framework of reversible-jump Markov Chain Monte Carlo (MCMC) is used together with a sampler based on the Metropolis-Hastings test. Details of the algorithm are presented as well as the theoretical concepts behind it. Results on simple contours

(animal silhouettes) and space-filling brain contours are shown, along with the convergence characteristics of the method. Limitations and the contributions of the method are discussed at the end of the chapter.

3.2 Fractals, Space Filling Curves and Fractal analysis

3.2.1 Fractals

A *fractal* can be defined as an object or quantity that unfolds self-similarity on all scales [170]. Although the object does not necessarily exhibit exactly the same structure at all scales, the same 'type' of structures can be viewed on all scales. (Figure 3.1).

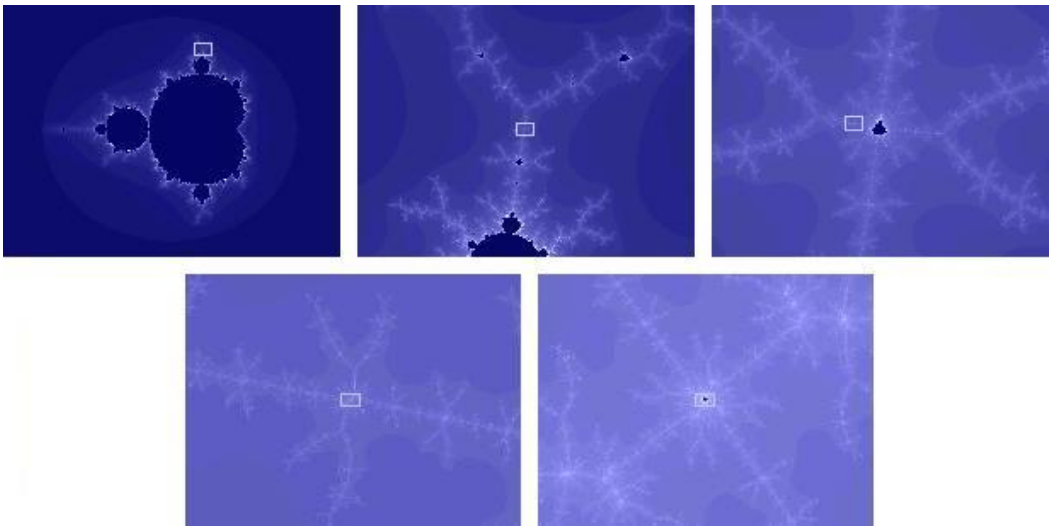


FIGURE 3.1: Central set surrounded by mini-Mandelbrot sets. In each picture a small rectangle indicates the limits of the following picture (Adapted from [92])

It is easy to find approximate fractals in nature, since these objects present self-similar structure (Figure 3.2). Therefore to a certain extent, fractals can be considered to be descriptors of the geometry of nature. Nevertheless, it is important

to emphasise that fractals should not be considered as the only approximation to nature's geometry [98] [99] [11].

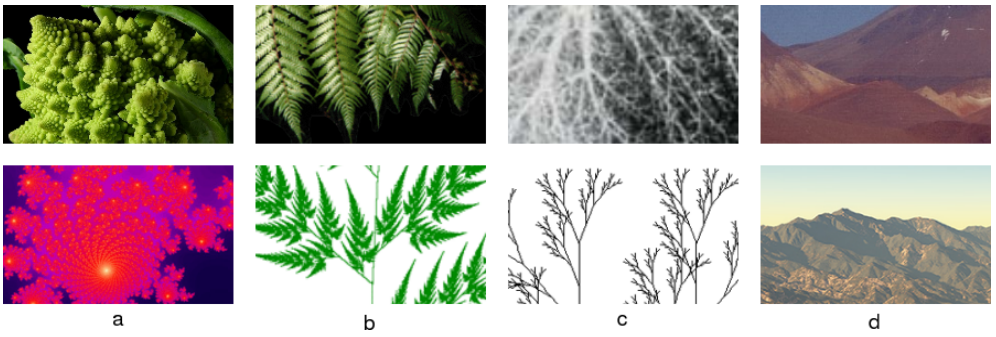


FIGURE 3.2: (a) Broccoli and Mandelbrot set, (b) Fern and Iterated Function Systems (IFS) fern, (c) Blood vessels (lung) and IFS branch, and (d) Mount Llullaillaco and Death Valley computer generated simulation. (Adapted from [155], [20] and [59]).

3.2.2 Space Filling Curves

Space filling curves are a type of fractal, and intuitively they can be regarded as a continuous path that visits each point exactly once and never crossing itself, so they fill the space without leaving 'holes' (Figures 3.3 and 3.4).

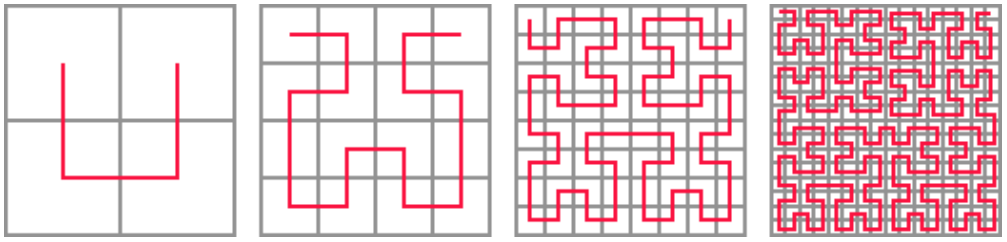


FIGURE 3.3: Hilbert Space Filling Curve

More precisely, a curve $f_\star(\mathcal{I})$, exists if there is a continuous function mapping the interval $\mathcal{I} = [0, 1]$ to an n-dimensional Euclidean space $\mathbb{E}^n$ such that $f : \mathcal{I} \rightarrow \mathbb{E}^n$. Then the curves that pass through every point of an n-dimensional region with positive Jordan content (area for $n = 2$, volume for $n = 3$) such as the square $\mathcal{Q}$ in $\mathbb{E}^2$ and the cube $\mathcal{W}$ in $\mathbb{E}^3$ are called *space-filling*.

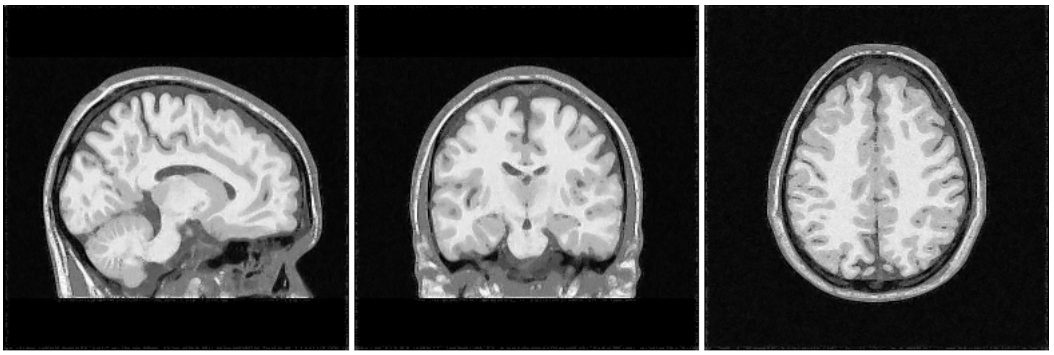


FIGURE 3.4: Cerebrospinal fluid (CSF) images of the brain [10]. From left to right: sagittal, coronal and axial views. The cerebral cortex is the **folded grey tissue** that covers the surface of each cerebral hemisphere. In Chapter 1 it was discussed that an analysis of the geometry of the human cerebral cortex was performed in [79] and it was shown that the human cerebral cortex does possess self-similarity that resembles a fractal structure.

Mathematically, if $f : \mathcal{I} \rightarrow \mathbb{E}^n$, with $n \geq 2$, is continuous and $J_n\left(f_\star\left(\mathcal{I}\right)\right) > 0$, then $f_\star\left(\mathcal{I}\right)$ is called a space filling curve. Where J_n represents the *Jordan content* (area, volume) of a *Jordan measurable* subset of $\mathbb{E}^n$ [136].

3.2.3 Fractal Analysis

The fractal geometry of shapes can be a useful tool to describe and understand biological systems with fractal properties. In general, fractal analysis can quantify the irregularity of a fractal object, with the fractal dimension acting as an index or descriptor for the shape complexity. This method has been successfully applied to different areas of brain research, such as cell morphometrics ([147],[149],[5]), macroscopic structures of grey matter ([53],[79],[94]), sulci surfaces ([160],[17]) and interior structure of cerebellar white matter ([183],[88]). It is relevant to mention that, to the best of the author's knowledge, only one study has examined the entire shape of the brain: [79].

Fractal Dimension may be helpful as an index to quantify structural or functional complexity of the neural system during the stages of development, degeneration, reorganisation, or evolution. This is because it can change dynamically in these processes [88].

3.2.3.1 Fractal Dimension

In general, *fractal dimension* is an estimate of how much space a set occupies near to each of its points. Although there is a wide variety of 'fractal dimensions' in [49], the principal definition is the *Hausdorff dimension*. The Hausdorff dimension can be defined for any set $\mathbf{F} \subset \mathbb{R}^n$ formally as:

$$dim_H\mathbf{F} = inf\{s : \mathcal{H}^s(\mathbf{F}) = 0\} = sup\{s : \mathcal{H}^s(\mathbf{F}) = \infty\} \tag{3.1}$$

where s is considered a positive real number ($s \geq 0$).

This means that, for a given set $\mathbf{F}$, there is a unique value $dim_H \in [0, \infty]$ such that:

$$\mathcal{H}^s(\mathbf{F}) = \begin{cases} \infty & \text{if } 0 \leq s < dim_H\mathbf{F} \\ 0 & s > dim_H\mathbf{F}. \end{cases} \tag{3.2}$$

where $\mathcal{H}^s(\mathbf{F})$ refers to the *Hausdorff measure* [49].

The Hausdorff dimension gives us a way of measuring the size of a set for dimensions s other than the integers $1, 2, 3, \cdots$ [47]. It also has the advantage of being defined for any set and is at the same time mathematically convenient. However, in many cases the Hausdorff dimension is hard to calculate or estimate by computational methods [49].

The Fractal Dimension can be explained in an alternative way. If any object residing in Euclidean dimension $\mathcal{D}$ reduces its linear size by the factor r in each spatial direction, its measure (length, area, or volume) would increase to N times the original, where

$$N = \mathsf{r}^{\mathcal{D}} \tag{3.3}$$

Informally speaking, this means that it is possible to decompose a line, a square or a cube into $\mathsf{r}^{\mathcal{D}}$ self similar pieces, each holding a magnification factor of r (see figure 3.5).

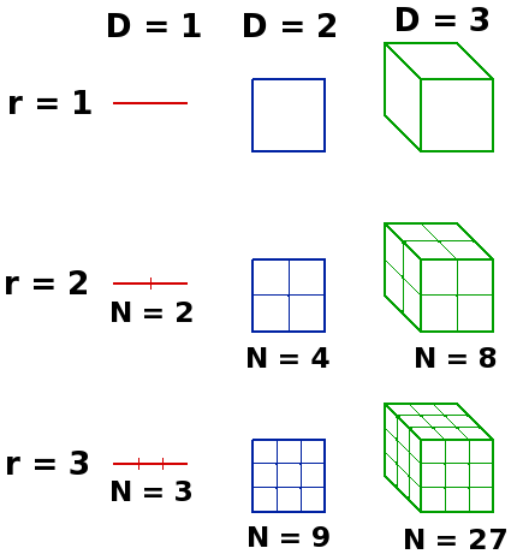


FIGURE 3.5: Alternative definition for fractal dimension

The latter offers another way to specify the dimension of a self-similar object, therefore the dimension is simply the exponent of the number of self-similar pieces with magnification factor r, into which the figure may be broken.

Then, given equation 3.3, it is possible to derive the following definition for dimension:

$$\mathcal{D} = \frac{log\ N(\mathsf{r})}{log\ (\mathsf{r})} \tag{3.4}$$

However, the dimension defined by equation 3.4 is still equal to the Euclidean dimension. By applying the above equation to fractal structure, it is possible to obtain the dimension of fractal structure (or an approximation to the Hausdorff dimension) as a fractional number:

$$\mathcal{D} = \lim_{\delta \to 0} \frac{log\ N(\delta)}{log\ (\frac{1}{\delta})} \tag{3.5}$$

where $N(\delta)$ represents the number of self-similar structures of linear size δ needed to cover the whole structure. Closely related to this is the box counting dimension, which is discussed next.

3.2.3.2 Box counting dimension

The *Box counting dimension* was used in this research because it is relatively easy to calculate mathematically and can be estimated empirically. Let $\mathbf{F}$ be any non-empty bounded subset of $\mathbb{R}^n$ and let $N_\delta(\mathbf{F})$ be the smallest number of sets of diameter at most δ which can cover $\mathbf{F}$. If the lower and upper box-counting dimensions of $\mathbf{F}$ are equal, then we refer to the common value as the *box-counting dimension* or *box dimension* of $\mathbf{F}$ [49]:

$$dim_B\mathbf{F} = \lim_{\delta \to 0} \frac{log\ N_\delta(\mathbf{F})}{-log\ \delta} \tag{3.6}$$

In other words, to obtain the box dimension of a plane set $\mathbf{F}$ we can consider drawing a mesh or a set of boxes of side length δ and then count the number $N_\delta(\mathbf{F})$ that overlap the set for various sizes of δ (hence the name 'box counting') (Figures 3.6 and 3.7). Thus, we may determine the fractal dimension by finding the slope of $log\ N_\delta(\mathbf{F})$ plotted as a function of $-log\ \delta$ [50].

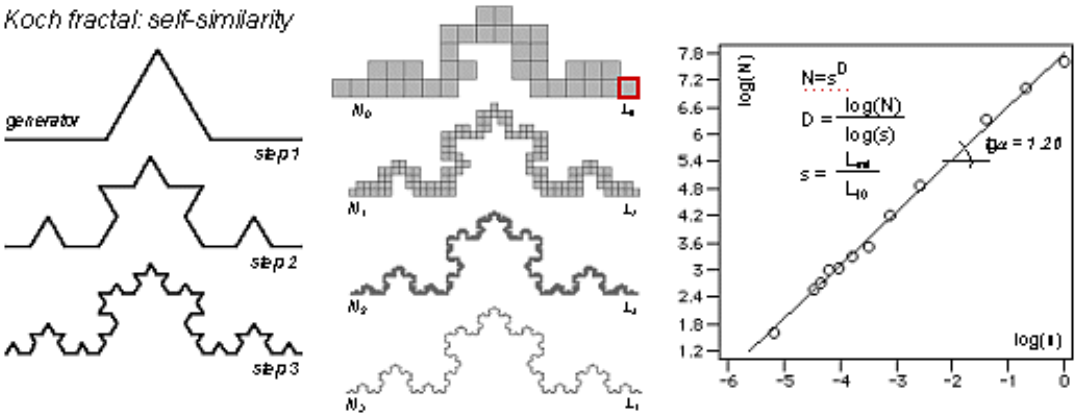


FIGURE 3.6: Box Counting Fractal Dimension for the Koch fractal (Adapted from [137]).

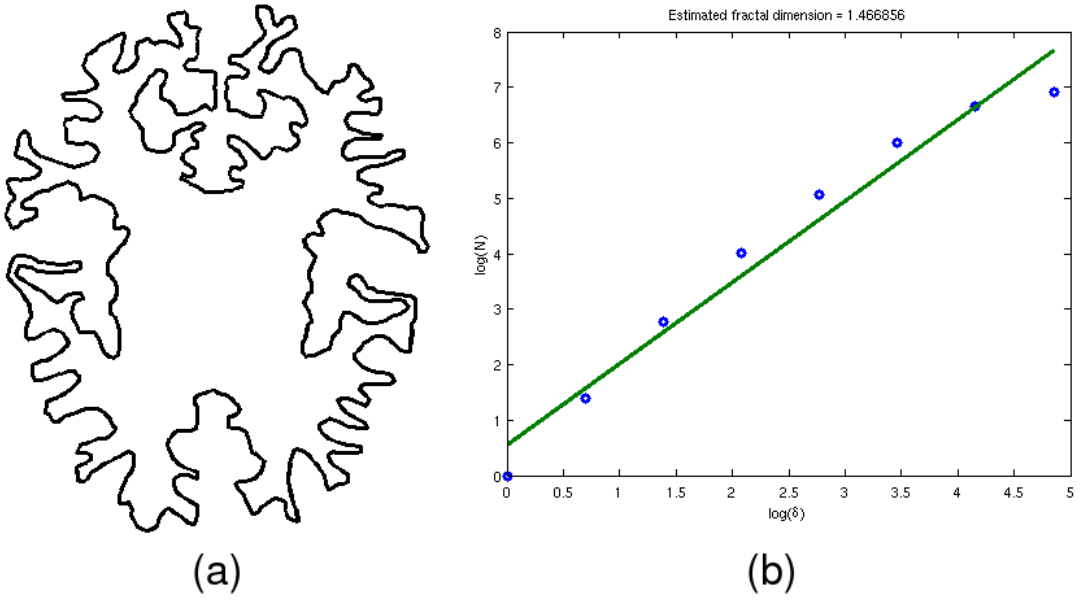


FIGURE 3.7: Box Counting Fractal Dimension result plot (b) for a white matter brain contour (a).

3.3 Bayesian Analysis, Monte Carlo methods and Markov Chains

In this section the main concepts that underpin the proposed Metropolis Hastings based algorithm that was used to obtain the presented results are described.

3.3.1 Bayesian Analysis

In general, *Bayesian Analysis* uses methods for making inferences from data using probability models. The data can be quantities we observe or we wish to learn about [56]. Specifically, Bayesian Analysis is about finding the best methods for making inferences about distributions of random variables. This implies that a random variable B possesses a given type of distribution depending upon an unknown parameter A. So the objective is to to draw some inference concerning A [70].

When we need to make conclusions about a parameter A, these are made in terms of probability statements of Bayesian statistics. The probability statements are

conditional on the observed value of B, simply written as $p(A|B)$ [56]. To make probability statements about A given B, a model that provides a joint probability distribution for A and B is needed. This model is provided by the Bayes' theorem for random variables (or vectors) [56, 84]:

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)} \tag{3.7}$$

where

$$P(B) = \int P(A)P(B|A)dA \tag{3.8}$$

so it can be written in a more compact form as:

$$posterior \ \propto \ prior \times likelihood, \tag{3.9}$$

or in detail

$$\pi(A|B) \propto \mathbb{P}(A) \cdot L(B|A). \tag{3.10}$$

Commonly, the conditional distribution $\pi(A|B)$ is known as the *posterior distribution* of A, $\mathbb{P}(A)$ is referred as the *prior distribution* and, the *likelihood*, $L(B|A)$ is the sampling or data distribution.

The *Prior* distribution, $\mathbb{P}(A)$, is known prior to the experimentation that is carried out to make an inference concerning A [70]. It reflects preliminary knowledge of the 'likely' values of B, so a prior that reflects this is needed.

From Bayes' theorem (3.10), it can be shown that once a probability model is chosen the data B affects the posterior inference only through the function $P(B|A)$, when regarded as a function of A, for a fixed B. This is called the *Likelihood* function [56] and is defined as follows [130]:

$$L(A|B) = L(a_1, \ldots, a_k|b_1, \ldots, b_k) = \prod_{i=1}^{n} f(b_i|a_1, \ldots, a_k) \tag{3.11}$$

Finally, the *Posterior* can be described given a probability distribution of a random variable B that depends on a parameter A. So, the aim here is to make inferences

regarding A on the basis of some observed values of the random variable B and of a prior for A [70]. In other words the posterior distribution can be considered as a representation of the knowledge about A given B.

3.3.2 Monte Carlo methods

Monte Carlo methods are often used when simulating physical and mathematical systems. They provide approximate solutions to many mathematical problems by performing statistical sampling experiments. Thus the analysis of the approximation error is a major factor to take into account when evaluating answers from these methods. The attempt to minimise this error is the reason there are so many different Monte Carlo methods.

Broadly speaking, they can be defined as statistical simulation methods, where statistical simulation is defined in general terms to be any method that utilises sequences of random numbers to perform the simulation. So, it is possible to state that the defining characteristic of Monte Carlo methods is the use of random numbers in its simulations. Hence, the methods are a collection of different techniques that basically all perform the same process. In fact, these methods derive their name from the fact that in the Monte Carlo casinos, the roulette wheels are a good example of a random number generator. Formally speaking, Monte Carlo methods are related to the idea of integration as a method of finding an expectation, but they also provide an extension, by *importance sampling*.

3.3.3 Markov Chains

A Markov chain is a sequence of random variables that evolve over time with a transition probability depending on the particular set in which the chain is drawn. This means that the random variables can only assume values from a certain finite or enumerable infinite set [43]. Additionally, the process A is a Markov chain if it satisfies the Markov condition [64].

A random process $A = A_n, n = 0, 1, 2, \ldots$ taking values in the state space $\mathcal{E}$ (a discrete finite space) is a Markov chain if:

$$P(A_{n+1} \in \mathcal{K} | A_n = a_n, a_{n-1} = A_{n-1}, \ldots, A_0 = a_0) = P(A_{n+1} \in A | A_n = a_n)$$

for all $n \geq 0$, $\mathcal{K} \subseteq \mathcal{E}$ and $a_0, a_1, \ldots, a_n \in \mathcal{E}$.

More intuitively, this says that the probability law to generate the next state in the chain from the current state is governed entirely by the current state and is independent of previous ones.

3.3.4 Markov Chain Monte Carlo Methods

Markov Chain Monte Carlo simulation is a method based on drawing values of A from values that approximate it, and then these draws are corrected to obtain a better approximation to the target posterior distribution $\pi(A|B)$. Since the samples are drawn sequentially and the distribution that generates these samples depends only on the last value drawn, consequently, the draws form a Markov chain [56].

To ensure the success of the algorithm it is necessary to create a Markov process whose stationary distribution is the specified $\pi(A|B)$ and then run the simulation long enough so that the distribution of the current draws is close enough to the stationary distribution [56].

3.3.5 Metropolis-Hastings algorithm

The Metropolis-Hastings (M-H) algorithm can be considered as a general term for a family of Markov chain simulation methods for drawing samples from Bayesian posterior distributions [56].

To understand how the transitions of a Metropolis-Hastings chain are produced, first $\mathcal{E}$ needs to be defined as the state space of a target distribution. Then, choose

for each $a \in \mathcal{E}$ a density $q(a, \cdot)$ on $\mathcal{E}$, so this specifies the transition probabilities (densities) of a Markov chain on the state space $\mathcal{E}$ given that the current state is a. Another requirement is that these transition probabilities/densities $q(a, \cdot)$ should be relatively easy to sample.

Next, suppose the current state of the proposed Markov chain is $A_n = a$, then a state $\mathbf{z}$ is sampled according to $q(a, \cdot)$, so this state $\mathbf{z}$ is a new state of the chain and is accepted with probability:

$$\alpha(a, \mathbf{z}) = min\left\{1, \frac{\pi(\mathbf{z})q(\mathbf{z}, a)}{\pi(a)q(a, \mathbf{z})}\right\}. \tag{3.12}$$

If the proposed state $\mathbf{z}$ is accepted, then the Markov chain moves to $\mathbf{z}$ that is $A_{n+1} = \mathbf{z}$. Otherwise the chain remains in a, that is $A_{n+1} = a$. See [26, 162] for full details.

A summarised version of the Metropolis-Hastings algorithm is [26]:

Algorithm 3.1 Metropolis-Hastings algorithm

Initialise with the arbitrary value $a^{(0)}$
for $j = 1$ to n **do**
  Generate b from $q(a^{(j)}, y)$ and u from $U(0, 1)$
  if $u \leq \alpha(a^{(j)}, b)$ **then**
    set $a^{(j+1)} \leftarrow b$
  else
    set $a^{(j+1)} \leftarrow a^{(j)}$
  end if
end for
Return the values $a^{(1)}, a^{(2)}, \ldots, a^{(n)}$

This comprises some of the basic concepts that are needed to understand the Metropolis-Hastings algorithm. Now, a more detailed explanation of the implementation and adaption to the problem of contour partitioning is given.

3.4 Methodology, modelling and practical considerations

In the following section the implementation details of a M-H algorithm for contour partitioning are presented. Figure 3.8 presents an schematic of the proposed method for contour partitioning that uses the Metropolis-Hastings algorithm with the corresponding adaptions, reviewed before in this section. Basically, the algorithm starts with a random partitions set, then creates a new one choosing one of the partitioning moves, so the posteriors can be calculated for each of the partitions sets. Next, the probability of moving to a new state is calculated, if the new value is accepted then the value is updated and if not it remains in the same state. The process is repeated n steps.

Let a piece-wise linear contour of n points be defined by its coordinates (equation 2.16), and let a given partition of m exclusive subsets be the parameter sets $\Theta = \{\Theta_1, \Theta_2, \ldots, \Theta_m\}$. Let the posterior probability of a given partition be given by

$$\pi(\Theta|\mathcal{X}). \tag{3.13}$$

Then by Bayes' rule (3.10), $\pi(\Theta|\mathcal{X}) \propto L(\mathcal{X}|\Theta)\mathbb{P}(\Theta)$, which is the product of the likelihood of the data given a partition Θ and the *prior probability* of the partition Θ.

As the aim of the partitioning process is to end up with τ_m contour subsections Θ_k of more or less equal size and equal fractal dimension, we can:

1. Model the prior as
$$\mathbb{P}_n(k;\Theta_j) = \frac{e^{-\tau_n}\tau_n^k}{k!}, \tag{3.14}$$
which is a Poisson density with rate parameter τ_n.

2. Model the data distribution by the likelihood function
$$L(\Theta_j|\mathcal{X}) = e^{-|\mathcal{FD}(\Theta_j)-\mathfrak{t}|}, \tag{3.15}$$

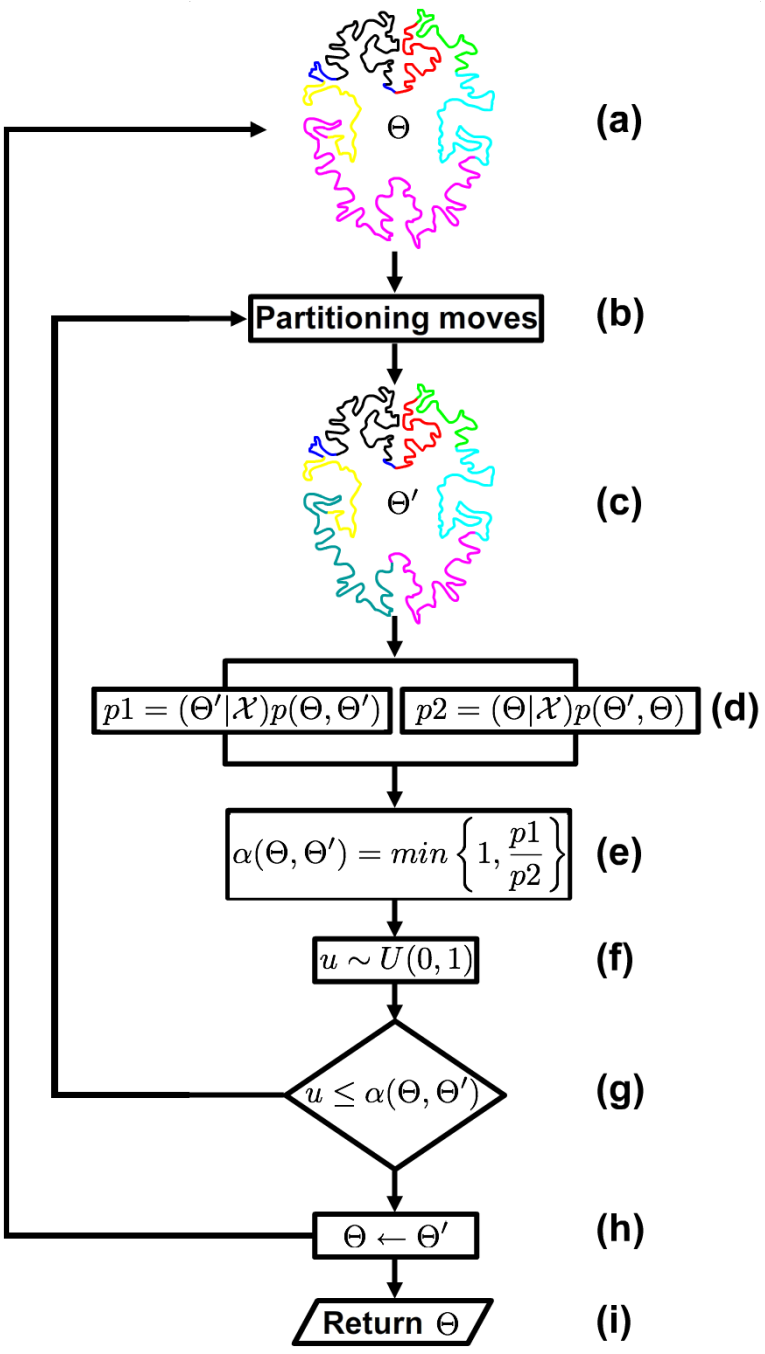


FIGURE 3.8: Schematic that depicts the used Metropolis-Hastings algorithm: the current state **(a)** is a random partition set to start the algorithm, then it updates according to the probability of move. To create the next state of the chain **(c)**, a partitioning move is selected at random (split, merge or alter). Posterior distributions are then calculated for both states **(d)** so the proposal is "accepted" as the next value of the chain **(h)** if α **(e)** drawn from $U(0,1)$ **(f)** satisfies the condition **(g)**. If the proposal is not accepted, then the current value is retained an the process is repeated n times. The posterior probability values **(i)** of the current state are returned and a plot is created (see figure 3.10).

which favours the partition having a fractal dimension, $\mathcal{FD}$, of $\mathfrak{t}$. Here, $\mathfrak{t}$ represents an estimate of the fractal dimension of a contour, so $1 < \mathfrak{t} < 2$.

3. Penalise the posterior by a density to keep the number of partitions to be the desired number, $\tau_m = m$, the required partition size,

$$\mathfrak{p}_m(k;\Theta) = \frac{e^{-\tau_m}\tau_m^k}{k!}. \tag{3.16}$$

Taken together, we can now expand the posterior (3.13) as

$$\pi(\Theta|\mathcal{X}) \propto \mathfrak{p}_m(card[\Theta];\Theta)\prod_{j=1}^{m} L(\Theta_j|\mathcal{X})\mathbb{P}_n(card[\Theta_j];\Theta_j) \tag{3.17}$$

where $(card[\Theta_j])$ is the cardinality of any partition Θ_k.

To keep the posterior calculations numerically stable, they are performed in the *log* domain. Taking natural *logs* of (3.17) we can replace the products by sums of *log* likelihoods and log Poisson densities. Also, for small m, factorial of $(card[\Theta_j])$ can become large, so it is convenient to use Stirlings approximation of factorial,

$$ln\ n! \approx n[ln\ n - 1], \tag{3.18}$$

thus:

$$\begin{aligned} ln\ \pi(\Theta|\mathcal{X}) \propto\ & m(ln[\tau_m] - ln[m] - 1) - \tau_m \\ & - \sum_{j}^{m} |\mathcal{FD}(\Theta_j) - \mathfrak{t}| \\ & + \sum_{j} n_j(ln[\tau_n] - ln[n_j] - 1) - \tau_n, \end{aligned} \tag{3.19}$$

where $n_j = card[\Theta_j]$ and $m = card[\Theta]$. As the M-H algorithm states, given a candidate-generating density, $\pi(\Theta, \Theta')$, of moving from state Θ to Θ' , it is clear that the stationary distribution is reached when $\pi(\Theta'|\mathcal{X})\mathbb{P}(\Theta, \Theta') = \pi(\Theta|\mathcal{X})\pi(\Theta', \Theta)$

is satisfied for all Θ, Θ'. The M-H algorithm performs rejection sampling by comparing the validity of the two states by forming an acceptance ratio:

$$\alpha(\Theta, \Theta') = min\left[1, \frac{\pi(\Theta'|\mathcal{X})\pi(\Theta, \Theta')}{\pi(\Theta|\mathcal{X})\pi(\Theta', \Theta)}\right],$$

and then, if $\alpha = 1$, the move $\Theta \longrightarrow \Theta'$ is accepted, otherwise it is accepted with probability $\alpha \geq r \sim U[0, 1]$. It is easy to see that if the candidate-generating densities are equal for the state changes $\Theta \longrightarrow \Theta'$ and $\Theta' \longrightarrow \Theta$, then the chain automatically moves to a higher probability state.

One of the important features of the approach is the creation of the new states of the Markov chain. In this case a new state can be defined as a new set of partitions of the contour. Every new potential partition set is created by moves in the state space, and there are three allowable potential moves that are chosen at random with probability $\frac{1}{3}$. The first move is *merging* 2 adjacent partitions (fig 3.9-(a)), where a partition is selected in the range $1 \ldots n$, and then a contiguous partition is selected to be merged. The second allowed move is *splitting* a partition (fig 3.9-(a)) at a random point, so this random point is chosen from the set of points of the current selected one. The third move is *Altering* the ending or starting point of a partition (fig 3.9-(b)), which is just a shift in the points to reduce the size of a partition by taking some points of a neighbouring one.

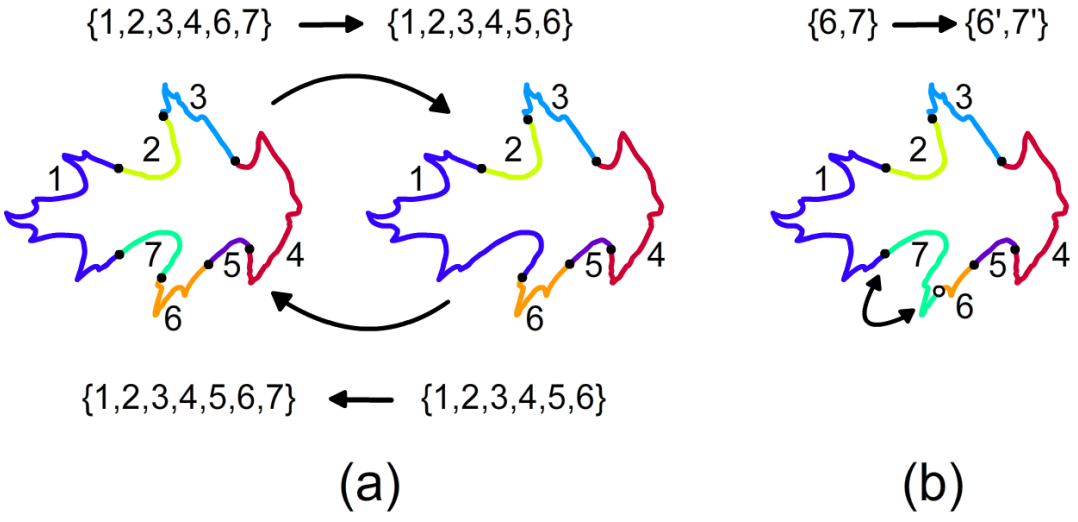


FIGURE 3.9: Illustration of the Split, Merge (a) and Alter Position (b) moves.

In the simulations presented below, for the likelihood function a value of $\mathfrak{t} = 1.57$ is used, which is based on studies of brain contours. This value was obtained by averaging the fractal dimension of 77 contours from different brain parts.

Note that this probability model is setup to assume that there are *local* parts of the contour which have fractal dimension $\mathfrak{t}$. Of course, if it were a fractal and not discretely sampled then by definition this assumption would hold at arbitrary scales.

As with all MCMC implementations, trial and error was used to determine at what number of steps convergence was being achieved. For the contours presented below (each with approximately 3000 points) we began by running 1000 steps, however convergence was observed after about 300 steps (see figure 3.10). Also, we manually selected one point on the contour as a fixed point which marked a line of symmetry: for all contours, this was one of the two points along the principal axis of the contour.

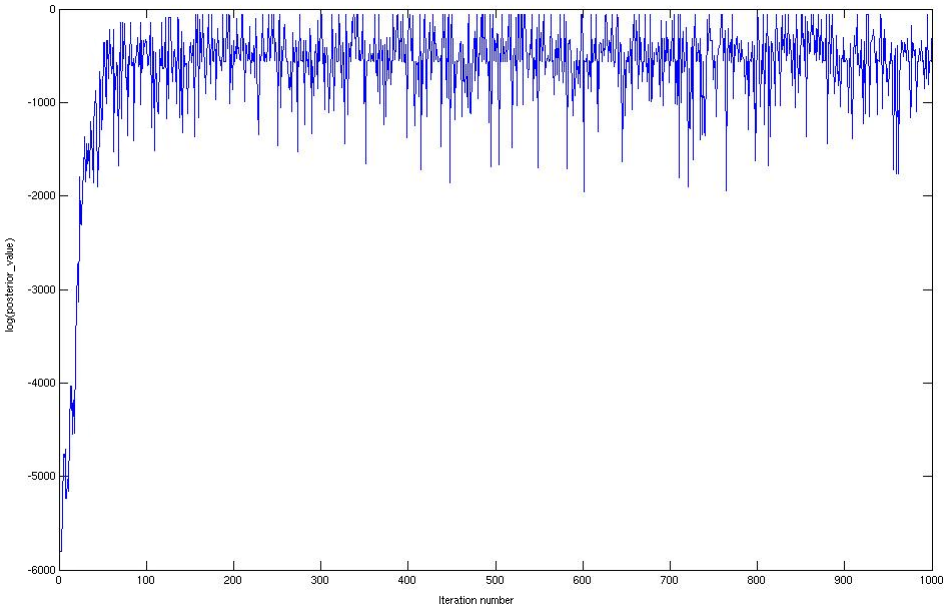


FIGURE 3.10: Typical convergence characteristics. Plot shows log posterior against iteration (move) number. (This plot is for the brain white matter contour 'slice 179' used in the experiments below.)

3.5 Experimental Results

Results of contour partitioning using MCMC simulation are presented on real contours: animal silhouettes (birds, fish) and brain white matter outlines. It was decided to vary the number of required partitions, τ_m e.g. 2, 4, 8 etc. and set the mean number of points per partition by $\tau_n = n/\tau_m$. Up to 1000 steps (moves) were run, although we saw convergence earlier. Each of the figures show results for different number of partitions. The plots of the resulting partitions present a colouring code going from black for the partition with the highest fractal dimension, to yellow, depicting the lowest. For each resulting set of partitions, a table is presented as well with the corresponding information regarding the number of points and the fractal dimension per partition.

The idea of experimenting on the bird and squid contours was to demonstrate that the proposed algorithm is able to find symmetries on the shapes, since these are not objects that present fractal behaviour. The fish/squid contours were from the University of Surrey's SQUID image database: `www.ee.surrey.ac.uk/Research/VSSP/imagedb/demo.html`.

Points	1029	1021
FD	1.4621	1.4523

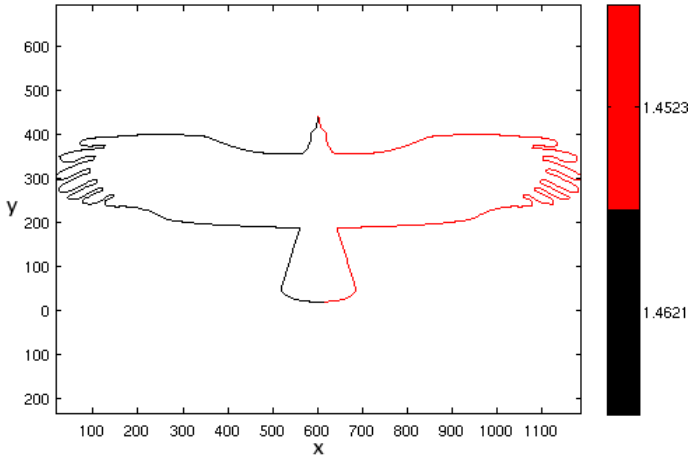
TABLE 3.1: Results for the bird shape: $\tau_m = 2$

Points	522	514	508	508
FD	1.2755	1.2986	1.3922	1.4615

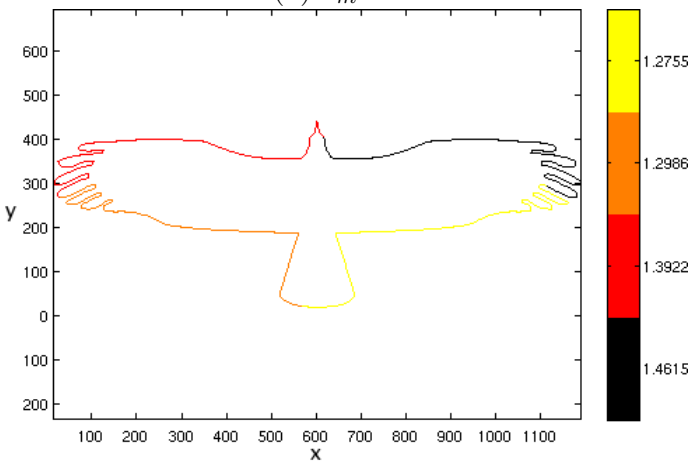
TABLE 3.2: Results for the bird shape: $\tau_m = 4$

Points	281	242	247	246	247	257	261	275
FD	1.2591	1.6376	1.5819	1.3312	1.3312	1.6037	1.6104	1.2591

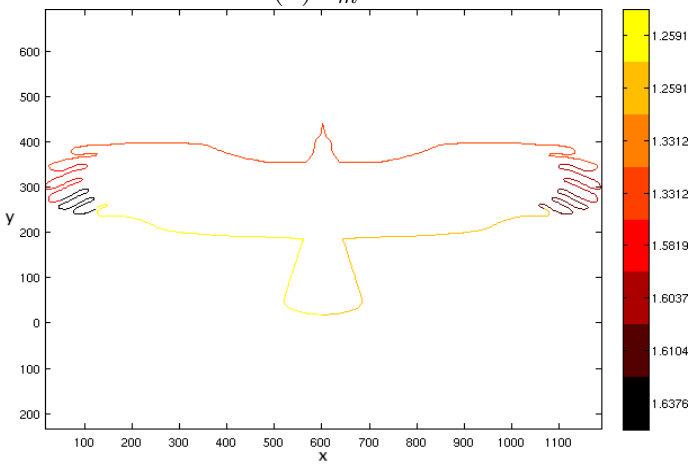
TABLE 3.3: Results for the bird shape: $\tau_m = 8$



(a) $\tau_m = 2$

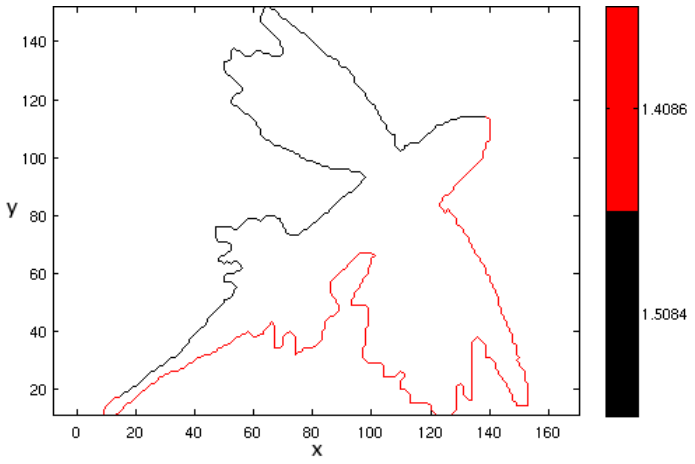


(b) $\tau_m = 4$

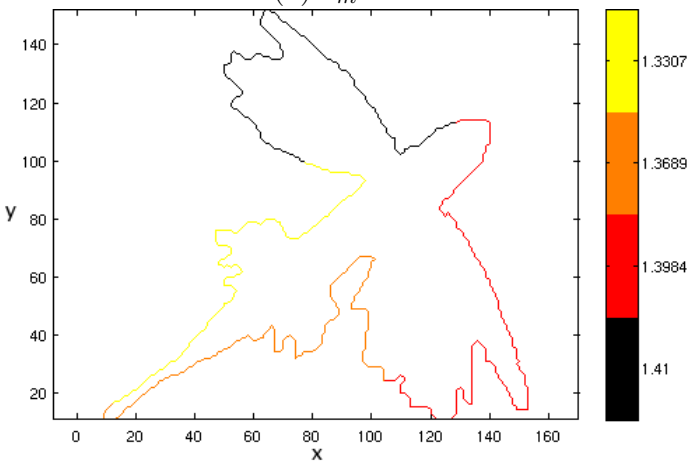


(c) $\tau_m = 8$

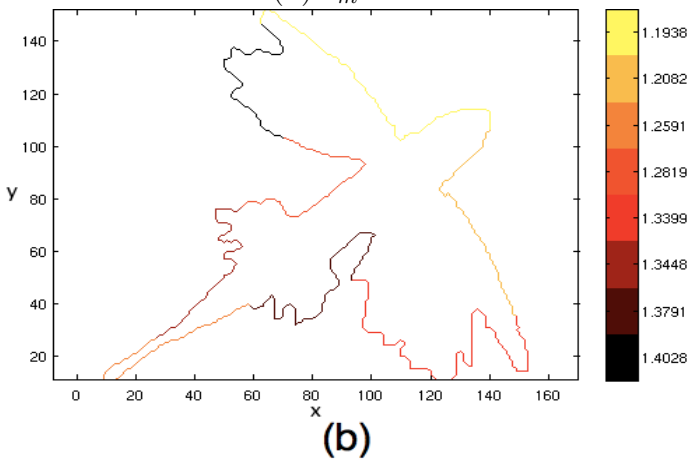
FIGURE 3.11: Example results on the bird contour divided in 2, 4, and 8 partitions



(a) $\tau_m = 2$



(b) $\tau_m = 4$



(c) $\tau_m = 8$

FIGURE 3.12: Example results on the squid database contours divided in 2, 4, and 8

Points	258	256
FD	1.4086	1.5084

TABLE 3.4: Results for the squid shape: $\tau_m = 2$

Points	130	129	129	128
FD	1.3307	1.4100	1.3984	1.3689

TABLE 3.5: Results for the squid shape: $\tau_m = 4$

Points	66	64	59	59	68	65	73	66
FD	1.2591	1.3448	1.2819	1.4028	1.1938	1.2082	1.3399	1.3791

TABLE 3.6: Results for the squid shape: $\tau_m = 8$

For the brain experiments, white matter slices (Subject #04) were used from the database from McGill University which contains 20 anatomical models of normal brains: `www.bic.mni.mcgill.ca/brainweb`.

Points	1636	1653
FD	1.7802	1.7281

TABLE 3.7: Results for the brain contour #179: $\tau_m = 2$

Points	835	792	840	824
FD	1.7313	1.6758	1.6515	1.6778

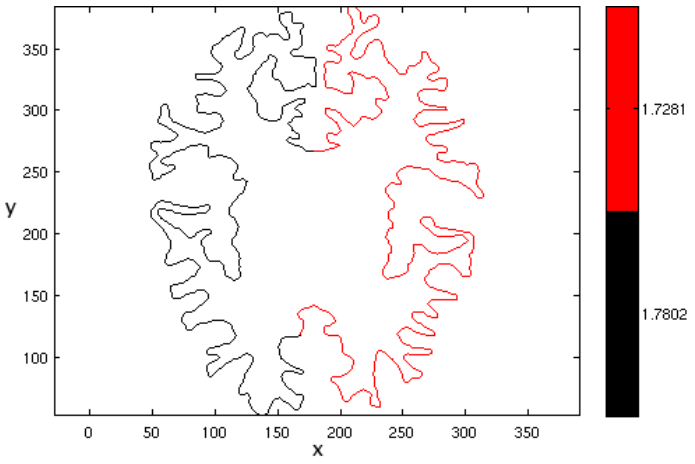
TABLE 3.8: Results for the brain contour #179: $\tau_m = 4$

Points	378	455	441	432	440	403	380	366
FD	1.7107	1.5794	1.6357	1.6138	1.5362	1.6208	1.4745	1.5955

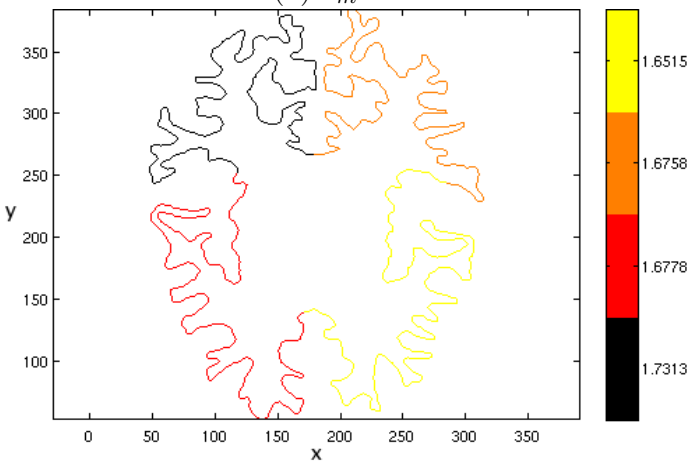
TABLE 3.9: Results for the brain contour #179: $\tau_m = 8$

Points	1636	1653
FD	1.7811	1.7281

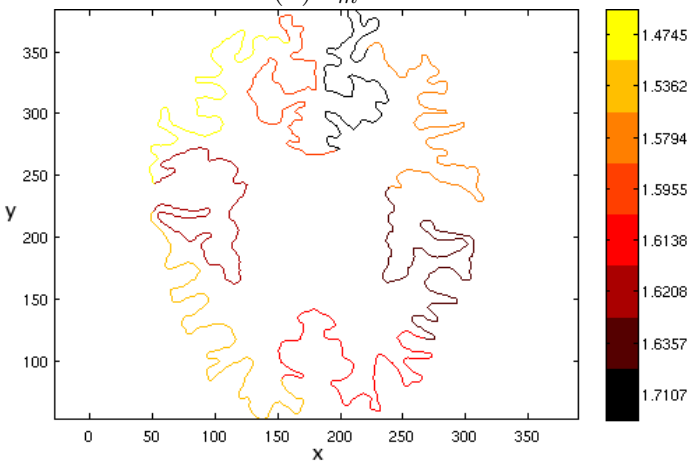
TABLE 3.10: Results for the brain contour #182: $\tau_m = 2$



(a) $\tau_m = 2$

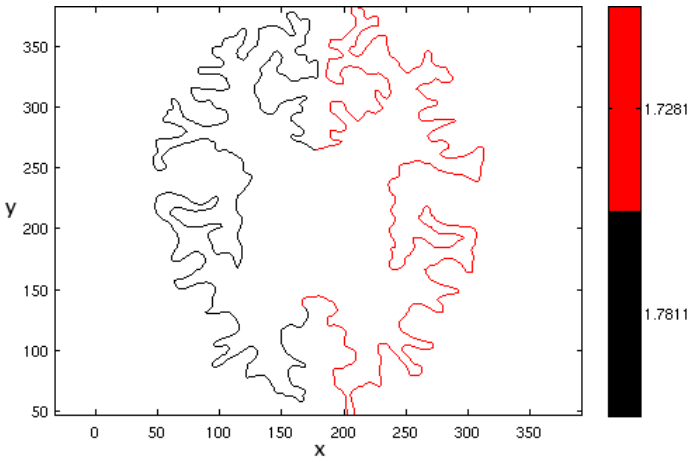


(b) $\tau_m = 4$

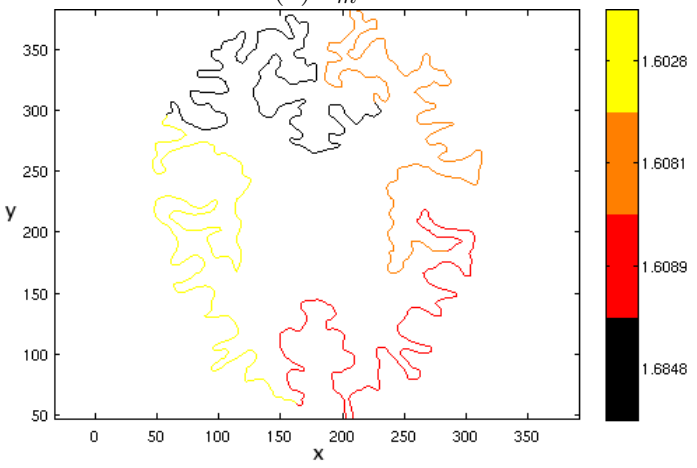


(c) $\tau_m = 8$

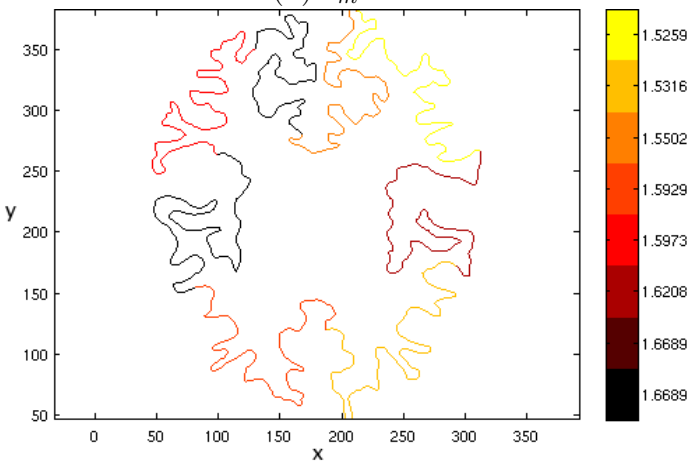
FIGURE 3.13: White matter brain contour partitioning - slice #179. Results shown for different partition sizes after 1000 iterations



(a) $\tau_m = 2$



(b) $\tau_m = 4$



(c) $\tau_m = 8$

FIGURE 3.14: White matter brain contour partitioning - slice #182. Results shown for different partition sizes after 1000 iterations

Points	765	785	755	809
FD	1.6848	1.6081	1.6089	1.6028

TABLE 3.11: Results for the brain contour #182: $\tau_m = 4$

Points	325	343	361	394	430	406	435	424
FD	1.6689	1.5502	1.5259	1.6208	1.5316	1.5929	1.6689	1.5973

TABLE 3.12: Results for the brain contour #182: $\tau_m = 8$

3.6 Evaluation and Discussion

For the bird (figure 3.11) and squid contour (figure 3.12), the data clearly has a one-fold symmetry along its length and this was expected to be found. In both cases the symmetry is close to that required, but since these are relatively low fractal dimension elements it will not change much in moving the partition boundary by a few points. Results for 4 and 8 partitions are less accurate but it is significant to observe that the partitioning is mirrored along the principal axis in both cases. Results for the bird (figure 3.11 and tables 3.1 to 3.3) and squid contour (figure 3.12 and tables 3.4 to 3.6) illustrate that the algorithm is able to find solutions that preserve symmetry and that present similar number of points and similar fractal dimension.

Figures 3.13 and 3.14 show results on a contour taken from white matter segmentation of two slices of a MRI brain image. The contours are similar as they are taken from brain slices which are close by to each other (slices #179 and #182). Likewise, tables 3.7, 3.8, 3.9, 3.10, 3.11 and 3.12 show the numerical results for the brain contours, showing once again that the solutions are good enough because they present similar number of points and similar fractal dimension in each resulting partition set. The MCMC partitioning was run for 500 steps for these cases, because the increase in the number of points for these shapes (3000 points) increases the convergence time. A point on the line which separates the left and right hemispheres of the brain, (on the sagittal line) was marked by hand. The symmetries of the two halves, and the anterior and posterior lobes are correctly

found in both case ($\tau_m = 2$ and $\tau_m = 4$). The results illustrate that the likelihood is sufficiently strong to drive the result away from the Poisson rate of $\tau_m = 8$. As an observation, if this were used to partition all the brain slices from an experimental set, a way to achieve better correspondence between slices might be to initialise neighbouring slices with converged partitions from the current slice.

Poisson distribution was chosen here since it is a discrete probability distribution that expresses the probability of a number of events occurring in a fixed period of time, if these events occur with a known average rate and independently of the time since the last event. Hence, Poisson distributions are generally a natural choice for constructing a Markov chain because of the similarity between the behaviour of both the systems. Markov chains and Markov processes are memoryless processes were the next event in the chain, strictly depends on the current event only. The switch between two events in a Markov chain is dependent on the current state and time interval between the events. These two basic properties make Poisson distribution suitable to model a Markov chain. As mentioned before, the Poisson process specifies a fixed arrival rate, then by this assumption, the rate of arrivals or in this case the number of contour partitions is being fixed (eq. 3.14). The same is done for the penalisation term to fix the required partition size (eq. 3.16). Additionally, there is scope to model the likelihood function in different ways. We chose to use fractal dimension but mean curvature could be a viable alternative, but also, our density uses an absolute norm $|\mathcal{FD}(\Theta_j) - \mathfrak{t}|$ and perhaps a Gaussian could be used instead.

For a genuine fractal curve, by definition, any arbitrary partition should have the same fractal dimension. Even contours found in biology are not always fractals but can exhibit self-similarity, they must be discretely sampled and they may be made up of parts which are fractal and non-fractal. Brain contours are space filling (in 2D and 3D) and the presented method shows potential in being able to produce a reasonable set of partitions. In the next chapter it is shown how the resulting sets from contour partitioning are useful for analysis of local variation in brain structures, results from this chapter only illustrate how partitions can be made taking in count the fractal dimension cast in a MCMC simulation.

The Metropolis-Hastings algorithm is a stochastic method that that prevents the processes to falling into local minima, therefore is able to search through large areas of parameter space in a single run. Moreover, it has been successfully used in many optimisation problems as well as in a variety of areas. By using a Bayesian framework and a stochastic sampler, the convergence is theoretically guaranteed: in practice, our experiments show convergence after a relatively modest number of moves. Even though convergence seems to be reached even after 300 steps, in fact taking the last value of the chain is actually the same as taking any value after convergence has been reached. The proposed MCMC simulation approach has some important characteristics. It is only necessary to set the number of required partitions and the simulation is able to effectively search the state space and adapt locally to shrink and grow to equalise the fractal dimension over the whole contour. This would not be possible to do exhaustively for nontrivial numbers of segments. Likewise, the idea here was to explore an alternative solution to this problem in the fashion of the Bayesian paradigm and to gain insight about the complexity of the brain structure.

Nevertheless, the described approach faces two major problems. The first problem is related to the need for a fixed set of partitions. The use of the MCMC simulation leads to different sets of valid results (see figure 3.15), but ideally an effective method should derive a unique result. The second problem is that the Fractal Dimension is not a stable enough criteria to find the best set of partitions. This means that due to the variability of the Fractal Dimension in self similar curves, this should not be the only criterion to generate the contour partitions, especially if objects that do not present any self similarity are used for experimentation.

3.7 Summary

The chapter began with a review of the two basic topics useful to understand the proposed method: fractal analysis and MCMC methods. A *fractal* can be defined as an object or quantity that unfolds self-similarity on all scales [170]. Although

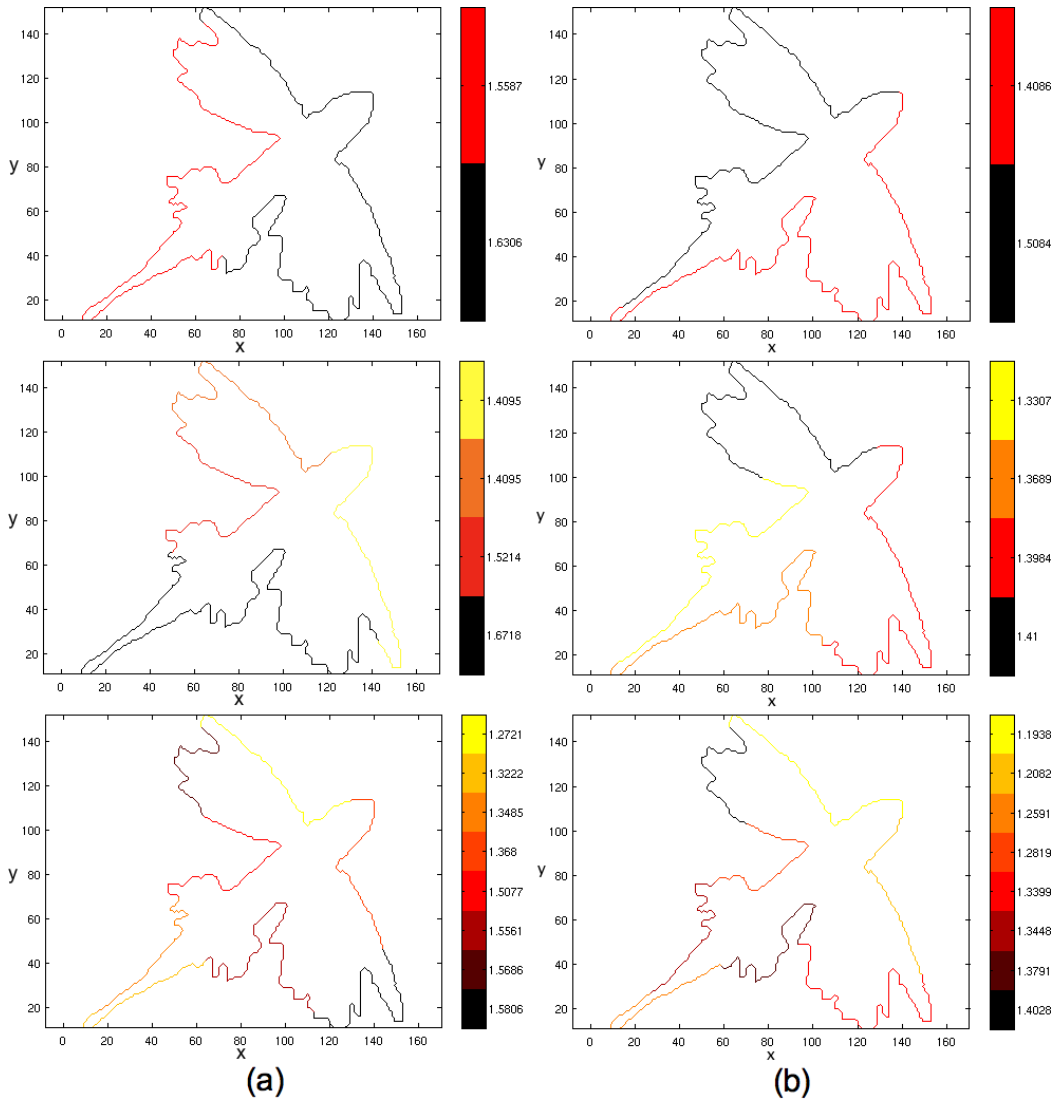


FIGURE 3.15: Two different results from the MCMC sampler: column (b) is the same as figure 3.12. Even though both partitions sets present valid results it is desirable to have a unique solution.

the object does not necessarily exhibit exactly the same structure at all scales, the same 'type' of structures can be viewed on all scales. *Space filling curves* are a type of fractal, and intuitively they can be regarded as a continuous path that visits each point exactly once and never crossing itself, so they fill the space without leaving 'holes'. The fractal geometry of shapes can be a useful tool to describe and understand biological systems with fractal properties. In general, fractal analysis can quantify the irregularity of a fractal object, with the fractal dimension acting as an index or descriptor for the shape complexity. *Fractal dimension* is an estimate of how much space a set occupies near to each of its points. Although there is a

wide variety of 'fractal dimensions' in [49], the principal definition is the *Hausdorff dimension*. Closely related to this is the box counting dimension, that was used in this research because it is relatively easy to calculate mathematically and can be estimated empirically.

This chapter has described and given details about the implementation of a method of modelling fractal curves, such as the boundary of brain white matter, and partitioning such curves into segments having equal fractal dimension. Since the solution space, for a given number of contour points and a required set of partitions is very large, a Bayesian framework of MCMC and a sampler based on the Metropolis-Hastings test was used. Results on both simple contours (animal silhouettes) and space-filling brain contours were presented and showed the convergence characteristics of the method. A discussion of the results and the approach problems was included as well.

From the results and the discussion, it becomes evident that other shape descriptors should be used to generate the partitions. It is possible to employ curvature using the zero-crossings/extrema points generated by the contour evolutions of the Curvature Scale Space method [112], which is discussed in the next chapter.

Chapter 4

Curvature Scale Space Representation for Contour Localisation

4.1 Introduction

The previous chapter introduced a method for partitioning curves that used fractal dimension and the Metropolis-Hastings test. The described approach faces a major problem as Fractal Dimension is not a stable enough criteria to find the best set of partitions: it produces multiple solutions, and there is the need for a unique one. As an alternative solution, in this chapter a method for contour partitioning based on the use of Curvature Scale Space to localise the contour and create sets of partitions in a supervised way is presented. The novel contribution of this work is the creation of a method that provides a simple way to cut-up self similar contours and obtains a set of meaningful shapes useful for shape analysis. The first part of the chapter describes briefly the concept of scale space representation which forms the basis of a CSS. Then in the following section, the CSS representation and its mathematical framework is introduced. Next, a description of the method and a explanation of how this can be incorporated into the analysis of local shape

variation proposed in Chapter 1 are given. Results on white-matter and leaf contours are shown, and finally in the last section, a discussion of these along with proposals on how this work can be taken forward are presented.

4.2 Scale Space representation

Scale-space theory is a framework for multi-scale image representation that has been developed by the computer vision community and complements ideas from physics and biological vision. The basic idea is to use the multi scale nature of real-world objects, implying that such objects may be perceived in different ways depending on the scale of observation. In computer vision and image analysis, the concept of scale is particularly important to design methods for deriving information from images and multi-dimensional signals. The idea of a scale-space representation of image data is, that in the absence of prior information about the scales that are appropriate for a given visual task, the only reasonable approach is to represent the data at multiple scales [86].

One of the precursor methods to represent the input data at multiple scales was given by Witkin [172]. This method breaks the curve into several segments such that each segment is a single valued function. Then a *scale space image* of the curve is constructed. Here the idea is to take a one-dimensional signal and first expand it into a two-dimensional scale-space image by convolution with Gaussians over a continuous sequence of sizes. The problem of this representation is that it is not invariant under rotation [111]. Nevertheless, the idea behind this work is taken by Mokhtarian and Mackworth to develop a more efficient method: the Scale Space Image of a planar curve or Curvature Scale Space (CSS) image [111]. The use of CSS as proposed by Mokhtarian in [112] and later applied by Abbasi et al [1] in their work on contour modelling for retrieval (the University of Surrey's SQUIDS retrieval-by example database), is also an important and useful development (figure 1.4-(e)).

4.3 Curvature Scale space representation

The Curvature Scale Space (CSS) representation finds its roots in the curvature deformation and heat equation [109]. It can be regarded as a multi-scale organisation of invariant geometric figures (curvature zero-crossing points and/or extrema) of a planar curve. This shape representation for planar curves presents some basic properties [110, 108]:

- It is robust with respect to noise, scale and orientation changes of the objects.
- Retains local information of the input shape. This is due to the fact that every concavity or convexity on the shape has its own corresponding contour on the CSS image. In addition, every point in the horizontal axis of the CSS image has its corresponding points on the actual boundary.
- The evolved contours generated by the method can be regarded as an early version of *active contours (snakes)* as they present similar behaviour in the absence of external constraints. In this case, both tend to shrink and minimise their curvature.

The CSS image is a fundamental concept of this study, since part of it is based on this representation. At present, an evaluation of the CSS representation according to the proposed criteria in [113] is presented.

The first criterion is invariance; meaning that the shape of the input curve should not change under shape-preserving transformations, namely rotation, scaling and translation. Neither translation or scaling causes changes to the CSS representations, and rotation causes only a horizontal shift of it. The second criterion is uniqueness, that requires that two curves with different shapes may be mapped to different representations. A CSS fulfils this criterion by showing that a planar curve can be reconstructed from any of its CSS representations. The third criterion is stability. This claims that any small change in the shape of the curve leads to a small change in its representation and vice-versa. It has been demonstrated that planar curves remain connected during the evolution process and therefore

their CSS representation can always be constructed. Also, experiments carried out show that the CSS representations are stable under the presence of significant uniform and non-uniform noise of the curves they represent.

The following criteria make reference to properties that the CSS representation has to be suitable for shape tasks. Local support is a criterion referring to the fact that very often it is necessary to recognise that the shape of a segment of a curve has the same shape as another curve segment. Since the CSS representation can be computed for open contours and, except near endpoints of the curve, will resemble the corresponding representation for a closed contour of which it is a part. Therefore, it is believed that this representation satisfies the local support criterion. The efficiency criterion means that the computational complexity should be a low-order polynomial. The CSS representations are computed by convolutions, so this process can be performed efficiently by using Fast Fourier transforms, parallelising the operations, expressing convolutions that involve Gaussians of large widths in terms of convolutions involving Gaussians of small widths only or convolving only in small neighbourhoods of the existing zero-crossings in order to find the zero-crossings just at the next higher levels. The next criteria makes reference to shape properties. A useful representation needs to be able to determine such properties of a curve. For example, if a curve has a symmetric shape, it might be desirable to be able to determine that fact from its representation (symmetry criterion). What is more, if the shape of a whole curve or part of it is the same as the shape of part of another curve, it might be useful to determine that relationship using its representations (part/whole criterion). This criterion is satisfied by the CSS since the representation of symmetric curves are also symmetric since a symmetric curve also has symmetric curvature zero-crossings across scales.

The representation, as mentioned before, is computed by convolving a parametric representation of a curve with a Gaussian function. This process of describing a curve at increasing levels of abstraction is referred as the *evolution* of the curve (Figure 4.1-(b)). As the standard deviation of the Gaussian varies from a small to a large value, it is possible to extract the curvature zero-crossing points of the resulting curves. The final step is the construction of the CSS image (Figure

4.2), which is obtained by plotting the smoothing factor against the number of curvature zero crossings. Only the generation of evolved versions of the curve and the locations of the curvature zero-crossings are relevant for this work, for further details see [113].

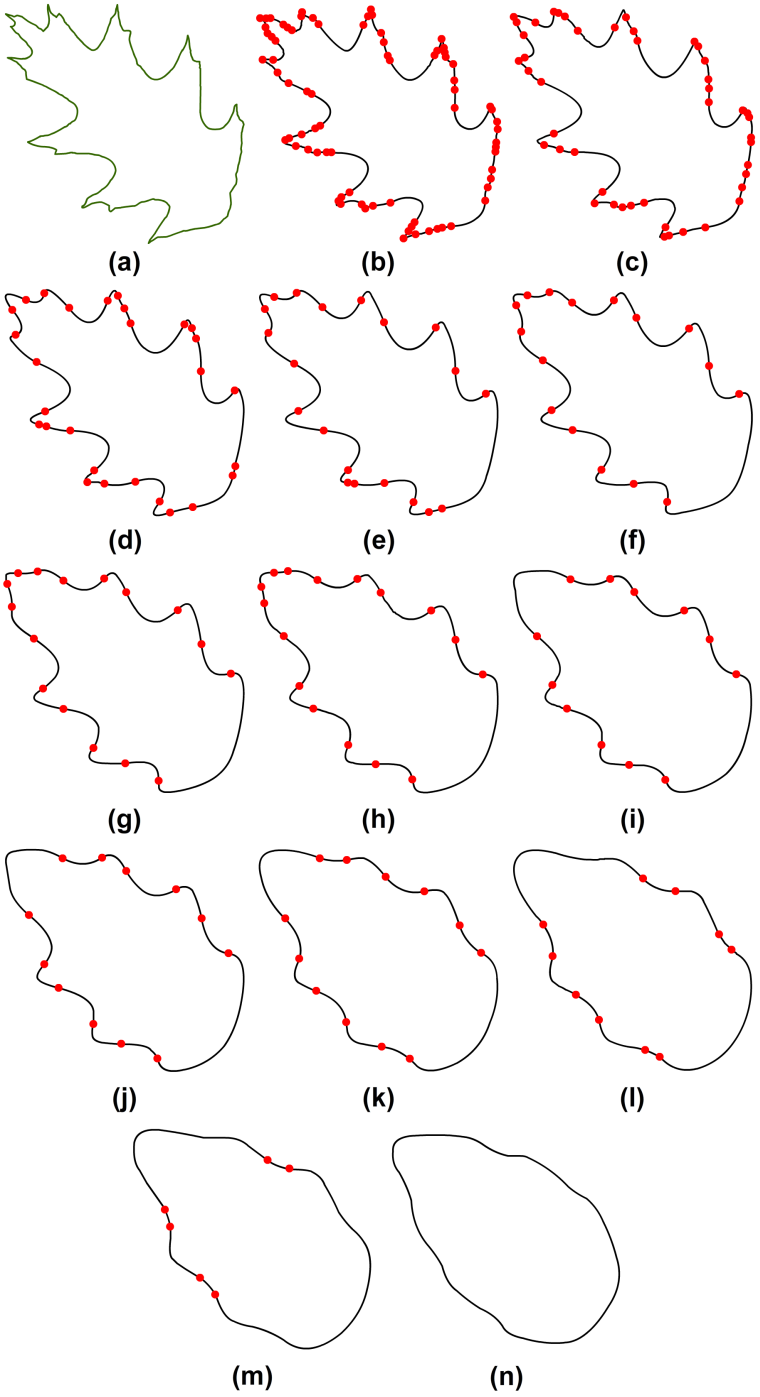


FIGURE 4.1: Evolution of a contour: **(a)** Original Contour and **(b)**-**(n)** Evolved versions or evolution of the curve.

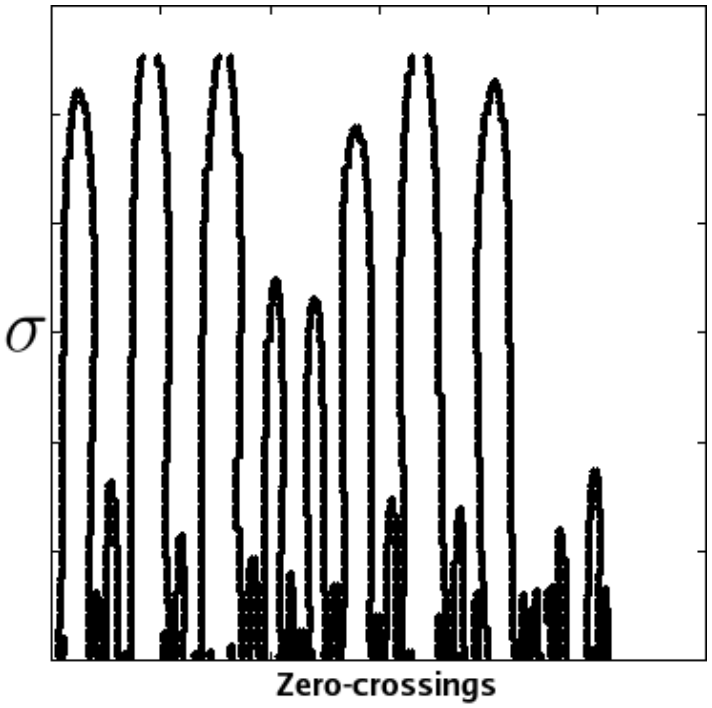


FIGURE 4.2: Final step of the CSS representation: CSS Image obtained by plotting the smoothing factor σ against the number of curvature zero crossings.

The following experiments were carried out to test the stability of the curvature scale space image under conditions of noise and rotation. Figure 4.3(a) presents the contour of a leaf of the type Macrophyllum and in 4.3(b) its CSS image. Figure 4.3(c) shows the leaf contour with a significant amount of uniform, random noise added to it. Figure 4.3(d) shows the curvature scale space image of the leaf contour with uniform noise. As expected, the CSS images in figures 4.3(b) and 4.3(d) show differences in detail. Even so, similarities can be found in the basic structures of the two CSS images. Figure 4.3(e) shows the Macrophyllum contour with severe, uniform noise added and figure 4.3(f) shows its corresponding CSS image. Even with the presence of severe noise, similarity can be observed between the two CSS images shown in figures 4.3(b) and (f). The same is valid for contours with non-uniform noise as in figure 4.3(g). Once again, figures 4.3(b) and (h) present similarities in the basic structures and differences in detail. The last experiment on figure 4.3(i) presents the input contour rotated and figure 4.3(j) shows its CSS image with the same characteristics as the others with respect to the CSS image 4.3(b). This experiment shows that the CSS image is robust and stable under

different levels of noise corruption of the contours, and similarly, under rotation of the input curve. Other examples of this can be verified in [113].

4.3.1 Mathematical framework

To build the CSS representation, the curve needs to be considered as a parametric vector equation $\mathcal{C}_i = (x_i, y_i)$, then a series of *evolved versions* of $\mathcal{C}_i$ are produced by increasing σ from 0 to ∞. Every new evolved version is defined as $\mathcal{C}_\sigma = (X(i,\sigma), Y(i,\sigma))$, where

$$X(i,\sigma) = x(i) \circledast g(i,\sigma) \ , \ Y(i,\sigma) = y(i) \circledast g(i,\sigma) \tag{4.1}$$

Here, $\circledast$ denotes the convolution operator and $g(i,\sigma)$ is a Gaussian of width σ [112]. Since the CSS representation contains curvature zero-crossings or extrema points from the evolved version of the input curve, these are calculated directly from any $\mathcal{C}_\sigma$ by:

$$\kappa(i) = \frac{\dot{X}(i,\sigma)\ddot{Y}(i,\sigma) - \dot{Y}(i,\sigma)\ddot{X}(i,\sigma)}{(\dot{X}(i,\sigma)^2 + \dot{Y}(i,\sigma)^2)^{3/2}}, \tag{4.2}$$

where

$$\dot{X}(i,\sigma) = \frac{\partial[x(i) \circledast g(i,\sigma)]}{\partial t} = X(i) \circledast \left(\frac{\partial g(i,\sigma)}{\partial i}\right),$$

$$\ddot{X}(i,\sigma) = \frac{\partial^2[x(i) \circledast g(i,\sigma)]}{\partial i^2} = X(i) \circledast \left(\frac{\partial^2 g(i,\sigma)}{\partial i^2}\right).$$

Similar equations are used to compute $\dot{Y}(i,\sigma)$ and $\ddot{Y}(i,\sigma)$:

$$\dot{Y}(i,\sigma) = \frac{\partial[y(i) \circledast g(i,\sigma)]}{\partial i} = Y(i) \circledast \left(\frac{\partial g(i,\sigma)}{\partial i}\right),$$

$$\ddot{Y}(i,\sigma) = \frac{\partial^2[y(i) \circledast g(i,\sigma)]}{\partial i^2} = Y(i) \circledast \left(\frac{\partial^2 g(i,\sigma)}{\partial i^2}\right).$$

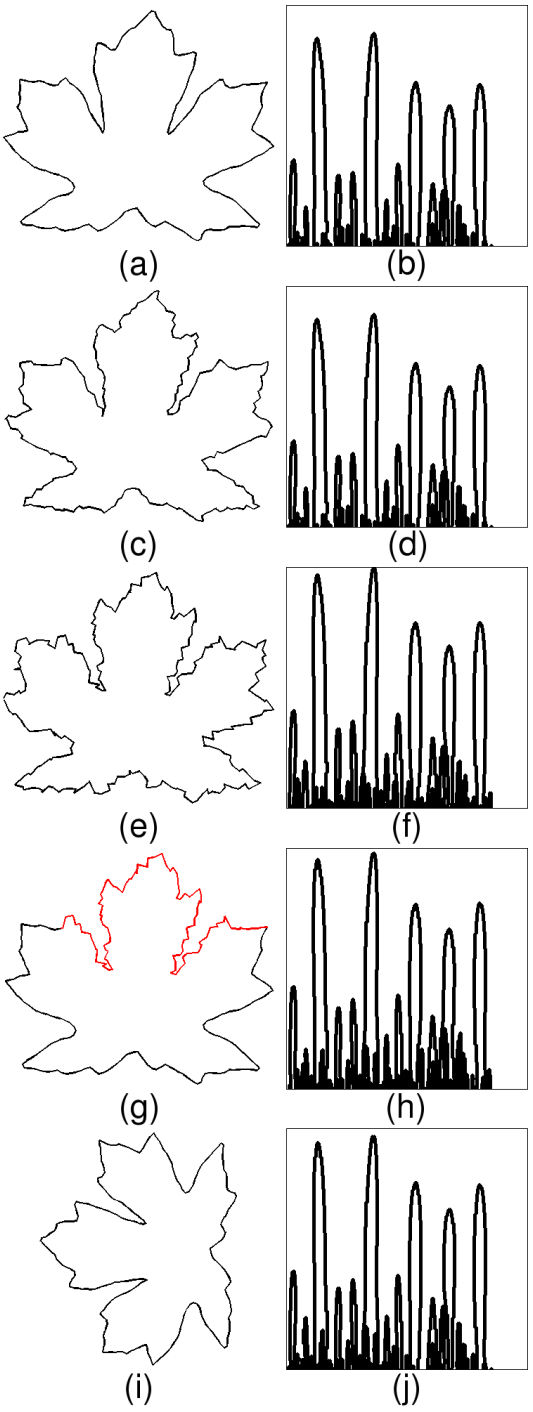


FIGURE 4.3: Results of experiments carried out to test the stability of the curvature scale space image under conditions of noise and rotation. (a) Contour of a leaf of the type Macrophyllum and (b) its CSS image. (c) leaf contour with a significant amount of uniform, random noise added to it. (d) curvature scale space image of the leaf contour with uniform noise. (e) Macrophyllum contour with severe, uniform noise and (f) its corresponding CSS image. (g) contour with non-uniform noise and (h) its corresponding CSS image. (i) rotated input contour and (j) shows its CSS image.

4.4 CSS for Local Shape Modelling

It is important to mention that in this work CSS is not used to describe the shape, since a PDM is the featured shape descriptor. Here, only a part of the CSS representation is used, in this case the evolutions of the curve and the extrema or zero-crossings. The following is an explanation of how these are employed, in the particular case of the proposed local shape model, to create contour partitions.

Given an input contour (Figure 4.4-**(a)**), there is the need to select the appropriate level of smoothing σ. Applying different values of sigma to the contour produces the aforementioned evolutions of the curve (Figure 4.4-**(b)**), and for each evolved contour it is possible to find the zero-crossings (displayed as red dots). This are the points where there is a change in the sign of the curvature of the contour. Then, having selected a value of σ, the set of generated inflection points are used to create the contour partitions (Figure 4.4-**(c)**), where each point constitutes the start and end for each partition. So these points provide a basic but efficient way to create meaningful contour partitions.

4.4.1 Supervised Shape Partitioning and Clustering

Having stated how CSS is used to create partitions, the aim here is the localisation and clustering of the featured model presented on the right side of Figure 1.7 in Chapter 1. The first task is done using CSS and the second using an interface, both in a supervised fashion. This section then, describes the the way these two tasks are incorporated into the local shape modelling framework, for which an overview is shown in figure 4.5-**(b)**. On the mentioned figure, part **(a)** corresponds to the data step in figure 1.7 and part **(c)** corresponds to the steps of registration, PCA and reconstruction, as well in figure 1.7.

To facilitate the research, a tool with a graphical interface was created, which was divided into 3 main sections corresponding to the three main steps of our methodology (Figure 4.5). Refer to Appendix A for a full description of this tool.

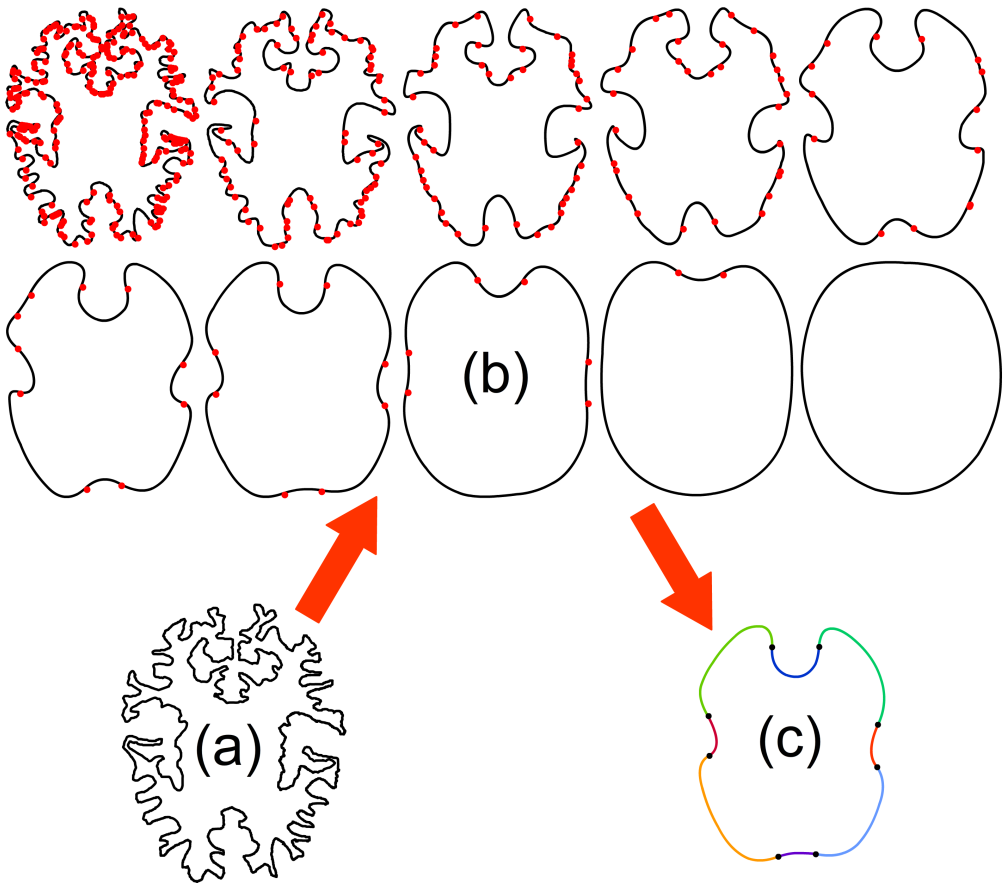


FIGURE 4.4: **(a)** Original contour, **(b)** CSS evolution of a white-matter brain contour. At some appropriate level of smoothing, a set of meaningful partitions can be identified. **(c)** Pairs of zero-crossings (red points) are used to create contour partitions.

Successive steps of the the method will now be described. From the input data set of images, we obtain the contours for each of them as a set of (x, y) coordinates with the method explained in section 2.4.4.1 (Figure 2.10).

The first step is to select a shape that will be the reference one to create the partitions for the rest of the contours in the set. In Figure 4.6-(a) the reference contour is shown together with the set of zero-crossing points, and a pair of these is manually selected to create a reference partition. Afterwards, to create the set of partitions from all the contours in the set, each one is smoothed with the same scale factor σ as the one the reference contour. Here, zero-crossings are taken in combinations of point-pairs and occupying the points between each of these combinations is how partitions are created. After obtaining the set of shapes, the next

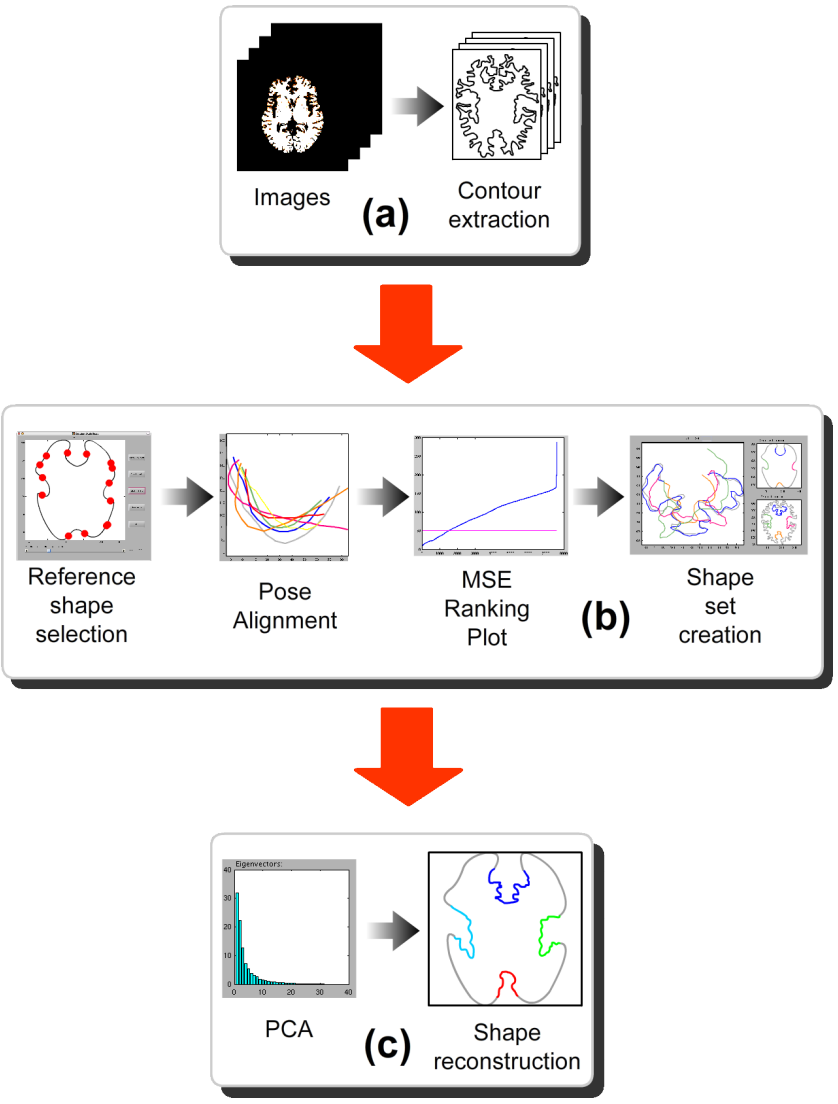


FIGURE 4.5: Overview of the method for proposed LSM with the incorporation of CSS for contour partitioning and the supervised clustering: **(a)** Contours are obtained from MRI brain data; **(b)** CSS is used first to obtain a reference shape from the smoothing of a target contour at some large scale, σ. A pair of zero-crossings selects a prototype section and all others are ranked accordingly by search and rigid alignment; **(c)** Statistical Shape Modelling and reconstruction based on selected modes.

major concern is deciding which of these partitions could be useful for the shape analysis. An MSE ranking plot (Figure 4.6-(c)) is constructed by doing a pose alignment of the partitions (Figure 4.6-(b)) and then calculating the mean square error of Euclidean distances for each partition of the set against the reference shape using:

$$D(\mathcal{C}_m, \mathcal{C}_n) = \sqrt{(x_{mj} - x_{ni})^2 + (y_{mj} - y_{ni})^2} \tag{4.3}$$

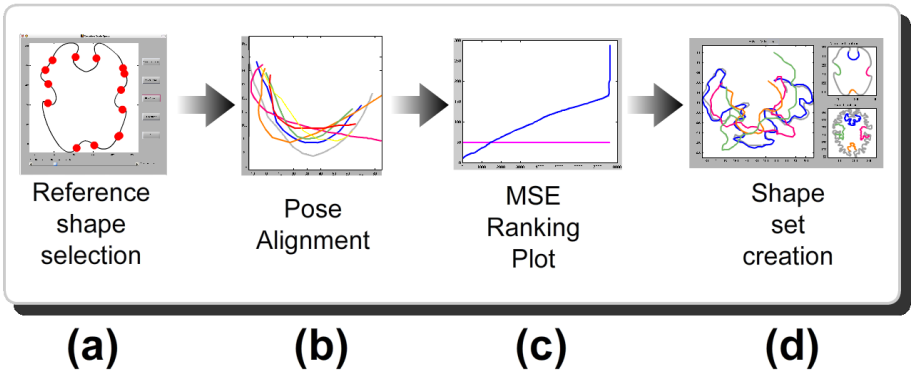


FIGURE 4.6: CSS for contour partitioning and the supervised clustering process: **(a)** A shape is selected that will be the reference one to create the partitions for the rest of the contours in the set. The reference contour is shown together with the set of zero-crossing points, and a pair of these is manually selected to create a reference partition. Then, to create the set of partitions from all the contours in the set, each one is smoothed with the same scale factor σ as the one the reference contour. Here, zero-crossings are taken in combinations of point-pairs and occupying the points between each of these combinations is how partitions are created. After obtaining the set of shapes, a pose alignment of the partitions is performed using the reference partition from previous step **(b)** and then MSE ranking plot is constructed **(c)**. This plot is used to indicate the number of partitions (by setting an error threshold) that are going to be used in the statistical analysis **(d)**.

This plot is used to indicate the number of partitions (by setting an error threshold) that are going to be used in the statistical analysis (Figure 4.6-(d)). Now, having a smaller set of shapes is possible to perform shape analysis in the way described in section 2.4.4.3. Finally, a reconstruction of the selected partitions can be visualised along with the modes of variation of the shape set (Details in section 2.4.4.4). In the next section this will illustrated in a more effective way with examples from two different data sets.

4.5 Experimental Results using natural contours: MRI Brain and Leaf data sets

A set of 40 simulated digital brain phantom images from normal subjects was used in this test. Each digital brain was created by registering and averaging four

T1, T2 and PD-weighted MRI scans from normal adults (Figure 4.7). For further details on the construction of these simulated brain images see [10]. The second set comprises 40 leaf contours of the type *Quercus Kelloggii* from the Oregon State University ([40]).

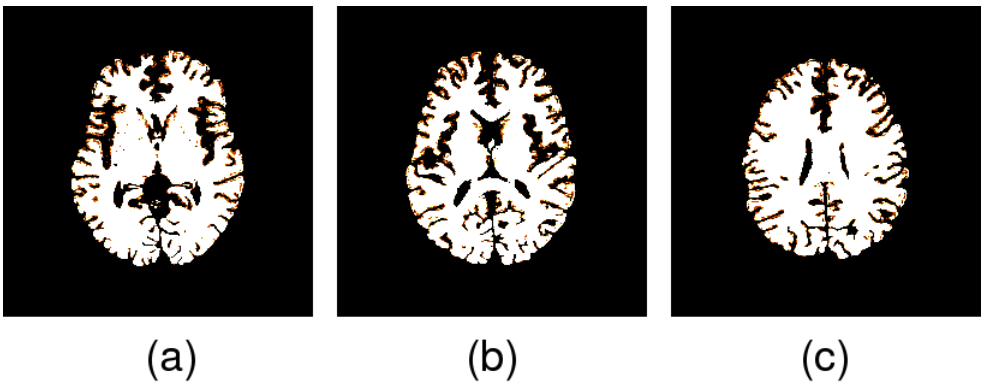


FIGURE 4.7: Three samples of brain white matter MRI simulator from subject 4 at different coordinates.

Having a set of contours (in this case brain contours), any of these can be used to obtain a prototype partition. This partition is the one that is used to create the ranking plot (Figure 4.8-(b)). It is obtained by selecting a pair of zero-crossings from a reference contour previously smoothed at a given scale factor σ (Figure 4.8-(a)).

Afterwards, a set of partitions is created by smoothing each of the available contours of the set with the same scale factor as the one in the reference contour, then the zero-crossings are taken in combinations of two points and occupying the points between each of these is how each partition is created. After obtaining the set of shapes, the next issue is how to decide which of all these partitions might be useful for the shape analysis. Pose alignment is performed and each partition is ranked with the prototype partition (Figure 4.11-(a)) following the method defined in the last section. Thus, an affine alignment is done over the set, where each shape is aligned to the selected reference shape (See figures 4.9 and 4.10). This is achieved by using the method explained previously in section 2.4.4.2.

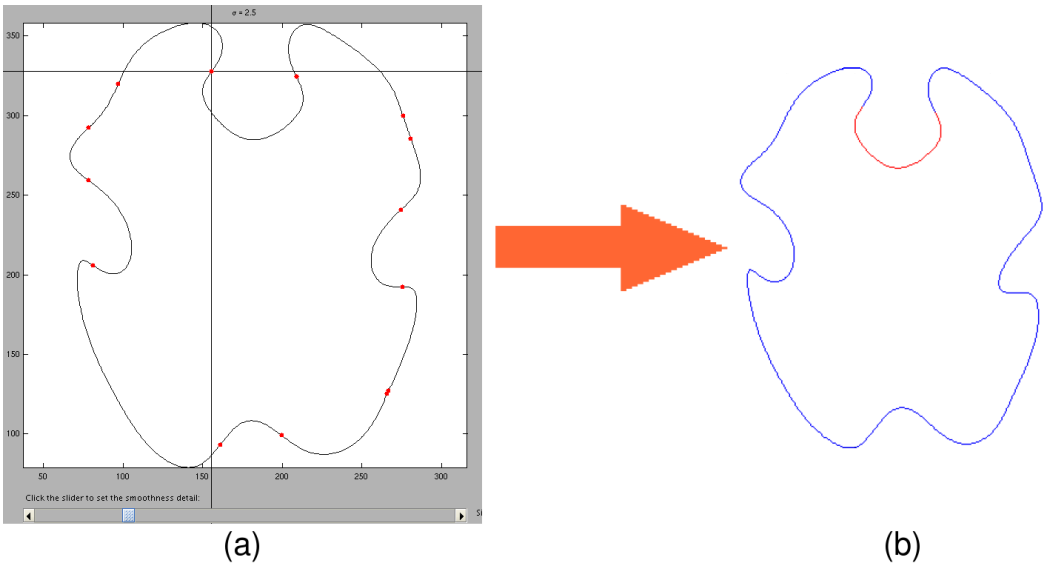


FIGURE 4.8: Creation of the prototype partition from a reference contour: (a) Selection of the prototype partition using a pair of zero crossings and (b) result of the selection in red.

Finally, Figure 4.11-(c) shows the smoothed CSS contour and the corresponding zero-crossings based partitions that are mapped to the original contour. The interface allows any given partitions to be selected and ranked against all remaining partitions. Since the brain contours exhibit self-similarity, using the CSS, at certain scales the potential partitions derived from the extrema points look the same (Figure 4.11-(b) and (c)). It is the case as well of the leaf contours as can be seen in figure 4.12. Here it is possible to observe that given the reference partition it is likely to find similar ones according to the manual threshold over the smoothed version of the contours.

Figures 4.13 and 4.14 shows reconstructions from the local shape model, with the partitions blended back into a smooth version of the original (see equation 2.28). It can be noted from the results that the local parts are similar and localised according to the prototype selection. It is important to mention that the brain phantoms are only used for the purpose of illustration. This is because they are suitable for generating realistic simulations where for instance, inter-subject anatomical variability's can be considered [10].

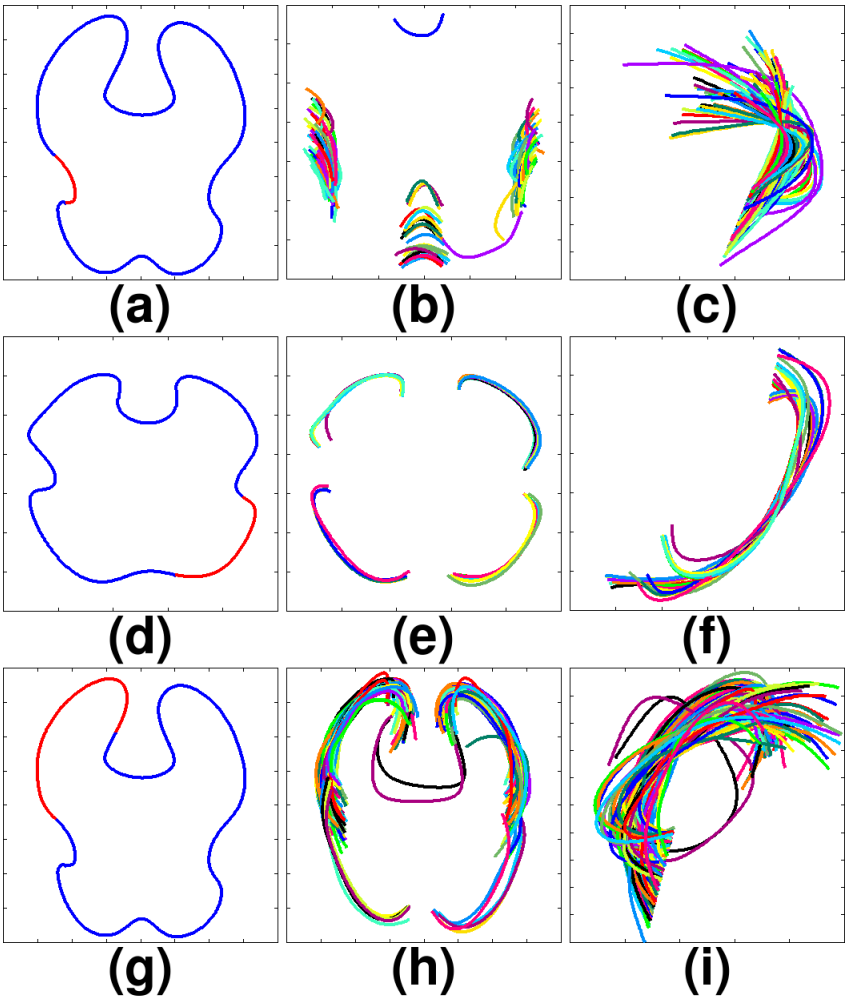


FIGURE 4.9: Example of pose alignment step needed to rank the partitions and perform the shape analysis, for the white matter example. **(a)**, **(d)** and **(g)** Prototype shape; **(b)**, **(e)** and **(h)** parts to be aligned from a different white matter contour than the prototype and **(c)**, **(f)** and **(i)** alignment of the selected prototype with the contour localisations of the second column.

4.6 Evaluation and Discussion

The main contribution of this chapter has been the creation of a simple way to extract similar partitions of the data input contours. The principal feature of this method has been the use of the consistency of the curvature extrema at low resolutions of the contour to partition similar parts of a irregular shape, such as brain and leaf contours. This localisation allows a linear shape space to be used directly on the aligned parts. The extrema points generated by the CSS representation are constant over a range of scales, i.e., as σ increases no new

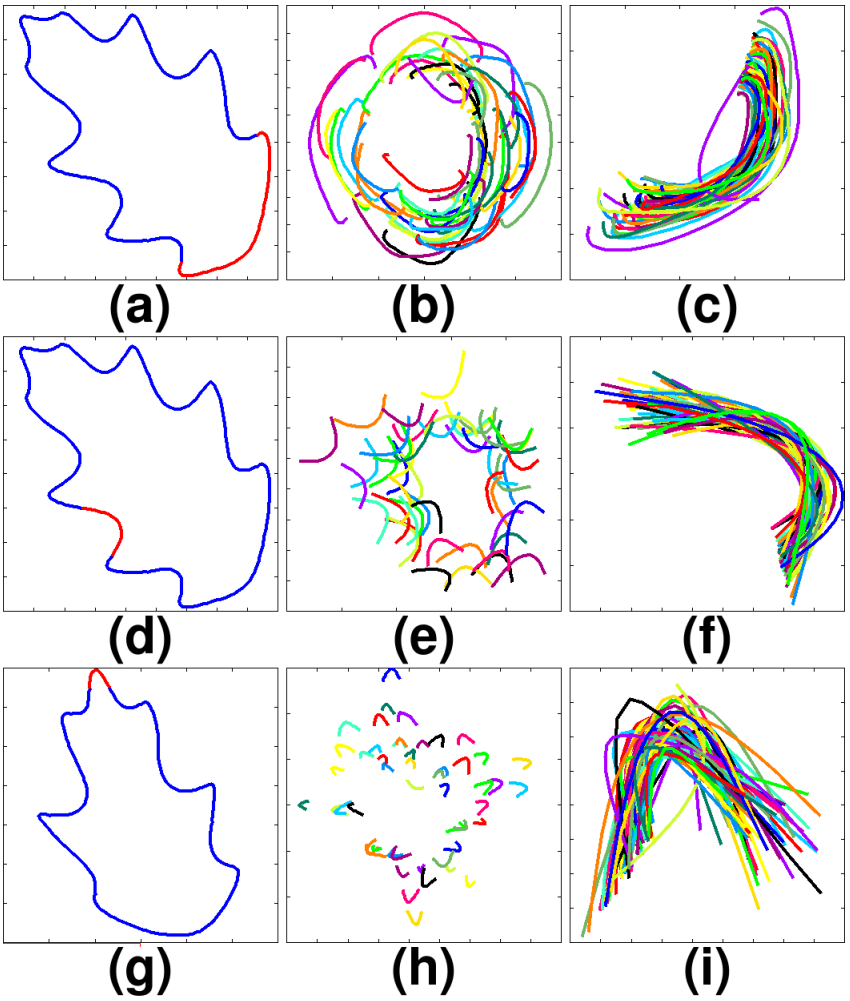


FIGURE 4.10: Example of pose alignment step needed to rank the partitions and perform the shape analysis, for the leaf example. **(a)**, **(d)** and **(g)** Prototype shape; **(b)**, **(e)** and **(h)** parts to be aligned from a different leaf contour than the prototype and **(c)**, **(f)** and **(i)** alignment of the selected prototype with the contour localisations of the second column.

curvature zero-crossings can appear on the contour (Figure 4.4). Also, since the contours of interest exhibit self-similarity, using the CSS method it is possible to find that at a certain scale the possible partitions derived from the extrema points look the same (Figures 4.11-(c) and 4.12-(c)). This provides a natural way to identify consistent parts in the contour by local variation. It is important to note that the evolved contours (Figure 4.1) can be regarded as an early type of *active contours (snakes)* as they have similar behaviour in the absence of external constraints. In this case, both tend to shrink and minimise their curvature [108]. Additionally, the CSS itself is a wavelet representation (by Gaussians), so it could

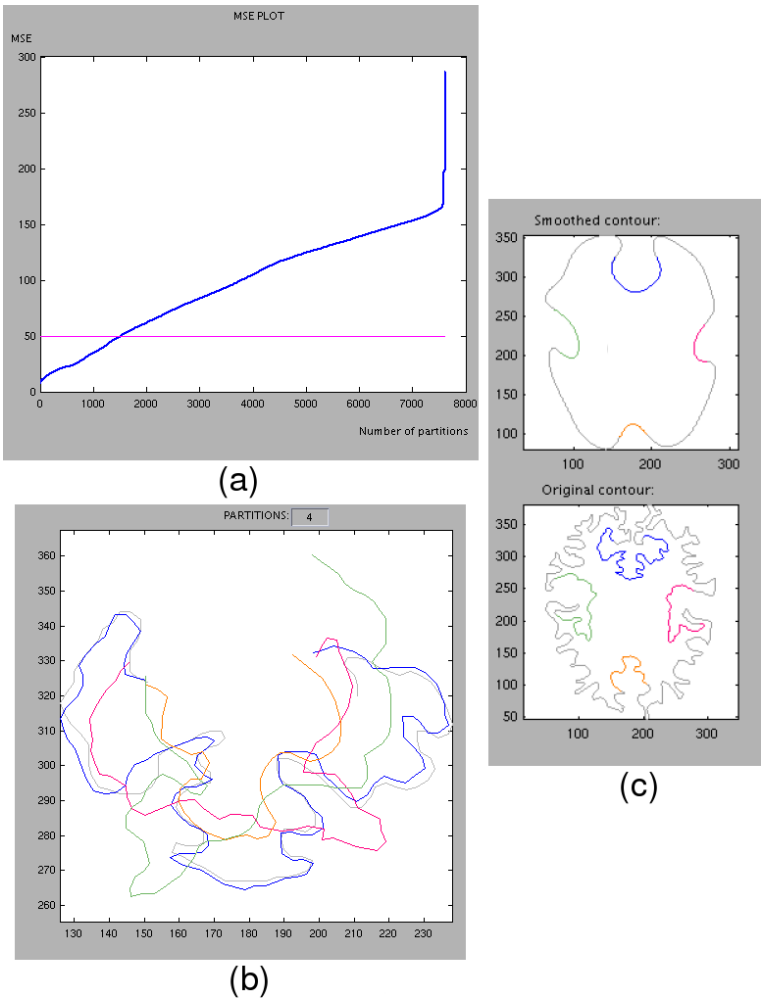


FIGURE 4.11: **(a)** Calculation of the MSE for each of the shapes results in plot used for ranking; **(b)** A manual threshold selects the number of shapes to be used in statistical shape analysis (pose alignment is already done); **(c)** Selected shapes from **(b)** along with their original positions in the smoothed and non-smoothed version of a given shape.

be used directly in a parametric, dimensionless way.

Due to the consistency of the CSS it is possible to overcome the problems presented in the MCMC approach. The first problem was related to the need for a fixed set of partitions, and with the CSS it is possible to obtain such a set based on the curvature zero-crossing and/or extrema points. The second problem was that the Fractal Dimension was not a stable enough criteria to find the best set of partitions. Here the CSS approach is a stable criteria as shown before to obtain the sets of shapes, due to its robustness with respect to noise or pose, and also because it

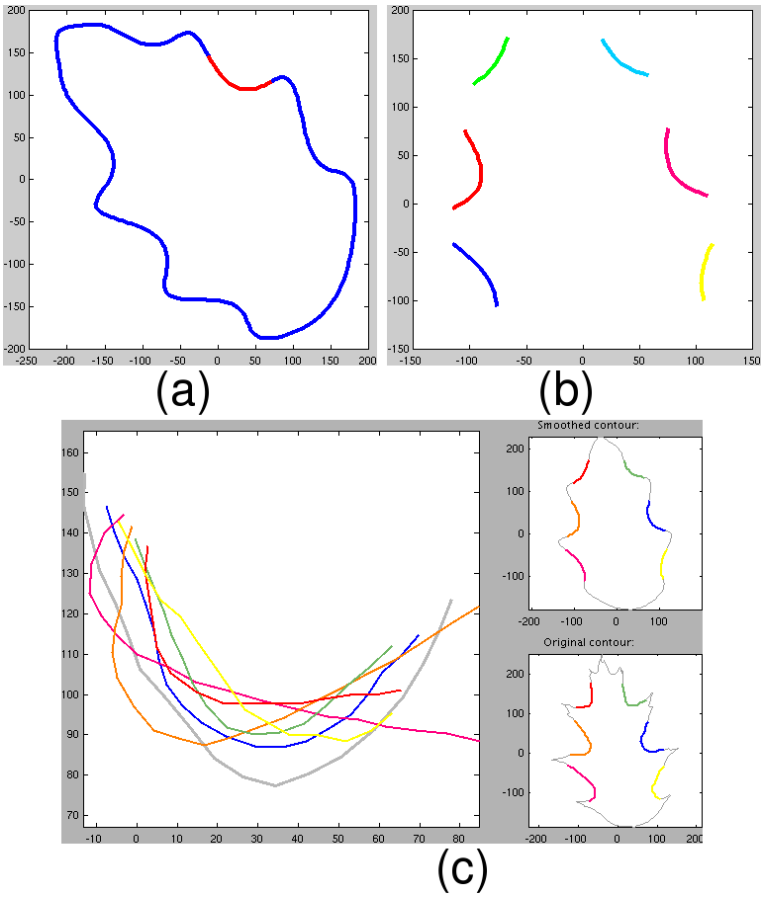


FIGURE 4.12: Example of the self similarity of the leaf partitions: **(a)** Reference partition in red **(b)** A set of similar partitions from the same contour and **(c)** Selected shapes from the MSE ranking along with their original positions in the smoothed and non-smoothed version of a given shape.

retains local information of the input shape (Figure 4.15). Hence, the contours are always partitioned in the same way, obtaining unique solutions.

Furthermore, the CSS could be improved by using it in conjunction with other shape parameters. This means that a more robust way of partitioning the contours can be produced by using for example the aforementioned fractal dimension measurement for the case of curves that present self similar structures, such as brain contours. The curvature sign at each point of the contour could be used to find concave or convex partitions, suitable to decompose curves that do not present a fractal behaviour like the leaf data set.

Nevertheless, there is a problem with the supervised method to rank the created set of partitions. The idea here as well, was to create a method to find a set of

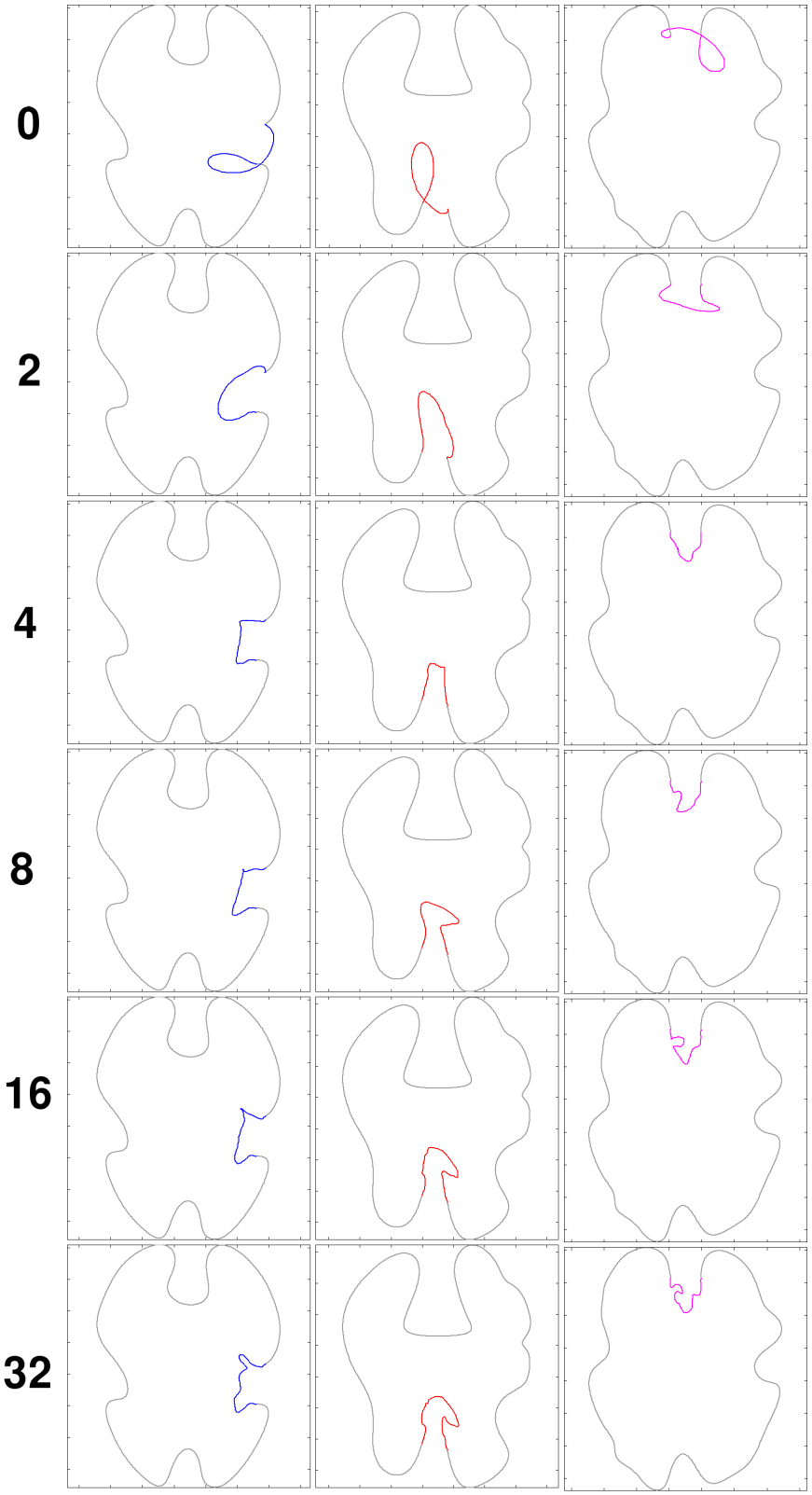


FIGURE 4.13: Reconstruction of the chosen set of white matter shapes, by adding a sequence of principal modes of variation: 0, 2, 4, 8, 16, 32. The modelled partitions are blended back into a smooth scale of the CSS, σ defocussing the general, irrelevant shape variations for the purposes of visualisation.

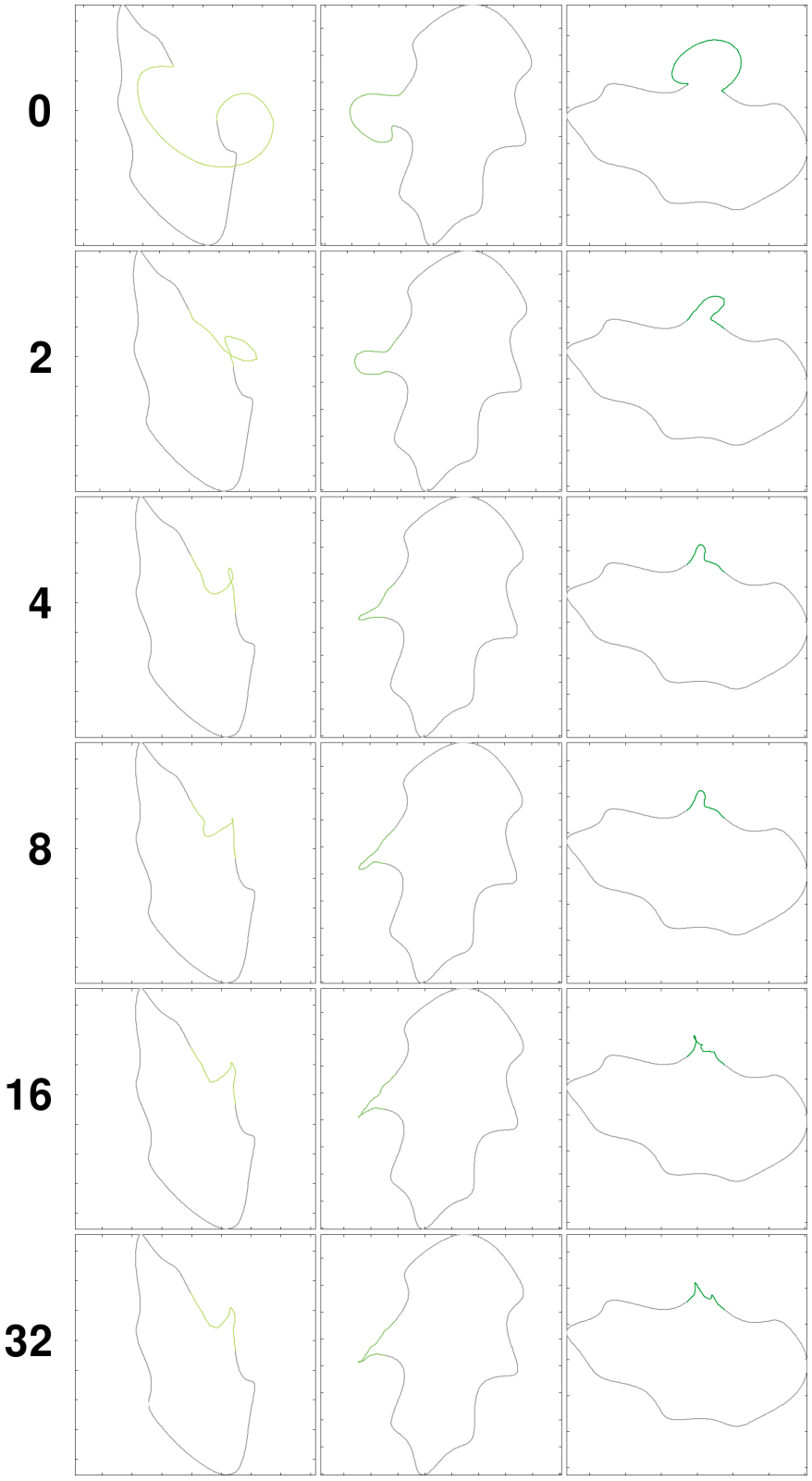


FIGURE 4.14: Reconstruction of the chosen set of leaf contours, by adding a sequence of principal modes of variation: 0, 2, 4, 8, 16, 32. The modelled partitions are blended back into a smooth scale of the CSS, σ defocussing the general, irrelevant shape variations for the purposes of visualisation.

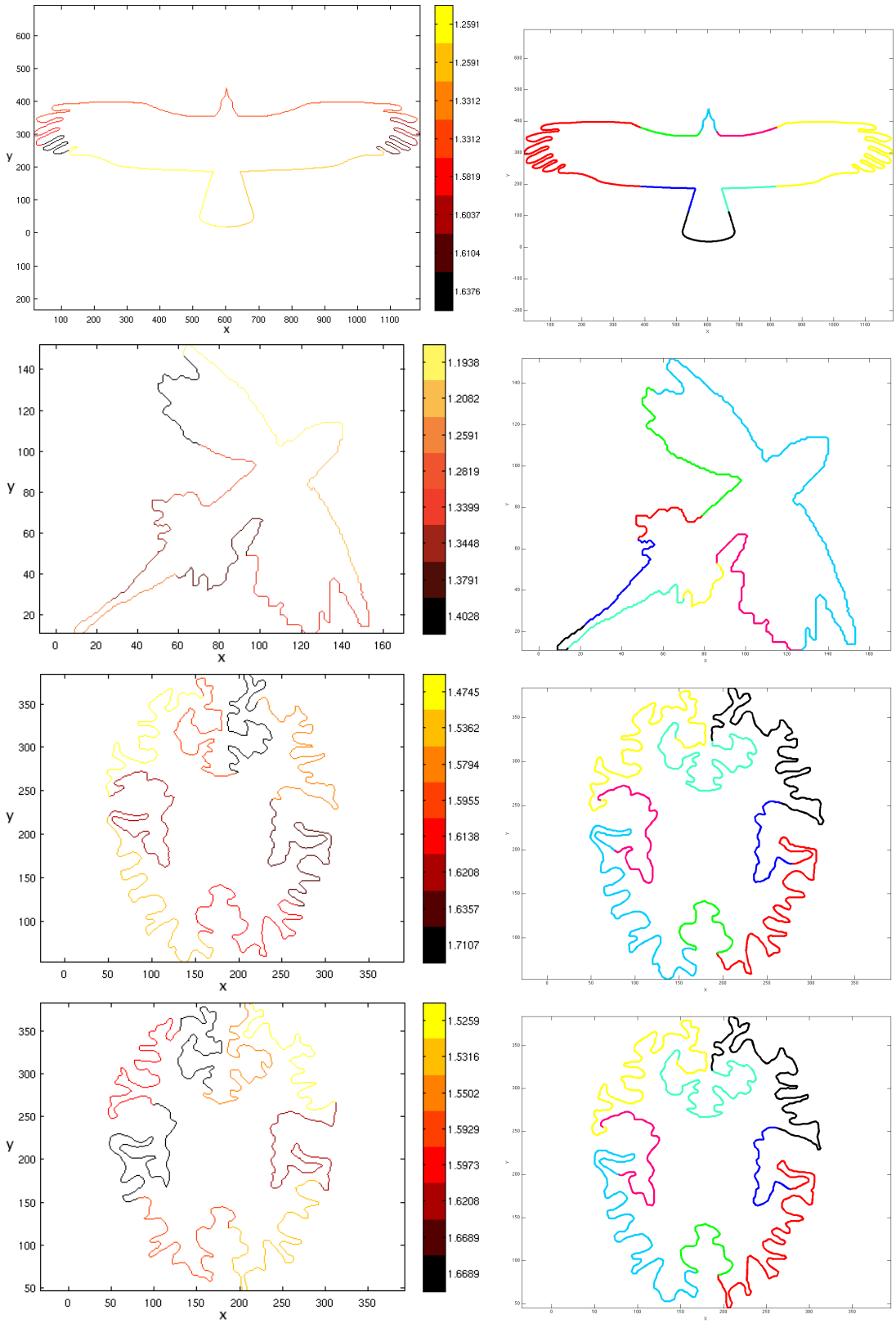


FIGURE 4.15: Column **(a)**: Results from the MCMC partitioning method and column **(b)** results from the CSS partitioning. By using CSS it is possible to overcome the problems presented in the MCMC approach, obtaining a unique set of partitions.

partitions ready for the shape analysis. The problem is that the ranking of shapes based on measuring the distances between the different shapes and the selected prototype is a limited method to create good sets of partitions. This is because this criteria is not flexible enough to create several sets for different types of shapes because the set is generated by the selection of a prototype shape. And there is a need for a supervised selection of this prototype shape and the use of a threshold to create shape data sets. Hence, a new solution needs to be explored. However, the novelty here was to introduce an initial approach to solve this problem. An unsupervised version of this method can be envisaged if the prototype selection and ranking step is replaced by a suitable clustering method. The resulting clusters could be used to build a set of shape-models, or, alternatively, a non-linear shape learning could be used as in [126].

4.7 Summary

This chapter has introduced an approach to the construction of a method for the creation of sets of partitions useful for local shape analysis based on the use of the CSS method. Here, a supervised method setting a threshold manually is used to create the sets. The mean square error of Euclidean distances for each partition of the set against a previously selected reference shape is used. Once a threshold is assigned, a set of partitions is created by letting those partitions with lower or equal distance to the set threshold to be part of the set. The methodology and details of the model have been presented with example results on white matter brain contours from MRI and leaf contours. As well, the problem of obtaining fixed sets of partitions has been addressed by using CSS, so it is possible to overcome the problems presented in the MCMC approach, obtaining a unique set of partitions.

A solution to overcome the problems mentioned regarding shape ranking can be addressed by unsupervised clustering using non-linear shape learning. This might also solve the problem of supervision in the creation of such sets. This is the main subject of the next chapter.

Chapter 5

Diffusion Maps for Local Shape Clustering and Analysis

5.1 Introduction

The study of variability in natural objects has been a topic of research for many years. Data derived from such research describes the characteristics and depicts the properties of such objects. Understanding the biological variability of anatomical objects is essential for scientific analysis and to distinguish between healthy and pathological structures. One of the most important activities for such analysis is to classify or group the gathered data into a set of categories or clusters [175]. In this chapter, a method that uses Diffusion Maps for non-linear, spectral clustering to build a set of linear shape spaces for local shape modelling and analysis is presented. The method uses a CSS description of shape to partition them into sets of self-similar parts (see last chapter for details) and these are then linearly mixed to more compactly model the global shape. Results on both synthetic (such as the Swiss roll) and natural data sets (leaves and white matter brain contours) are presented to assess the veracity of the method. The main contribution is the creation of a method that obtains a set of meaningful shapes according to a non-supervised spectral clustering technique, that is useful in local shape analysis.

5.2 Shape signatures

A *shape signature* can be defined as any 1D function that represents 2D areas or boundaries [181]. From the literature [120, 181, 182, 76] it has been reported that the most suitable shape signature is the Centroidal.

The *Centroidal distance* function is defined as the distance from the boundary points from the centroid of the shape [181]:

$$r(i) = \sqrt{(x_i - \overline{x})^2 + (y_i - \overline{y})^2} \tag{5.1}$$

where $\overline{x}$ and $\overline{y}$ denotes the coordinates of the centroid of the object.

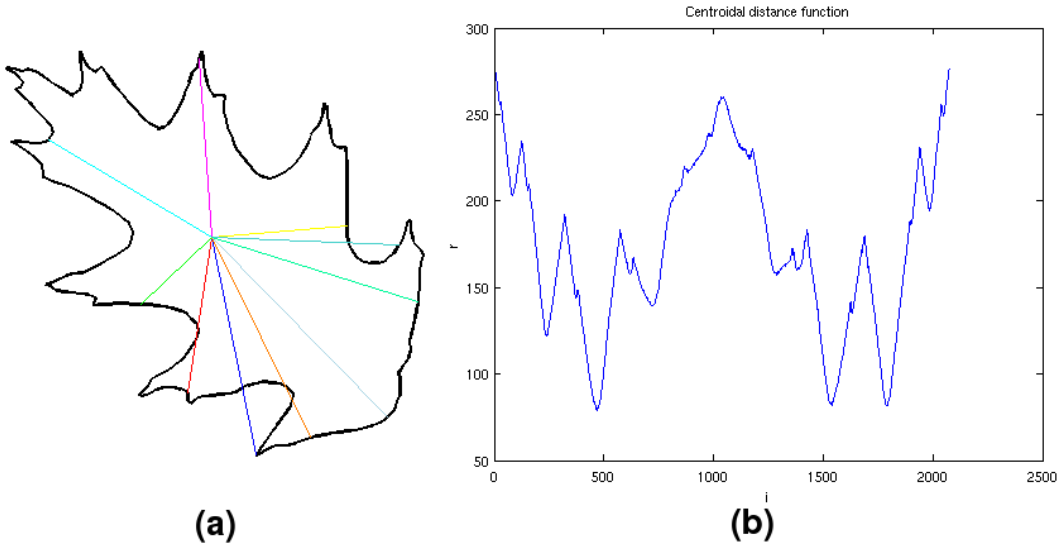


FIGURE 5.1: **(a)** An example of the centroidal distance function of the boundary and **(b)** its plot.

5.3 Fourier Descriptors for shape representation

Fourier transformation of shape signatures is a common first step in shape analysis, and methods based on Fourier descriptors are also used in other areas such as character recognition, shape coding, shape classification and shape retrieval [182].

5.3.1 Fourier Descriptors

In the context of shape analysis, the Fourier transformation of a boundary signature function generates a set of complex numbers, the *Fourier descriptors*. These descriptors represent the shape of an object in the frequency domain.

Given a closed curve $\mathcal{C}$ like in equation 2.16, we can derive a shape signature $r(m)$ where m represents the mth point on the boundary such that $1 \leq m \leq LB$, and LB is the length of the boundary. Since $r(m)$ is periodic it is possible to expand $r(m)$ into Fourier series as [93]:

$$r(m) = \sum_{n=-\infty}^{\infty} \mathbf{a}(n) e^{2j\pi nm/LB} \tag{5.2}$$

where the Fourier coefficients are given by:

$$\mathbf{a}(n) = \frac{1}{LB} \sum_{m=1}^{LB} r(m) e^{-2j\pi nm/LB} \tag{5.3}$$

The lower frequency descriptors contain information about the general shape, and the higher frequency descriptors contain information about the local details of the shape [76, 181]. Since the Fourier transform can only capture the structural features of a shape, it is crucial to derive FDs from a meaningful shape signature, in this case the Centroidal distance function.

5.4 Manifold Learning and Spectral Clustering

A *manifold* is a topological space that is locally Euclidean, meaning that around every point, there is a neighbourhood that is topologically the same as the open unit ball in $\mathbb{R}^n$ [134].

Manifold learning is a useful tool in data driven methods to interpret data when it lies on a low dimensional, non linear manifold. Manifold learning approaches seek to define either explicitly or implicitly a low-dimensional embedding of the

data points that preserves some properties of the high-dimensional point set [150]. The implied dimensionality reduction normally relies on classical techniques such as Principal Components Analysis and Independent Component Analysis. The problem with these techniques is that they try to represent data as linear combinations of a small number of basis vectors, and tend to fail because the data sets have an intrinsic dimensionality that is much less than the basis vectors required to linearly reconstruct them.

The generic problem of dimensionality reduction is the following. Given a set $\mathfrak{A} = (\mathfrak{a}_1, \ldots, \mathfrak{a}_z)$ of z points in $\mathbb{R}^k$, find a set of points $\mathfrak{B} = (\mathfrak{b}_1, \ldots, \mathfrak{b}_z)$ in $\mathbb{R}^j$ $(j \ll k)$ such that $\mathfrak{b}_i$ "represents" $\mathfrak{a}_i$, or in other words [13]:

$$\boldsymbol{\Psi} : \mathfrak{A} \in \mathbb{R}^k \rightarrow \mathfrak{B} \in \mathbb{R}^j \tag{5.4}$$

In this work the special case where $(\mathfrak{a}_1, \ldots, \mathfrak{a}_i)$ in $\mathcal{M}$ and $\mathcal{M}$ is a manifold embedded in $\mathbb{R}^k$ is considered.

From this definition, it can be observed that the global structure of the manifold can be more complicated or at least different than the lower dimensional domain. This implies that one needs to be aware of the geometry of the manifold and therefore the use of geodesic distances is needed [133, 66, 150]. The later idea is illustrated by figure 5.2.

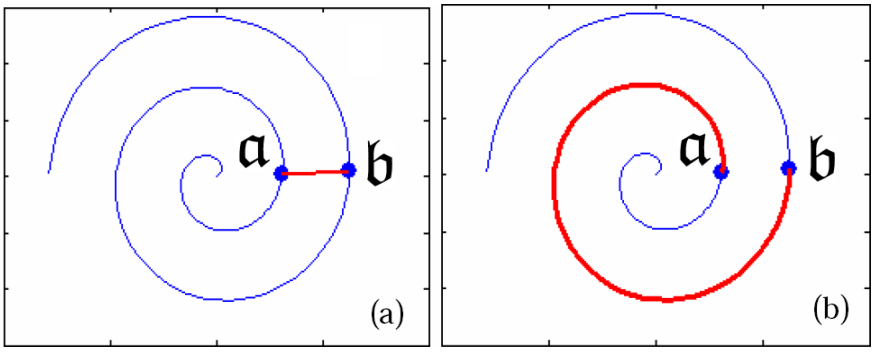


FIGURE 5.2: (a) Euclidean distance between data points $\mathfrak{a}$ and $\mathfrak{b}$. (b) Geodesic distance between data points $\mathfrak{a}$ and $\mathfrak{b}$.

Figure 5.2 is a simple example that shows the advantage of geodesic distances on manifolds over Euclidean distance. In this case, the data is sampled from a spiral, a one dimensional manifold. Figure 5.2 (a) shows the Euclidean distance between data points $\mathfrak{a}$ and $\mathfrak{b}$ and (b) shows the geodesic distance along the spiral. In this example we can see that the intrinsic geometrical structure can only be characterised by the geodesic distance.

The approach of dimensionality reduction considered in this work utilises maps provided by the eigenvectors of the graph Laplacian and eigenfunctions of the Laplace-Beltrami operator on the manifold [126]. Similarly, this solution can be interpreted in the framework of clustering and has very close relation to other spectral based clustering techniques. Here, the top eigenvectors of a matrix derived from the distances between points are used.

Manifold learning is a useful tool in data driven methods to interpret data when it lies on a low dimensional, non linear manifold. Manifold learning approaches seek to define either explicitly or implicitly a low-dimensional embedding of the data points that preserves some properties of the high-dimensional point setHence, given a set of n objects, it is possible to create a matrix of pair wise similarities between the objects and to formulate a general graph-theoretic framework for clustering as follows [13]. Let $\mathbf{G} = (V, E)$ be a weighted graph, with $\mathbf{W}$ as the matrix of weights where the vertices are numbered arbitrarily. Any weight $\mathbf{W}_{ij}$ associated with the edge $\mathbf{e}_{ij}$ is namely a similarity between v_i and v_j. If a clustering of two classes is considered, then the aim is to divide V into two disjoint subsets $\mathbf{A}$, $\mathbf{B}$ such that, $\mathbf{A} \cup \mathbf{B} = V$, so that the "flow" between $\mathbf{A}$ and $\mathbf{B}$ is minimised. This "flow" can be regarded as the measure of similarity between the two clusters and the simplest definition of the "flow" or "cut" between $\mathbf{A}$ and $\mathbf{B}$ is the total weight of the edges that have to be removed to make $\mathbf{A}$ and $\mathbf{B}$ disjoint:

$$cut(\mathbf{A}, \mathbf{B}) = \sum_{u \in \mathbf{A}, v \in \mathbf{B}} W(u, v) \tag{5.5}$$

Trying to minimise this $cut(\mathbf{A}, \mathbf{B})$ will favour cutting off weakly connected outliers that lead to poor partitioning quality. To avoid this problem a measure on the

set of vertices is introduced. The weight of a vertex is its "importance" relative to other vertices:

$$vol(\mathbf{A}) = \sum_{u \in \mathbf{A}, v \in V} \mathbf{W}(u, v) \tag{5.6}$$

where $\mathbf{W}(u, v)$ is the weight on the edge between u and v.

It so happens that the combinatorial optimisation problem as stated is NP-hard, but the problem can be reduced to minimising the Laplacian of the graph under certain conditions [13]. Graph Laplacians will be explored in greater depth in the next section, but first the general algorithm for the graph-based methods is presented.

Graph-based algorithms perform three steps [148]:

1. Build an undirected similarity graph $\mathbf{G} = (V, E)$.
2. Estimate local properties, i.e. define a weight matrix $\mathbf{W}$ to define the weighted similarity graph $\mathbf{G} = (V, E, \mathbf{W})$, where $\mathbf{w}_{ij} \geq 0$ represents the weight for the edge between vertex i and j. These weights are obtained by means of a kernel, a term defined later. A weight of 0 means that the vertices are not connected.
3. Define an optimal global embedding Ψ which preserves these local properties.

There are two often used variations to building the similarity graph $\mathbf{G}$, also known as an *adjacency graph*:

1. ϵ-neighbourhood graph which connects vertices i and j using:

$$||u_i - u_j||^2 < \epsilon \tag{5.7}$$

2. *k-nearest neighbour* (kNN) graph that connects vertices i and j by an edge if i is among the k-nearest neighbours of j or if j is among the k-nearest neighbours of i.

Finally, a *kernel* $\mathbf{k}$ is defined as $\mathbf{k} : \mathfrak{X} \times \mathfrak{X} \rightarrow \mathbb{R}^2$ on a data set $\mathfrak{X}$ as a function that defines edge weights for a matrix $\mathbf{W}$ in the weighted graph.

Once an adjacency graph is defined, manifold learning consists of mapping data points into a lower dimensional space while preserving the local properties of the adjacency graph (See equation 5.4). This dimensionality reduction can be achieved using spectral methods, which means analysing the eigen-structure of matrices derived from the adjacency matrix [48].

There are various techniques to achieve the dimensionality reduction such as Isomap [157], Locally Linear Embedding (LLE) [138], Hessian LLE [42], Diffusion Maps [27] or Laplacian Eigenmaps [13]. In the remaining part of the section Laplacian Eigenmaps will be discussed due to their strong relationship to Diffusion Maps, a technique that will be fully described in the next section.

5.4.1 Laplacian Eigenmaps

In general, Laplacian [1] Eigenmaps use the notion of the Laplacian of the graph to compute a low dimensional representation of the data set which preserves the local neighbourhood information in an optimal way. Then, a representation map is generated that can be viewed as a discrete approximation to a continuous map that arises from the geometry of the manifold [13].

Laplacian Eigenmaps are similar to LLE in that they try to preserve distance relations and that they can be solved by an eigenvalue problem. However, they differ in the fact that they additionally reflect the geometric structure of the manifold by approximating the Laplace-Beltrami operator using the weighted Laplacian of the similarity graph only when the manifold is uniformly sampled. The Laplace-Beltrami operator is an extension of normal Laplacians to manifolds [148].

Graph Laplacians are the central tool in spectral graph theory. The normalised and unnormalised versions, both based on the similarity graph, are built in the

[1] **Laplacian** or *Laplacian operator* (denoted by Δ) is a differential operator with many applications in physics and mathematics. For more information see [51].

first step of all the presented non-linear algorithms. Now, some of the variations are defined.

The unnormalised graph Laplacian is defined as:

$$\mathbf{L} = \mathbf{D} - \mathbf{W} \tag{5.8}$$

where is $\mathbf{W}$ defined using a kernel as described in the previous section, and the degree matrix $\mathbf{D}$ is diagonal such that:

$$d_{ii} = \sum_{j=1}^{n} \mathbf{w}_{ij} \text{ and } d_{ij} = 0 \ \forall i \neq j \tag{5.9}$$

Next, the Normalised graph Laplacian (random walk) is defined as:

$$\mathbf{L}_{rw} = \mathbf{D}^{-1}\mathbf{L} = I - \mathbf{D}^{-1}\mathbf{W} \tag{5.10}$$

Bearing in mind the previous equation, a mathematical formulation of Laplacian Eigenmaps for data in $\mathbb{R}^n$ and its extension to shape manifolds is presented as in [48]. Let $\mathcal{M}$ be a manifold of dimension m lying on $\mathbb{R}^n$ with $(m \ll n)$. Let $m = 1$, since generalisation to any dimension $(m < n)$ is immediate. Here, the dimensionality reduction problem consists of finding a mapping $f : \mathcal{M} \longrightarrow \mathbb{R}$, such that if two points $\mathfrak{x}$ and $\mathfrak{z}$ are *close* in $\mathcal{M}$, then so are $f(\mathfrak{x})$ and $f(\mathfrak{z})$. To characterise such mapping the following inequality is given [13]:

$$|f(\mathfrak{x}) - f(\mathfrak{z})| \leq d_{\mathcal{M}}(\mathfrak{x}, \mathfrak{z})||\nabla f(\mathfrak{x})|| + o(d_{\mathcal{M}}(\mathfrak{x}, \mathfrak{z})) \tag{5.11}$$

where $d_{\mathcal{M}}$ is the geodesic distance on the manifold $\mathcal{M}$. Then the optimality condition can be defined as:

$$f^* = \underset{||f(\mathfrak{x})||_{L^2(\mathcal{M})}=1}{argmin} \int_{\mathcal{M}} ||\nabla f||^2 \tag{5.12}$$

$$= \underset{||f(x)||_{L^2(\mathcal{M})}=1}{argmin} \int_{\mathcal{M}} \mathcal{L}(f) f \tag{5.13}$$

where $\mathcal{L} = -div(\nabla f)$ is the Laplace-Beltrami operator and div is the divergence of the vector field. It follows from Stokes' theorem that $-div$ and ∇ are formally adjoint operators, hence the equivalence of 5.12 and 5.13.

Minimising the objective function of equation 5.12 can be reduced to finding the eigenfunctions of the Laplace-Beltrami operator $\mathcal{L}$, i.e. the problem is equivalent to solving the eigen problem $\mathcal{L}(f) = \lambda f$. The optimal mapping is then given by the eigen functions corresponding to the smallest m non-zero eigenvalues. However in practice the discrete formulation must be used, that is given by the matrix in equation 5.8. Optimal dimensionality reduction is then accomplished by finding the eigenvectors of L, corresponding to the m smallest non-zero eigenvalues.

As in the previous section, an algorithm to compute the embedding $\mathbf{\Psi}$ in three steps is given [148]:

1. Build the undirected similarity graph $\mathbf{G} = (V, E)$.

2. Choose a weighted matrix $\mathbf{W}$ either by setting $\mathbf{w}_{ij} = 1$ for all connected vertices or using the heat kernel with parameter t:

$$\mathbf{w}_{ij} = e^{(-||z_i - z_j||^2/t)} \tag{5.14}$$

   If the graph is not fully connected, proceed with step 3 for each connected component.

3. Find the eigenvalues $0 = \lambda_1 \leq \ldots \leq \lambda_n$ and eigenvectors $\psi_1, \cdots, \psi_n$ of the generalised eigenvalue problem: $L\psi = \lambda D\psi$. Define the embedding: $\mathbf{\Psi} : z_i \rightarrow (\psi_2(i), \ldots, \psi_d(i))$.

The justification for the last algorithm comes from the role of the Laplace-Beltrami operator that provides an optimal embedding for the manifold. The manifold is approximated by the adjacency graph computed from the data points. Then, the Laplace Beltrami operator is approximated by the weighted Laplacian of the adjacency graph with the weights chosen in an appropriate way, in this case using

the heat kernel. The fundamental role of the Laplace-Beltrami operator in the heat equation makes it possible to use this kernel to choose the weight decay function in a principled manner. As a result, the embedding maps for the data approximate the Eigenmaps of the Laplace-Beltrami operator which are maps intrinsically defined on the entire manifold [13].

Laplacian Eigenmaps can be regarded as an special case of Diffusion Maps, as they only handle cases where the manifolds from which the data is uniformly sampled.

5.5 Diffusion Maps

A *Diffusion Map* is a kernel eigenmap method based on diffusion processes. It uses the eigenfunctions of Markov matrices that define a random walk on the data to obtain new data set descriptors via a family of mappings called "diffusion maps". These mappings embed the data points into Euclidean space, in which the usual distance describes the relationship between pairs of points in terms of their connectivity. This defines a useful distance between points in the data set termed "diffusion distance". Different geometric representations of the data set are then obtained by iterating the Markov matrix of transition. An equivalent method is to run the random walk forward. Here, diffusion maps are the tool that allows us to set up a relationship of the spectral properties of the diffusion process with the geometry of the data set. In particular, it is important to note that not only a single representation of the geometry is obtained, but a multi-scale family of geometric representations corresponding to descriptions at different scales is obtained [27].

It is worth mentioning that despite the fact that Diffusion Maps use the same kernel definition (defined in section 5.4) as other methods there is a difference. This is that a new diffusion operator P is used instead of the operator in equation 5.10 [148].

5.5.1 Construction of a random walk

The goal here is to define a distance metric on an arbitrary set that effectively reflects the point connectivity within the set. Given $V = \{v_1, \ldots, v_k\}$ and $\mathbf{W} = \{\mathbf{w}(u,v)\}_{u,v \in V}$, let $\mathbf{G} = (V, W)$ be a finite graph with n nodes. The graph $\mathbf{G}$ represents our knowledge about the local geometry of the set. Next the Markov walk on this graph is defined by introducing the degree $d(u)$ of node u as [27]:

$$d(u) = \sum_{w \in V} \mathbf{w}(u,w) \tag{5.15}$$

Now, if $\mathbf{P}$ is defined as a $n \times n$ matrix whose entries are given by:

$$\mathbf{P} = p(u,v) = \frac{\mathbf{w}(u,v)}{d(u)} \tag{5.16}$$

then $p(u,v)$ can be viewed as the transition kernel of a Markov chain on V.

As $\mathbf{P}$ contains geometric information about the data set, the transitions that it defines directly reflect the local geometry defined by the immediate neighbours of each node in the graph of the data. In other words, $p(u,v)$ represents the probability of transition from node u to node v in one time step [83]. For $t \geqslant 0$, the probability of transition from u to v in t time steps is given by $p_t(u,v)$, which is the kernel of the t-th power $\mathbf{P}^t$ of $\mathbf{P}$. One of the main ideas of the diffusion framework is that if the chain is run forward in time, or equivalently, powering $\mathbf{P}$, will allow the integration of the local geometry and therefore reveal the relevant geometric structures of V at different times. In other words, the higher the value of t, the further a probability weight can diffuse to other points further away. Under such a framework, a cluster is a region in which a probability of escaping this region is low.

The generated Markov chain is reversible, since it follows the detailed balance condition:

$$\varrho(u)p(u,v) = \varrho(u)p(v,u) \tag{5.17}$$

where $\varrho(u)$ represents the stationary distribution of the generated Markov chain. Then, equation 5.17 means that $\mathbf{P}$ has a discrete sequence of eigenvalues $\{\lambda_t\}_{t\geqslant 0}$ and eigenfunctions $\{\Psi_l\}_{l\geqslant 0}$ such that:

$$1 = \lambda_0 \geqslant |\lambda_1| \geqslant |\lambda_2| \geqslant \cdots \tag{5.18}$$

and

$$\mathbf{P}\Psi_l = \lambda_l \Psi_l \tag{5.19}$$

5.5.2 Diffusion Distances and Diffusion Maps

The relationship between the spectral properties of the Markov chain and the geometry of the data set can be described as follows. The idea of defining a random walk on the data set is based on the principle that the kernel k specifies the local geometry of the data and captures some geometric feature of interest. Then, the Markov chain defines fast and slow directions of propagation, based on the values taken by the kernel, and as the walk runs forward, the local geometry information propagates and accumulates in the same way local transitions of a system can be integrated in order to obtain a global characterisation of this system [27].

As previously mentioned, running the Markov chain forward is equivalent to computing powers of the operator $\mathbf{P}$. In theory, this can be computed using the eigenvalues and eigenvectors of $\mathbf{P}$ but instead, these objects can be directly employed in order to characterise the geometry of the data set.

A *family of diffusion distances* $\{D_t\}_{t\in\mathbb{N}}$ at time t is defined as:

$$D_t(u,v)^2 \triangleq ||p(w,t|u) - p(w,t|v)||^2_{\mathbf{w}} = \sum_w (p(w,t|u) - p(w,t|v))^2 \mathbf{w}(z) \tag{5.20}$$

where $p(w,t|u)$ is the probability that the random walk that started at u arrived at w in t steps.

For a fixed value of t, D_t defines a distance over the data set, and by definition, the notion of proximity that it defines reflects the connectivity in the graph of the

data. This means that $D_t(u, v)$ will be small if there is a large number of short paths connecting u and v, that is, if there is a large probability of transition from u to v and vice versa.

As mentioned before, $\lambda_1, \lambda_{2,\ldots,}$ tend to 0 and have a modulus strictly less than 1. As a consequence equation 5.20 can be computed to a present accuracy $\gamma > 0$. If the following term is defined:

$$\mathbf{s}(\gamma, t) = max\{l \in \mathbb{N} \text{ such that } |\lambda_l|^t > \gamma|\lambda_1|^t\}, \tag{5.21}$$

then, up to relative precision γ, the diffusion distance can then be approximated using the first nontrivial eigenvectors and eigenvalues according to:

$$D_t(u, v) = \left(\sum_{l=1}^{\mathbf{s}(\gamma,t)} \lambda_l^{2t} (\psi_l(u) - \psi_l(v))^2 \right)^{\frac{1}{2}} \tag{5.22}$$

Hence, its possible to define the family of diffusion maps $\{\Psi_t\}_{t\in\mathbb{N}}$ given by:

$$\Psi_t(u) \triangleq \begin{pmatrix} \lambda_1^t \psi_1(u) \\ \lambda_2^t \psi_2(u) \\ \vdots \\ \lambda_{\mathbf{s}(\gamma,t)}^t \psi_{\mathbf{s}(\gamma,t)} \end{pmatrix}. \tag{5.23}$$

Each component of $\Psi_t(u)$ is known as a *diffusion coordinate*. The mapping $\Psi_t : V \longrightarrow \mathbb{R}^{\mathbf{s}(\gamma,t)}$ embeds the data set into an Euclidean space of $\mathbf{s}(\gamma, t)$ dimensions.

The connection between diffusion maps and diffusion distances can be summarised as follows. The diffusion map Ψ_t embeds the data into the Euclidean space $\mathbb{R}^{\mathbf{s}(\gamma,t)}$ so that in this space, the Euclidean distance is equal to the diffusion distance (up to relative accuracy γ), or equivalently:

$$||\Psi_t(u) - \Psi_t(v)|| = D_t(u, v). \tag{5.24}$$

5.5.3 Algorithm

The algorithm to apply the diffusion maps to spectral clustering is as follows:

1. Construct a similarity graph.
2. Compute the normalised Laplacian.
3. Solve generalised eigenvector problem, $\mathbf{L}\psi = \lambda\mathbf{D}\psi$.
4. Define the embedding into a n-dimensional Euclidean space via diffusion maps.
5. Cluster points with K-means.

The implementation and results from this algorithm will be presented in a later section.

5.6 Evaluating the Clusters

As mentioned earlier, what clustering methods do is to find some desired number of groups, but what is really wanted is meaningful clusters that represent the true phenomena. So, the resulting groups must be analysed and then determine whether or not they aid in understanding the problem. In general, this can involve an assessment of the data to see if it contains any clusters but more often, it means an examination of the output of the algorithm to determine whether or not the clusters are meaningful.

Here, in order to make an objective comparison the adjusted Rand Index (aRI) performance measure is used. Such measure is based on counting pairs of points found in the same clusters in two clustering results and those found in different clusters in the two results. Having two partitions of the same data set called $\mathcal{Z}_1$ and $\mathcal{Z}_2$ with $\mathbf{z_1}$ groups and $\mathbf{z_2}$ groups, respectively. This can be represented in a $\mathbf{z_1} \times \mathbf{z_2}$ matching matrix $\mathbf{N}$ with elements n_{ij}, where n_{ij} is the number of

observations in group i of partition $\mathcal{Z}_1$ that are also in group j of partition $\mathcal{Z}_2$. It is interesting to note that the number of groups in each partition do not have to be equal, and that the classifications can be obtained through any method. Then the adjusted Rand Index can be calculated as:

$$RI_A = \frac{\mathbf{N}}{\mathfrak{D}} \tag{5.25}$$

where

$$\mathbf{N} = \sum_{i=1}^{\mathbf{z_1}} \sum_{j=1}^{\mathbf{z_2}} \binom{n_{ij}}{2} \; - \; \sum_{i=1}^{\mathbf{z_1}} \binom{n_{\,i\,\bullet}}{2} \sum_{j=1}^{\mathbf{z_2}} \binom{n_{\,\bullet\,j}}{2} \; \div \; \binom{n}{2}, \tag{5.26}$$

$$\mathfrak{D} = \left[\sum_{i=1}^{\mathbf{z_1}} \binom{n_{\,i\,\bullet}}{2} \; + \; \sum_{j=1}^{\mathbf{z_2}} \binom{n_{\,\bullet\,j}}{2} \right] \div 2 \; - \; \sum_{i=1}^{\mathbf{z_1}} \binom{n_{\,i\,\bullet}}{2} \sum_{j=1}^{\mathbf{z_2}} \binom{n_{\,\bullet\,j}}{2} \; \div \; \binom{n}{2}, \tag{5.27}$$

and

$$n_{\,\bullet\,j} = \sum_{i=1}^{\mathbf{z_1}} n_{ij} \;, \;\; n_{\,i\,\bullet} = \sum_{j=1}^{\mathbf{z_2}} n_{ij} \;. \tag{5.28}$$

For more details of the measure refer to [104].

5.7 Shape Clustering using Diffusion Maps

The shape clustering problem is of practical importance since it can significantly facilitate the automatic labelling of objects present in data collections, where labelling objects usually requires manual examination. Another challenge in shape clustering is the high dimensionality of the input space and the presence of noise or outliers, where performing clustering may be meaningless therefore a suitable dimensionality reduction is required.

It has been demonstrated that the manifold learning techniques can be applied in shape clustering. This is because it is possible to find a non-linear projection algorithm that can attract together similar shapes. This suggests the existence

of an isometry between the shape space and a low dimensional nonlinear embedding [177].

Since a shape representation invariant to translation, rotation, and scale transformations is desirable in the context of shape analysis, we have chosen Fourier descriptors (FD) as the shape representation since they are effective in many problems of pattern classification and computer vision [76]. As the main scope of this work is to analyse the unsupervised learning of shape manifolds, no claim of optimality is made about the shape representation using FDs. The following describes the procedure followed elsewhere ([118, 126]) that use FDs and Diffusion maps for spectral clustering.

Let us regard any shape as a contour (closed curve) represented as a set of boundary points represented as in equation 2.16. Then the Centroidal distance function is calculated using 5.1. Next, the distance vector $\mathbf{r} = \{r(1), r(2), \ldots, r(N)\}$ is transformed into the frequency domain using a DFT. Now the feature vector $\mathbf{f}$ is derived as follows:

$$\mathbf{f} = \left(\frac{|F_1|}{|F_0|}, \frac{|F_2|}{|F_0|}, \cdots, \frac{|F_{N/2}|}{|F_0|} \right). \tag{5.29}$$

Here $|F_i|$ denotes the *ith* Fourier coefficient and $|F_0|$ the *DC* component. In the last equation, due to the fact that the centroidal distance function is real valued, only half of the FDs are needed to index the shape. Taking the magnitudes of the coefficients yields rotation invariance, and scale invariance is obtained by dividing them by the DC component [181].

Like most manifold learning methods, the first step of diffusion maps is to define the feature vectors. $\Omega = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n\}$ (where n denotes the total number of shapes) can be regarded as the set of feature vectors which correspond to our data set of various shapes. Then, Ω become the nodes of an adjacency graph $\mathbf{G}$ such that $\mathbf{G} = (\Omega, \mathbf{W})$, where $\mathbf{W}$, is the similarity matrix between $\mathbf{f}_i$ and $\mathbf{f}_j$, and is computed using a Gaussian kernel of width σ:

$$\mathbf{w}(\mathbf{f}_i, \mathbf{f}_j) = e^{\frac{(-||\mathbf{f}_i - \mathbf{f}_j||^2)}{2\sigma}}. \tag{5.30}$$

The graph G with weights $\mathbf{W}$ represents our knowledge of the local geometry of the set. We next define a Markov random walk on this graph, by defining the degree $d(\mathbf{f}_i)$ of node $\mathbf{f}_i$ as:

$$d(\mathbf{f}_i) = \sum_{z \in \Omega} \mathbf{w}(\mathbf{f}_i, z). \tag{5.31}$$

Now, if $\mathbf{P}$ is defined as an $n \times n$ matrix whose entries are given by:

$$\mathbf{P} = p_{ij} = \frac{\mathbf{w}(\mathbf{f}_i, \mathbf{f}_j)}{d(\mathbf{f}_i)}, \tag{5.32}$$

then $\mathbf{P}$ can be viewed as the transition kernel of a Markov chain on V.

The definition of the Markov matrix for a given set of shapes belonging to different classes allows the embedding of the sub-manifolds corresponding to the shape classes in a lower dimensional space as follows. As a result of equation 5.17 we have 5.18 and 5.19, then a mapping from the shape feature Ω to a lower dimensional Euclidean space $\mathbb{R}^m$ is given by 5.23.

Finally, the last stage is to perform K-Means clustering over the obtained diffusion coordinates. In practice, the MATLAB implementation of the K-Means algorithm from the Multivariate Statistics toolbox was used.

5.8 Experimental Results

First results for common data sets to verify the accuracy of the method are presented. In this case we experiment with the helix, toroidal helix and the Swiss roll data sets, each with 800 points (See figures 5.3, 5.4, 5.5 respectively).

The next experiments used the same data set as in [126], which consists of six different shape classes from the Kimia database of silhouettes [141]. The classes are: carriage (20 shapes), dog (49 shapes), rat (20 shapes), fish(32 shapes), hand (16 shapes) and horse (20 shapes) for a total of 157 samples (Figures 5.6 and 5.7).

The following results correspond to the use of six different species of leaves 5.8 belonging to two genera, Acer (Maples) and Quercus (Oaks) from the Oregon

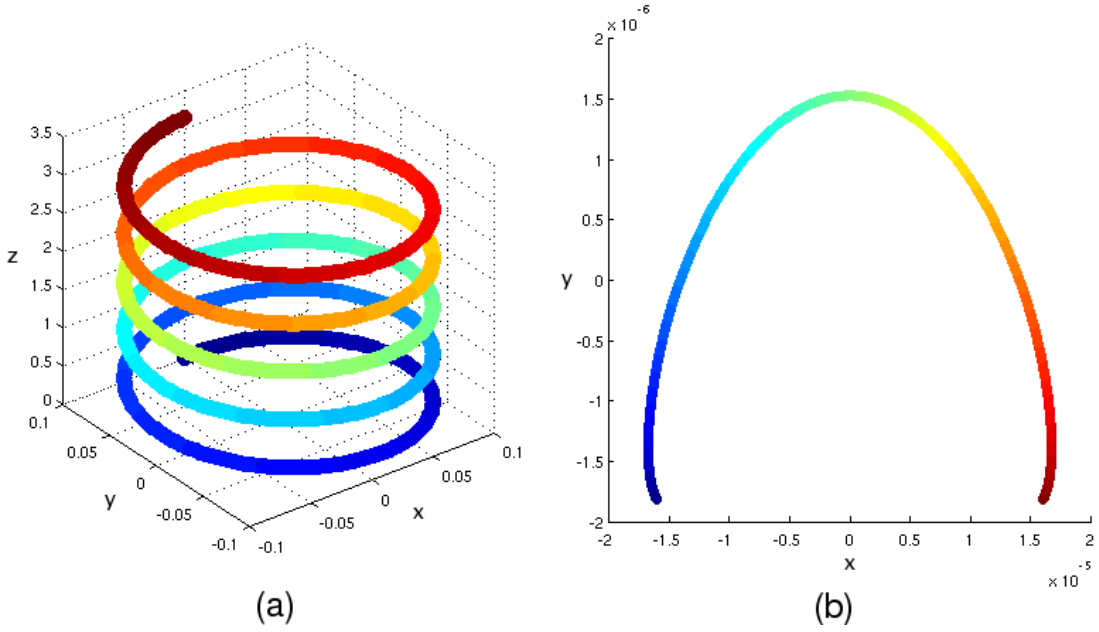


FIGURE 5.3: Helix data set: (a) Original set of 800 points, (b) Diffusion coordinates plot.

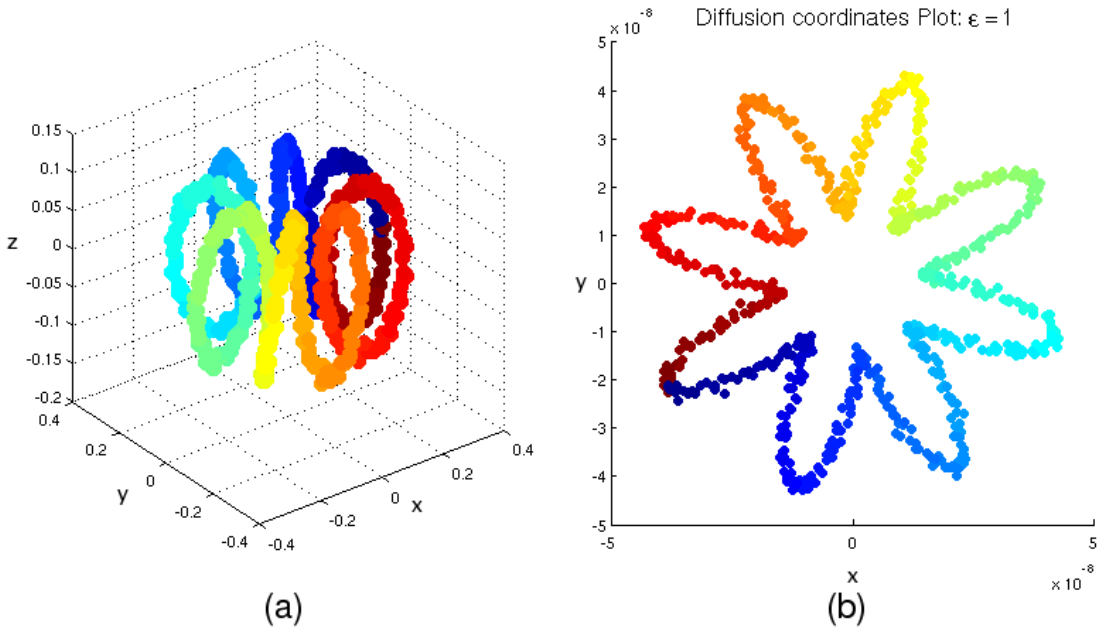


FIGURE 5.4: Toroidal helix data set: (a) Original set of 800 points, (b) Diffusion coordinates plot.

State University ([40]). The other data set is the same brain data used in the previous chapter [10].

A couple of experiments were performed in order to assess the veracity of the

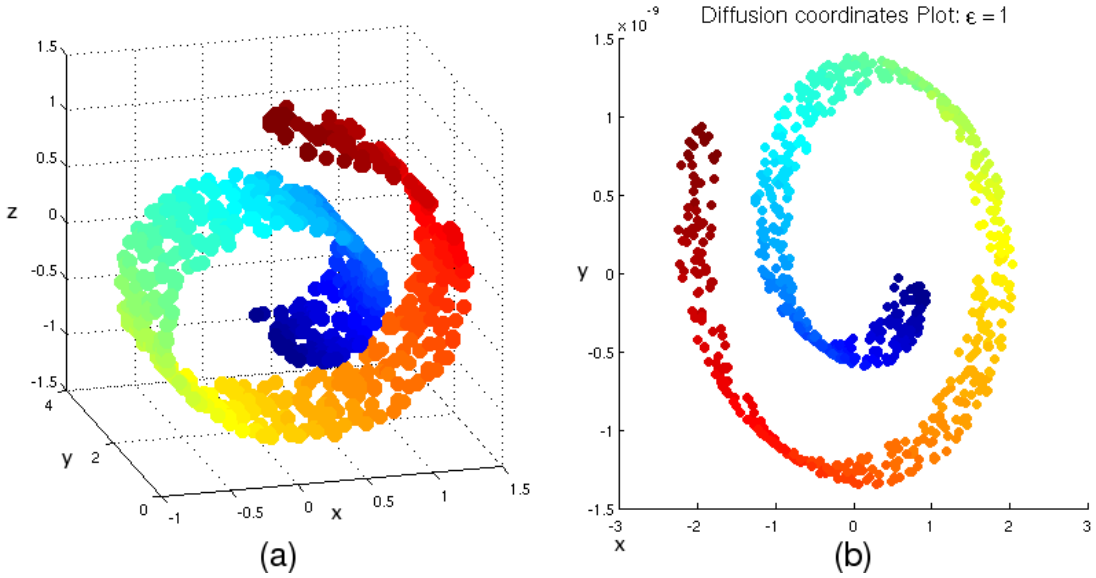


FIGURE 5.5: Swiss roll data set: (a) Original set of 800 points, (b) Diffusion coordinates plot.

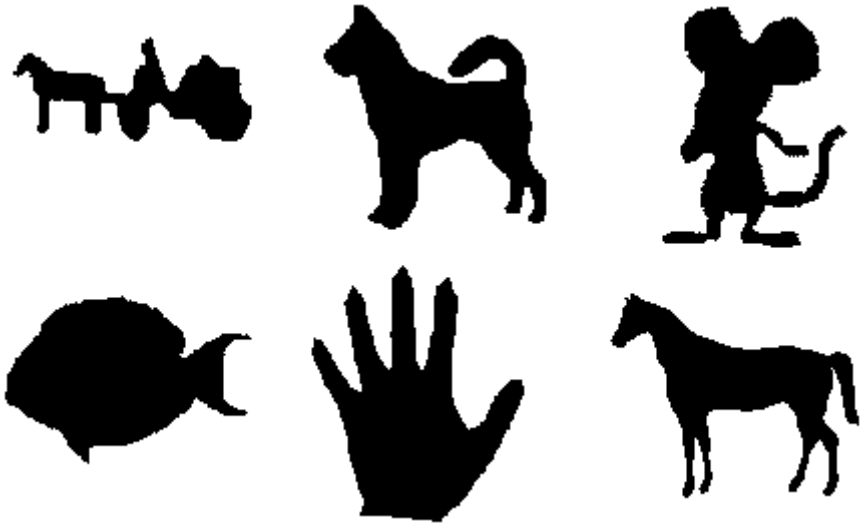


FIGURE 5.6: Types of shapes used for the experiments: carriage, dog, rat, fish, hand and horse.

method using non-closed contours. Two data sets where used: a single leaf data set of 50 contours each divided in 16 partitions for a total of 800 shapes; and a single white matter contour dataset of 100 contours each divided in 8 partitions for a total of 800 shapes.

The following results correspond to the single leaf data set where after obtaining the CSS zero-crossings, 4 different types of partitions can be clearly identified

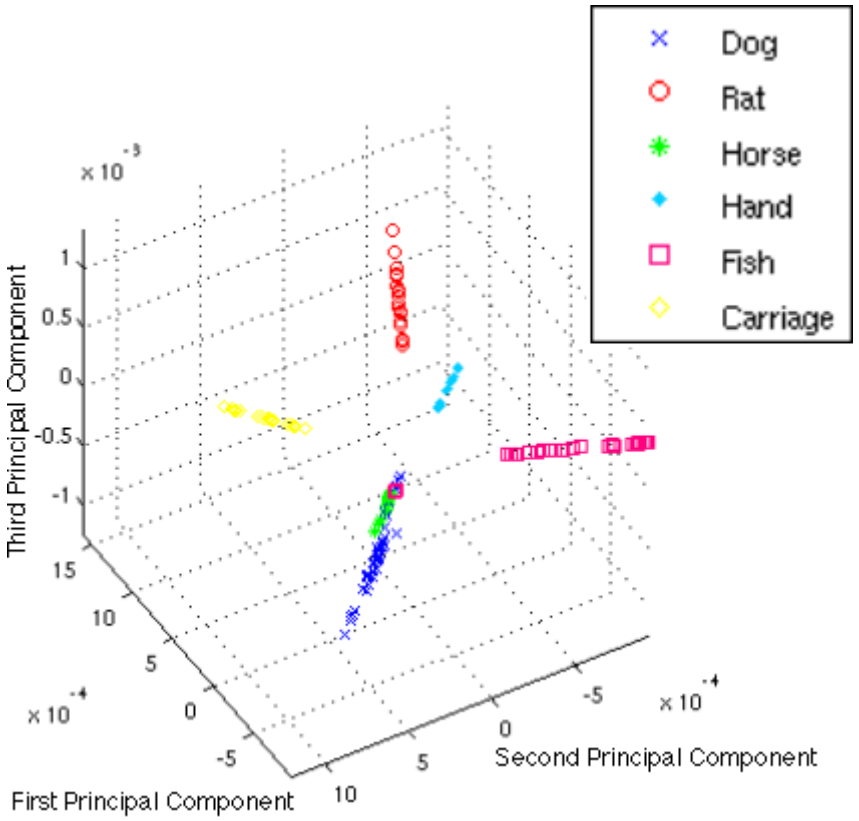


FIGURE 5.7: Results from the Diffusion coordinates and K-means to cluster the data from Kimia database. Classes (carriage, dog, rat, fish, hand and horse) can be distinguished easily.

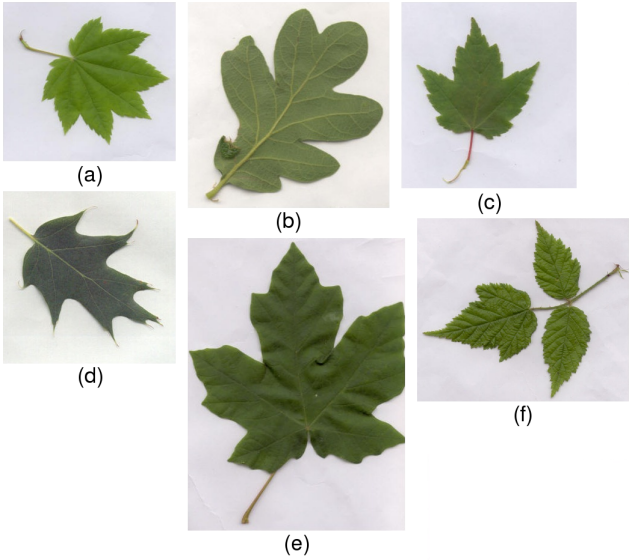


FIGURE 5.8: Sample plant species used for performance evaluation: **(a)** Acer Circinatum, **(b)** Quercus Garryana, **(c)** Acer Glabrum, **(d)** Quercus Kelloggii **(e)** Acer Macrophyllum and **(f)** Acer Negundo.

(figure 5.9-(a)). Then, the clusters are identified after the spectral clustering (figure 5.9-(b)) and the resulting cluster sets can be visualised in (figure 5.10) with the correspondent number of members in each class presented in table 5.1.

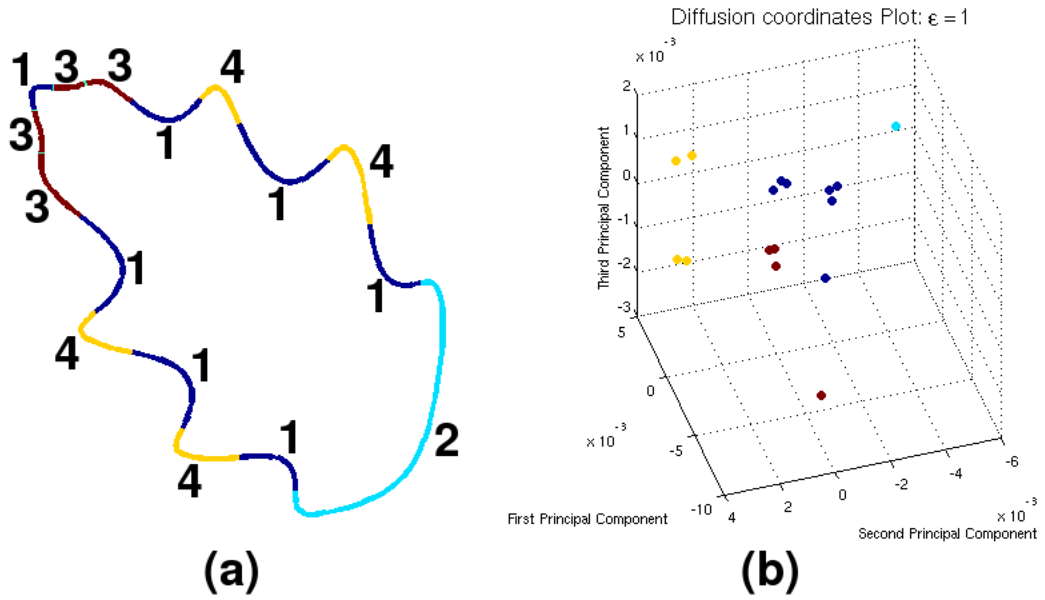


FIGURE 5.9: Single leaf contour data set: **(a)** Different types of partitions according to the CSS zero-crossings and **(b)** Diffusion coordinates plot illustrating the clusters of the data set.

Class 1	Class 2	Class 3	Class 4
350	50	200	200

TABLE 5.1: Results for the single leaf contour: 4 clusters

Similar results are presented for the single white matter contour data set. In figure 5.11-(a) 5 different types of partitions can be identified and in figure 5.11-(b) the resulting cluster sets can be visualised. The resulting cluster sets can also be seen in figure 5.12 with the corresponding number of members in each class given in table (5.2).

Class 1	Class 2	Class 3	Class 4	Class 5
100	200	200	200	100

TABLE 5.2: Results for the single white matter contour: 5 clusters

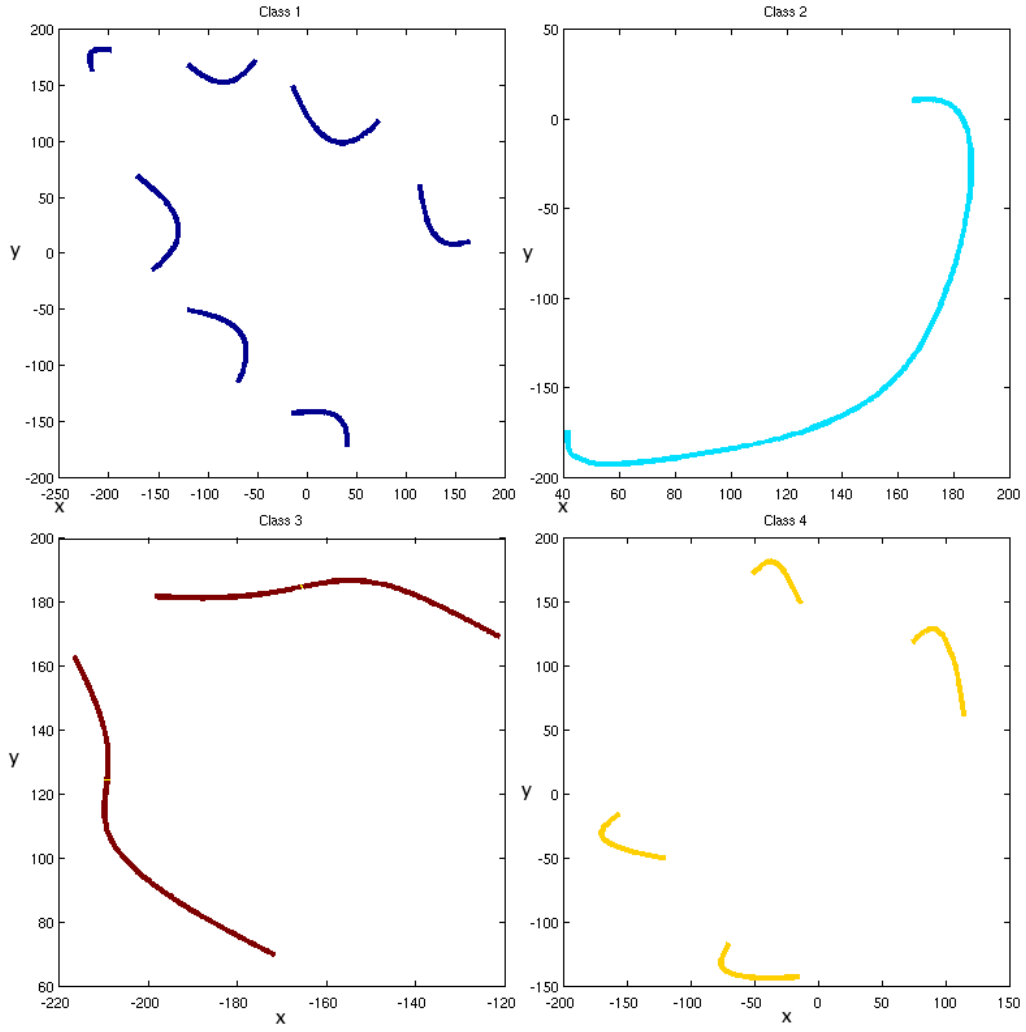


FIGURE 5.10: Different types of clusters according to the diffusion maps clustering over the single leaf contour set.

These results begin to demonstrate that it is possible to effectively cluster shapes and create accurate clusters. Next, results on multiple shape data sets are presented. The following were obtained using different leaf images from the class Quercus Kelloggii. Once again we created partitions out of each contour using CSS zero-crossings and then submitted the output to the Diffusion Maps.

In the following experiments, two different clustering methods were used: Laplacian Eigenmaps and Diffusion Maps. Therefore, to establish comparative results the aRI performance measure was employed, comparing the resulting labels from each method against the manual labels for each contour partition of the data sets. In this case, the correct value of σ is determined by the highest value on the aRI

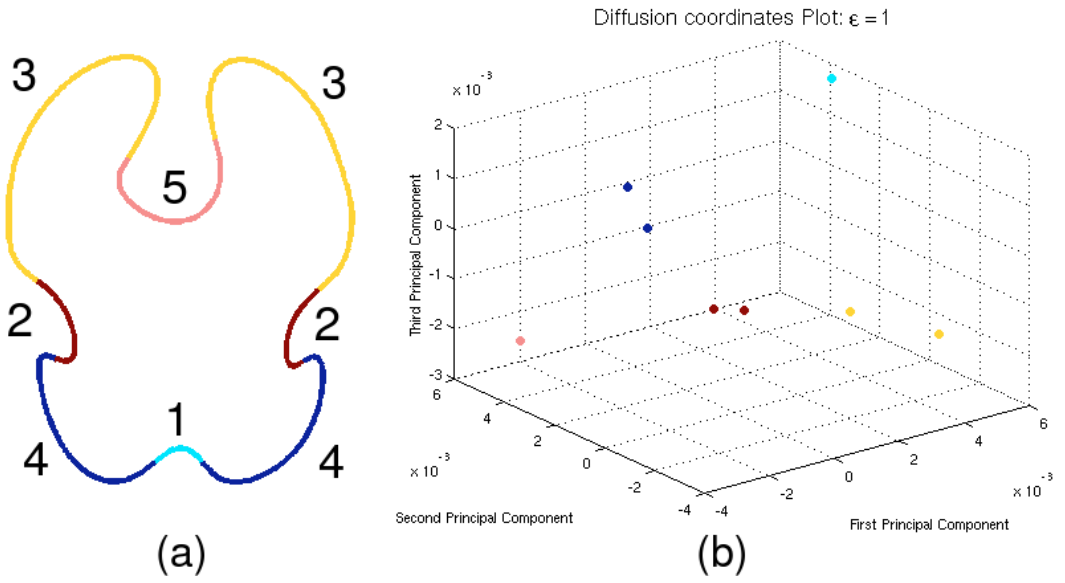


FIGURE 5.11: Single brain contour data set: (a) Different types of partitions according to the CSS zero-crossings and (b) Diffusion coordinates plot illustrating the clusters of the data set.

measure that should correspond to the highest value closest to one.

The first experiment was made using leaf contours of the Quercus Kelloggii and the total number of partitions obtained from 50 contours was 662. Comparative results between both clustering methods are presented in table 5.3 and from here, the value that produced maximum aRI value was $\sigma = 0.0003$ (in red) for the Diffusion Maps and $\sigma = 0.003$ (in magenta) for the Laplacian Eigenmaps.

After obtaining a suitable value of σ from the Diffusion Maps spectral clustering method (figure 5.13) it was possible to identify four different classes. Figure 5.14-**(a)** presents four different leaves with its partitions coloured according to the identified classes and in 5.14-**(b)** it is possible to observe the original hand labelling for comparison purposes.

The final experiment was maid using the brain contours data set that consisted of 501 partitions from 60 white matter contours. Once again as in the last experiment, to estimate the correct value of σ a ground truth test was used. Once more all the shapes were manually labelled and then results for different values of σ from both methods were compared against this using the aRI measure. Results are presented

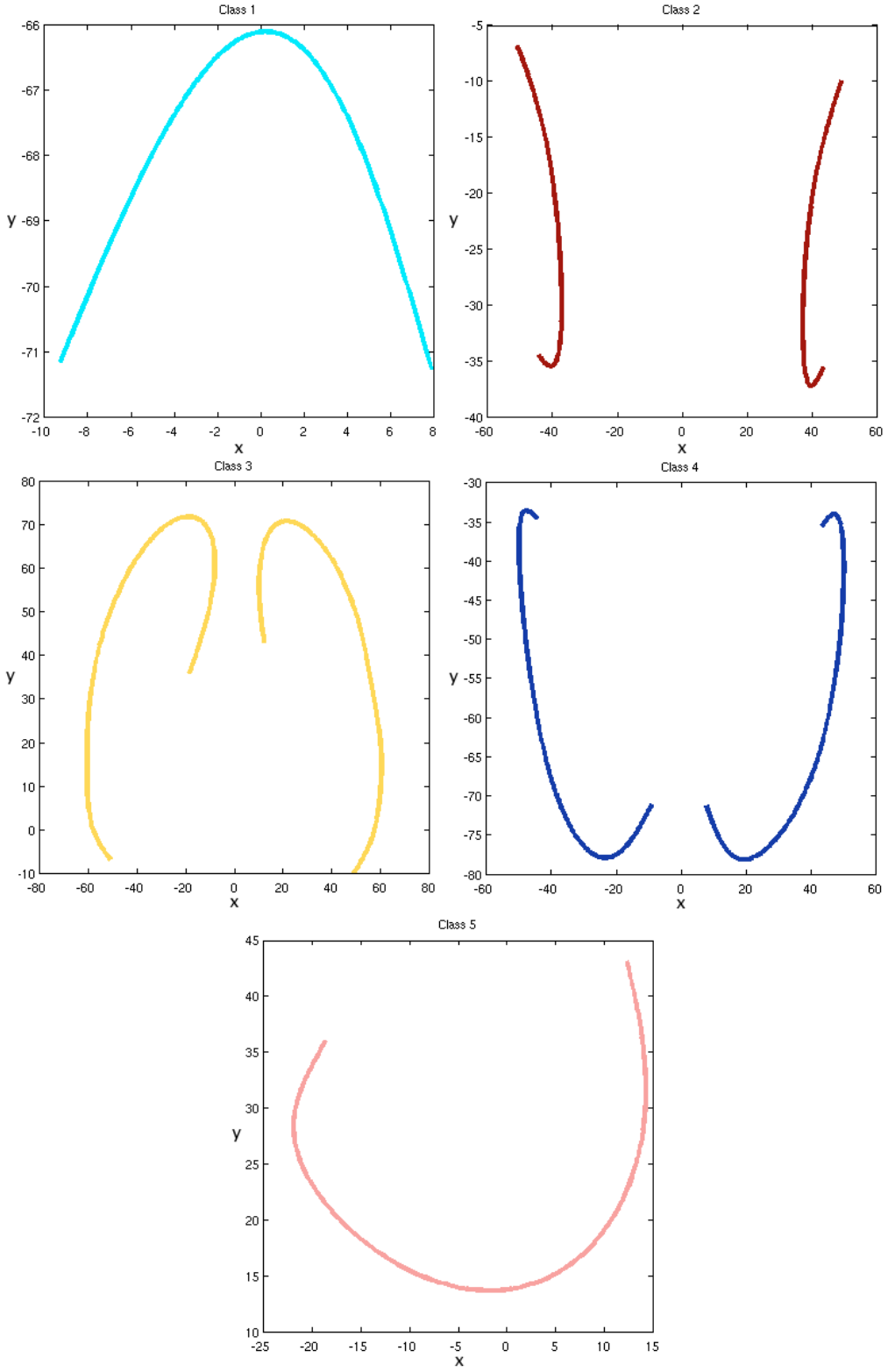


FIGURE 5.12: Different types of clusters according to the diffusion maps clustering over the single brain contour set.

	aRI	
σ	Dif. Maps	Lap. Eigenmaps
10	0.09693	0.03105
20	0.09802	0.02940
30	0.09830	0.03105
40	0.09802	0.03105
50	0.09802	0.03105
60	0.09802	0.03105
70	0.09802	0.03105
80	0.09802	0.03105
90	0.09802	0.01753
100	0.09802	0.03105
1	0.09906	0.03105
2	0.09830	0.03004
3	0.09802	0.03004
4	0.09802	0.01485
5	0.09830	0.01485
6	0.09802	0.03105
7	0.09802	0.03105
8	0.09830	0.03105
9	0.08603	0.03105
0.1	0.10290	0.03160
0.2	0.10001	0.01397
0.3	0.09990	0.03121
0.4	0.09924	0.03073
0.5	0.09924	0.03073
0.6	0.09919	0.03073
0.7	0.09919	0.03073
0.8	0.09906	0.03073
0.9	0.09906	0.03004

	aRI	
σ	Dif. Maps	Lap. Eigenmaps
0.01	0.16633	0.06337
0.02	0.15643	0.05494
0.03	0.15160	0.03252
0.04	0.14227	0.03525
0.05	0.11069	0.03784
0.06	0.11069	0.03646
0.07	0.10101	0.03227
0.08	0.09833	0.03129
0.09	0.10020	0.03017
0.001	0.16633	0.09436
0.002	0.15643	0.08735
0.003	0.15160	**0.19513**
0.004	0.14227	0.06461
0.005	0.11069	0.06582
0.006	0.11069	0.06362
0.007	0.10101	0.06143
0.008	0.09833	0.04828
0.009	0.10020	0.05393
0.0001	0.21115	0.0303917
0.0002	0.23502	0.0514889
0.0003	**0.27156**	0.0709292
0.0004	0.13644	0.09425
0.0005	0.20549	0.07110
0.0006	0.18056	0.06119
0.0007	0.13758	0.05242
0.0008	0.16751	0.03893
0.0009	0.13758	0.07748

TABLE 5.3: Comparison between Laplacian Eigenmaps and Diffusion Maps using the aRI measure for the leaf contours data set. The biggest value of σ (in red) is determined by the highest value on the aRI measure that should correspond to the highest value closest to one, in this case corresponds to 0.0003 from the DM method.

in table 5.4 and from here, the value that produced a maximum aRI value was $\sigma = 0.003$ again from the Diffusion Maps method.

Figure 5.15 show the diffusion coordinates plot for four classes of partitions. Finally, figure 5.16 depicts the identified classes.

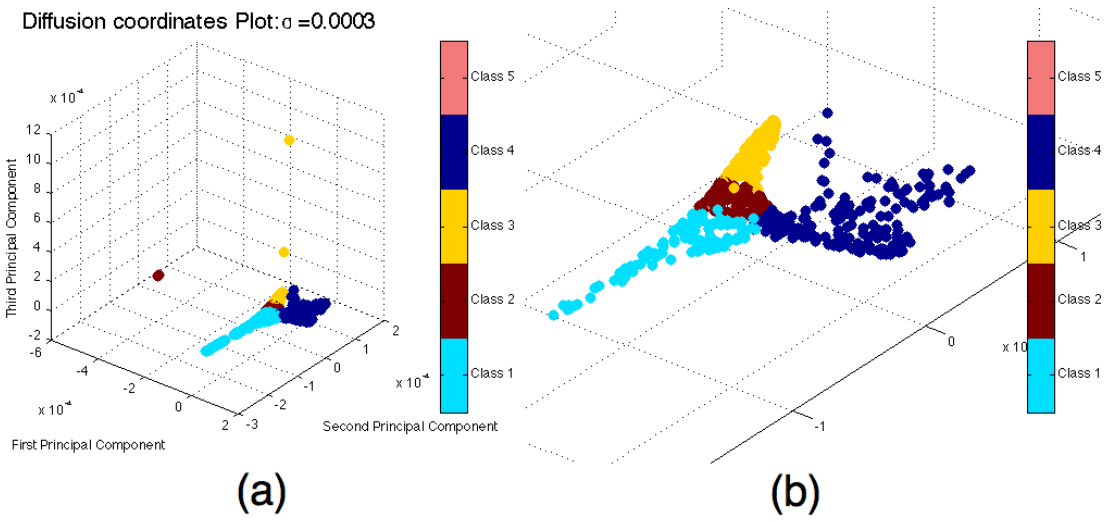


FIGURE 5.13: **(a)** Diffusion coordinates plot identifying four different clusters from 50 leaf contours and **(b)** zoom on the plot.

5.9 Evaluation and Discussion

The main contribution of this chapter is the creation of a method that obtains a set of meaningful shapes, so that it is possible to find local parts that are similar and localised according to a non-supervised clustering technique. Furthermore, another objective of this was to generate ordered sets of partitions from contours to establish a way of determining meaningful local sets of shapes. Additionally, another contribution of this work is that it helps to clarify a way to solve the problem of the choice of the prototypical shape used for the shape analysis (See [16] as well). Here, the sets of generated clusters are created with similar shapes, and potentially, any shape of the resulting set could be used as the prototypical shape.

The first four experiments (figures 5.3 to 5.6) were made with the intention of analysing the performance of the Diffusion Maps method under commonly used data sets. Results were satisfactory using such data as can be seen in [173] and [126]. This shows that the spectral clustering is effective in discovering the non-linear manifold of the shape space. A major concern in obtaining the results was that in most reported experiments in the literature closed contours were used. Since in this work non-closed partitions of whole contours derived from the CSS are used, the loss of fine detail representation caused by the use of half of the

	aRI	
σ	Dif. Maps	Lap. Eigenmaps
10	0.12047	0.10181
20	0.09544	0.04505
30	0.11265	0.07244
40	0.11275	0.10181
50	0.12304	0.10181
60	0.11275	0.10181
70	0.09544	0.10798
80	0.10754	0.10181
90	0.12501	0.10181
100	0.09544	0.10403
1	0.11422	0.106555
2	0.13091	0.09991
3	0.09874	0.10798
4	0.13091	0.10798
5	0.12505	0.10181
6	0.12527	0.04319
7	0.09381	0.10181
8	0.10843	0.10181
9	0.09381	0.10798
0.1	0.14852	0.10734
0.2	0.10280	0.09767
0.3	0.11831	0.10655
0.4	0.11463	0.10396
0.5	0.12293	0.10094
0.6	0.10543	0.10094
0.7	0.11234	0.10094
0.8	0.12237	0.10621
0.9	0.10504	-0.00603

	aRI	
σ	Dif. Maps	Lap. Eigenmaps
0.01	0.24233	**0.25299**
0.02	0.27515	0.07412
0.03	0.28550	0.10808
0.04	0.22788	0.11289
0.05	0.25438	0.11655
0.06	0.18952	0.11547
0.07	0.17658	0.11522
0.08	0.16773	0.11259
0.09	0.16177	0.11199
0.001	0.18014	0.01655
0.002	0.20778	0.03244
0.003	0.19963	0.04521
0.004	0.22100	0.05523
0.005	0.22695	0.10211
0.006	0.23650	0.15634
0.007	0.24648	0.22088
0.008	0.25746	0.03862
0.009	0.22385	0.24533
0.0001	0.22231	0.01710
0.0002	0.36243	0.03936
0.0003	**0.39919**	0.02555
0.0004	0.31535	0.04190
0.0005	0.28894	0.05198
0.0006	0.26114	0.06640
0.0007	0.12246	0.06060
0.0008	0.13315	0.07650
0.0009	0.14684	0.01335

TABLE 5.4: Comparison between Laplacian Eigenmaps and Diffusion Maps using the aRI measure for the white matter contours data set. The biggest value of σ (in red) is determined by the highest value on the aRI measure that should correspond to the highest value closer to one, in this case corresponds to 0.0003 from the DM method.

Fourier descriptors (see equation 5.29) does not represent a problem due to the "smoothing" nature of the CSS evolutions. To the knowledge of the author, this is the first attempt in clustering non-closed contours using spectral clustering techniques.

The first experiments on both data sets, leaf and brain white matter contours, reflect that given the same shape repeated several times the method is effective in

finding accurate clusters as it can be seen in tables 5.1 and 5.2, were it is possible to find that each cluster has the right amount of shapes. For example, taking the first experiment with the leaves, from figure 5.9 we can observe four different labeled classes, so as the same contour was repeated 50 times, it was expected for instance, to find 350 shapes for class 1 since there are 7 partitions of this type in each repeated contour. Variation in the data sets for the following experiments (see figures 5.14 and 5.16) is reflected in the low values of the aRI measure for both methods, since for those experiments the shapes for both data sets were all different, making more difficult the correct classification of partitions. Even though results were not high according to the aRI measure (see tables 5.3 and 5.4) from these experiments it is possible to conclude that the Diffusion Maps technique is better than Laplacian Eigenmaps. The combination of CSS and diffusion maps is a way to map self-similar contours to a piece-wise shape description.

Finally, in order to demonstrate the compactness of the proposed LSM, spectral-clustering was used to build a set of four and five local (linear) shape models respectively, and then comparing each model against a single (global) one. Figures 5.17 and 5.18 show the cluster variation for the local versus global models and demonstrates the better compactness of the four classes over the global model for the Kelloggii leaves and the Brain white matter data sets.

Some alternatives to improve this work can be considered. First, Fourier Descriptors can be replaced by other techniques to extract shape features. An option is to use shape context descriptors [14] that offers a globally discriminative characterisation by capturing the distribution of remaining points in an image relative to a reference point. Diffusion maps are sufficient in finding suitable clusters for the shape analysis, nevertheless, experimentation with other spectral clustering techniques such as Isomap [157] and other spatial clustering methods such as Gaussian Mixture Models [106, 175] instead of K-Means, could be useful. This is because of the elongated shape of the clusters (see figures 5.13-(b) and 5.15-(b)), where the mentioned techniques can lead to better results.

5.10 Summary

In this chapter a method that obtains a set of meaningful shapes using a non-supervised clustering method has been introduced and explained.

The chapter began with an explanation of Shape signatures and Fourier Descriptors, and then described Manifold Learning and Spectral Clustering. Manifold learning is a useful tool in data driven methods to interpret data when it lies on a low dimensional, non linear manifold. Manifold learning approaches seek to define either explicitly or implicitly a low-dimensional embedding of the data points that preserves some properties of the high-dimensional point set. They are a useful tool in data driven methods to interpret data when it lies on a low dimensional, non linear manifold and seek to define either explicitly or implicitly a low-dimensional embedding of the data points that preserves some properties of the high-dimensional point set. The subsequent section explained Laplacian Eigenmaps technique since it is closely related to Diffusion Maps, the main matter of this chapter. This technique relies on the basic assumption that the data lies in a low dimensional manifold in a high dimensional space. Diffusion Maps were then explained in the following section. Such methods use the eigenfunctions of Markov matrices that define a random walk on the data to obtain new data set descriptors via a family of mappings called "diffusion maps". After this, an explanation of how to evaluate the clusters was given and the aRI measure of compactness was described.

Experiments that were carried out with different sets of shapes are described. First experiments with synthetic data (Swiss roll and Kimia data sets for example) and then with natural data (leaves and brain white matter contours). A discussion of the results was given along with a demonstration of the compactness of the proposed LSM by building a set of local (linear) shape models and then comparing each model against a single global one. Finally, possible ways to improve the obtained results was discussed.

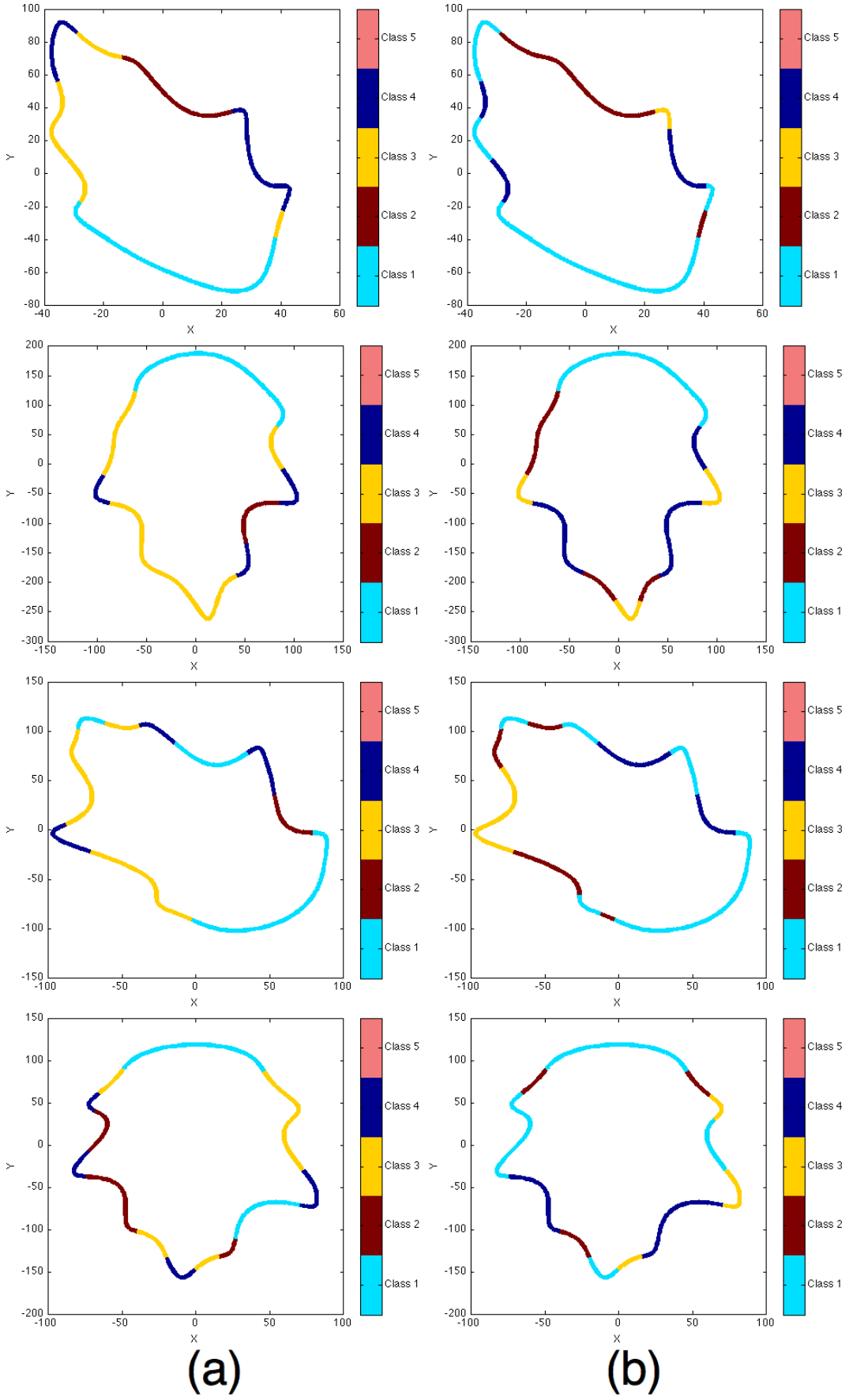


FIGURE 5.14: Column **(a)** Four different leaf contours with its partitions coloured according to the class identified by the Diffusion Maps clustering and column **(b)** contours with the manual labels.

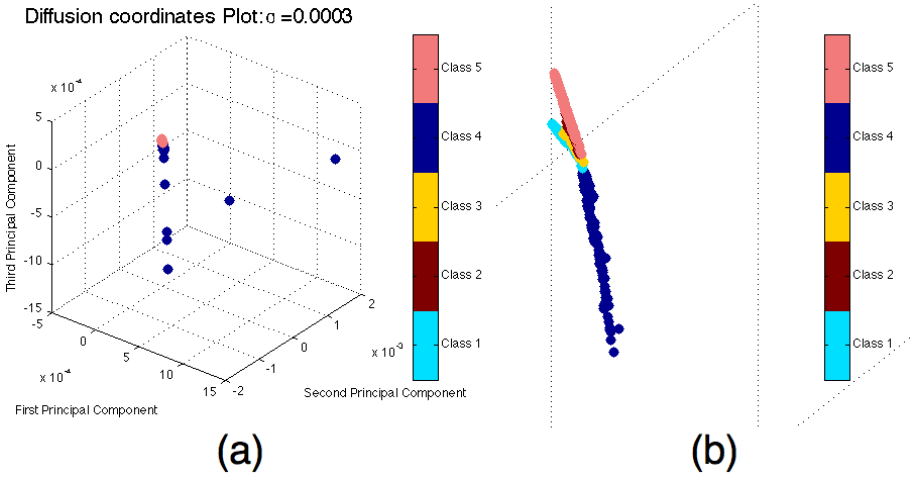


FIGURE 5.15: **(a)** Diffusion coordinates plot identifying four different clusters partitions from 60 white matter contours and **(b)** zoom on the plot.

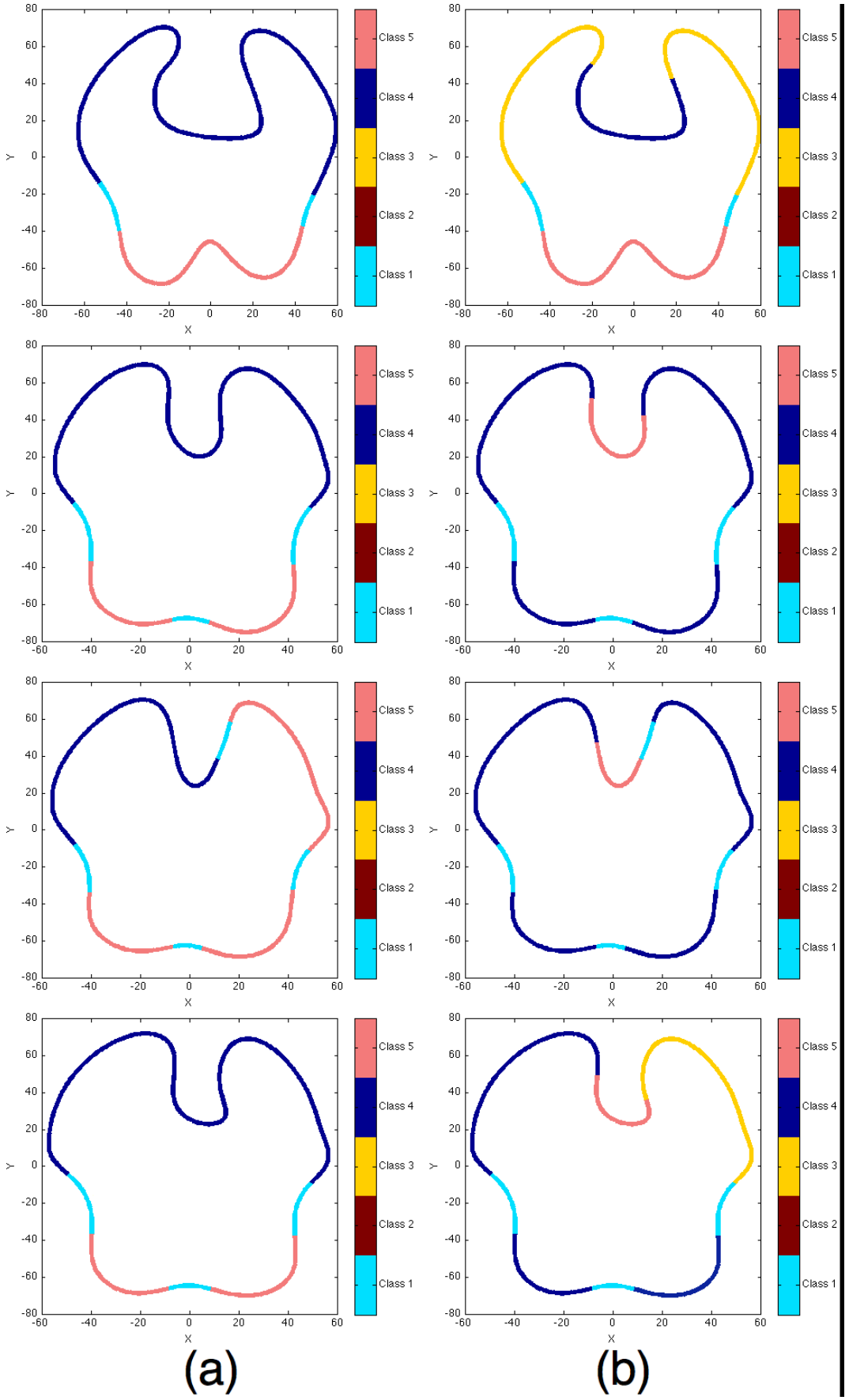


FIGURE 5.16: Column **(a)** Four different brain white matter contours with its partitions coloured according to the class identified by the Diffusion Maps clustering and column **(b)** contours with the manual labels.

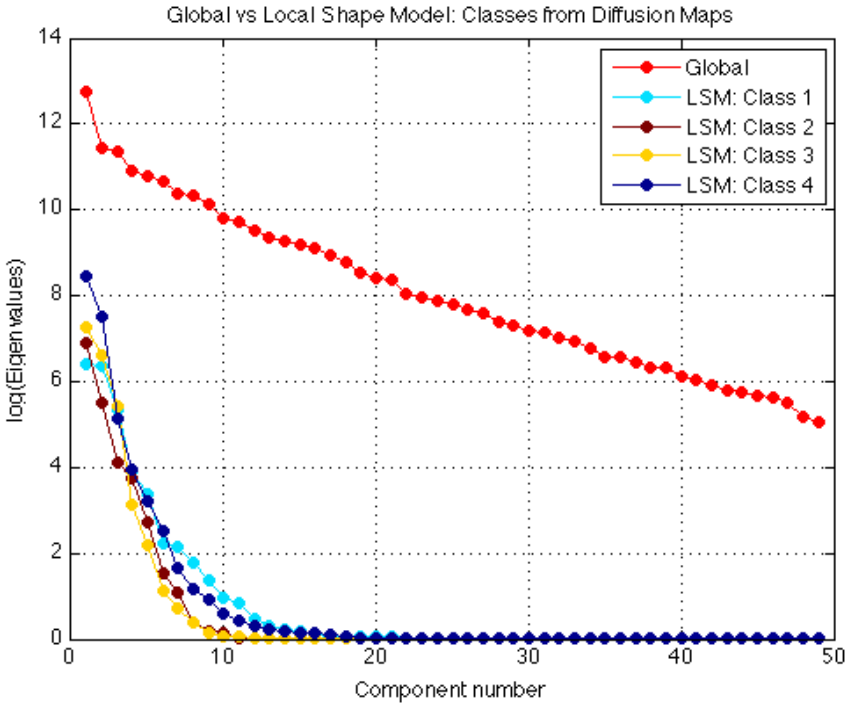


FIGURE 5.17: Kelloggii leaves set: Log plot of eigenvalues for the different clusters and for the global shape model in red.

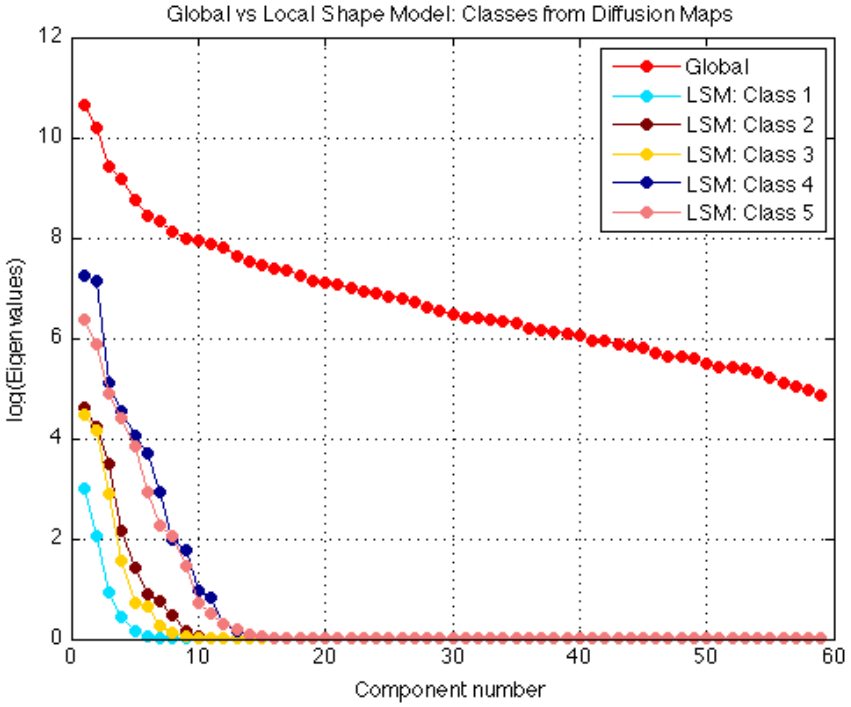


FIGURE 5.18: Brain white matter set: Log plot of eigenvalues for the different clusters and for the global shape model in red.

Chapter 6

Hierarchical Contour Shape Analysis

6.1 Introduction

Having described and developed in previous chapters the proposed LSM, to asses its validity it is desirable to compare its performance with some other method. This chapter introduced a novel shape representation which performs shape analysis in a hierarchical fashion using Gaussian and Laplacian pyramids. It then uses the it to compare the proposed LSM model. A background on hierarchical shape analysis is given along with a detailed explanation of the hierarchical method, and as well results are shown on natural contours. Then, the comparison between this method and the proposed LSM is performed using different shape sets. The chapter concludes with a summary and how the new approach can be extended.

6.2 Hierarchical Shape Analysis

Hierarchical Shape Analysis can be regarded as a technique able to derive and quantify correlated behaviour between any number of structures [127]. Among these techniques, it is possible to find the Hierarchical ASM [38], where the premise

of the method is that a small number of training samples can be used to estimate the covariation of patterns of smaller numbers of variables, allowing their hierarchical model to capture local and fine shape details. This is done by representing in a hierarchical fashion the shapes in terms of a wavelet transform followed by a PCA on the wavelet coefficients.

In [185] a technique called Partitioned Active Shape Model (PASM) is introduced, where these 3D PASMs can be regarded as partitioned representations of 3D ASMs. Here a mesh is partitioned into 'tiles' or surface patches, then PCA is applied to the coordinates of the tile vertices. Next, training samples are projected as curves in a single hyperspace so the deformed points are fitted into an allowable region of the model using a curve alignment scheme.

Rao *et al.* [127], presented an approach where two well known multivariate statistical techniques have been used to investigate the statistical variation of brain structures. Canonical Correlation Analysis is used to quantify the correlations between a number of different brain structures and Partial Least Squares Regression is performed over shapes of different structures to predict unknown shapes from unseen subjects, given known shapes of other structures from that subject. As well, CCA and PLSR facilitate the embedding of statistical shape modelling of the brain within a hierarchical framework since they can be used to extract and quantify correlated behaviour between any number of brain substructures.

More recently, Yu *et al.* [178] introduced a method that extracts shape features and conducts statistical analysis using a procedure that registers and normalises cortical surfaces, as well as decomposing them using spherical wavelets. The wavelet coefficients obtained are used as shape features to study the folding pattern at different spatial scales and locations, as the underlying wavelet basis functions have local support in the space and frequency domains. Then, the patterns of cortical shape variation are studied using PCA allowing the correlation of these shape variations with age and neuropsychological measurements at different spatial scales.

In table 6.1 an overview of these methods is provided. Each entry in the table contains information about the modelled object of interest, the image modality, and the techniques used for the work.

Authors	Object(s) of interest	Core technique(s)
Davatzikos *et al.* (2003)	Corpus Callosum	Wavelet transform PCA
Zhao *et al.* (2005)	Lateral ventricle Left Thalamus Left Hippocampus	PDM PCA Curve Alignment model fitting
Rao *et al.* (2006)	Lateral ventricle Pallidum Caudate Putamen Thalamus Amygdala Hippocampus Accumbens Brain Stem	PLSR CCA
Yu *et al.* (2007)	White Matter Grey Matter	SPHARM PCA

TABLE 6.1: Shape analysis tasks solved with Hierarchical Shape Models. Refer to the text for a description of abbreviations (Section 6.2).

6.3 Generating Hierarchical Shape Models

6.3.1 The Contour Laplacian Pyramid as a Compact Shape Code

In 1983 Burt and Adelson [23] introduced an image encoding technique that uses local operators at many scales as basis functions. The iterative process to encode the image generates a pyramid data structure, which is equivalent to sampling an image with Laplacian operators of many sizes, hence Laplacian pyramids. In this section a novel method for shape analysis is introduced based on the idea of Laplacian operators for image enhancement.

As in Burt and Adelson's method [23], the approach applies a Gaussian pyramid, $\mathcal{GP}$, as a first step. To generate a $\mathcal{GP}$ for any shape $\mathcal{S}$ (contour of form 2.16), it is 'reduced' using a Gaussian function that acts as a low pass smoothing filter and reduces the number of points. Such an operation is performed by the function $REDUCE$, that can be defined as:

$$REDUCE(\mathcal{S}) \;=\; \Downarrow(X(i,\sigma), Y(i,\sigma)) \tag{6.1}$$

where $X(i,\sigma), Y(i,\sigma)$ has the same form as in equation 4.1 and $\Downarrow$ denotes an operator that downsamples the number of points of a shape by two. Hence,

$$\mathcal{GP}(i) = REDUCE(\mathcal{S}) \tag{6.2}$$

with $i = 0, \ldots, l$ and l is the number of levels of the pyramid. Figure 6.1 presents an example of a Gaussian pyramid for a leaf and a brain white matter contour.

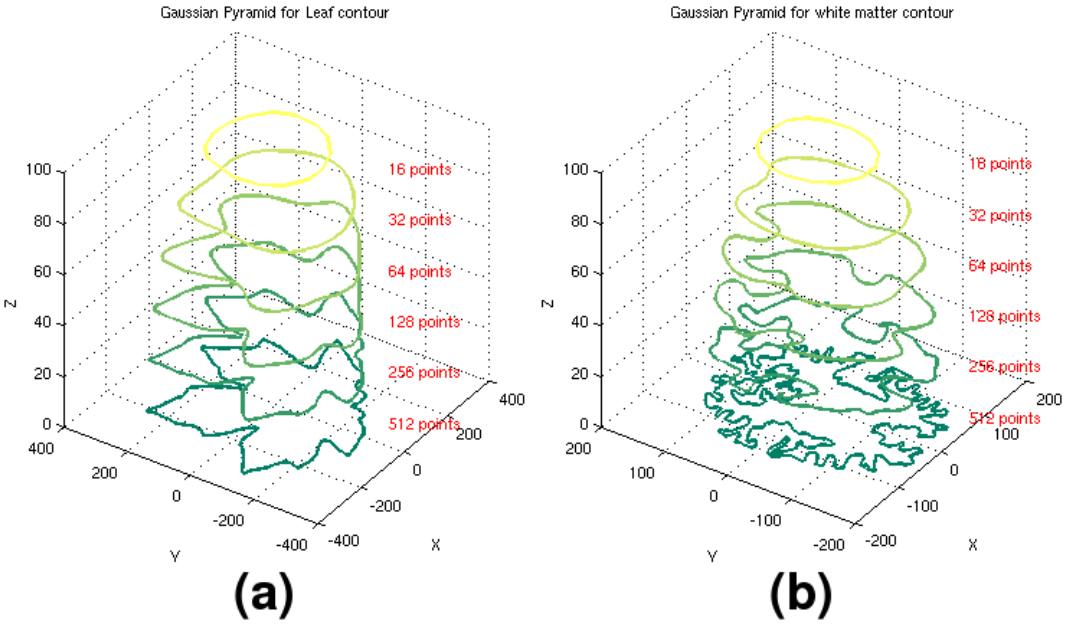


FIGURE 6.1: Example of Gaussian pyramid for: **(a)** leaf and **(b)** brain white matter contour.

The next step is the construction of the Laplacian pyramid $\mathcal{LP}$, where each level is a difference between two levels of the Gaussian pyramid. First the function

$EXPAND$ needs to be defined as:

$$EXPAND(\mathcal{S}) \;=\; \Uparrow(\mathcal{GP}(i)) \tag{6.3}$$

where $\Uparrow$ denotes an operator that upsamples the number of points of a given shape by two. Then, each level of the Laplacian pyramid $\mathcal{LP}$ is given by:

$$\mathcal{LP}(i) = \mathcal{GP}(i) - EXPAND(\mathcal{GP}(i+1)) \tag{6.4}$$

Since there is no level $i+1$ of $\mathcal{GP}$, to serve as a prediction level for $\mathcal{GP}(l)$ we say:

$$\mathcal{LP}(l) = \mathcal{GP}(l) \tag{6.5}$$

Figure 6.2 presents an example of a Laplacian pyramid for a leaf and a brain white matter contour.

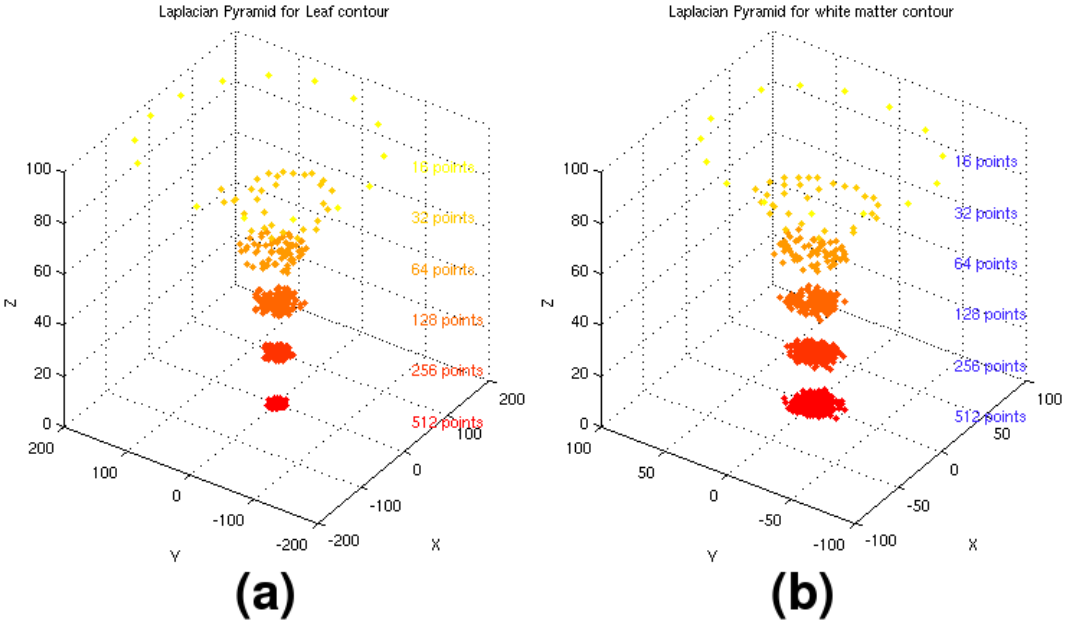


FIGURE 6.2: Example of Laplacian pyramid for: **(a)** leaf and **(b)** brain white matter contour.

Finally, it is possible to recover the original contour by, expanding $\mathcal{LP}(l)$ once and add it to $\mathcal{LP}(l-1)$, then expand this once and add it to $\mathcal{LP}(l-2)$, and so

on until level 0 is reached and $\mathcal{GP}(0)$ is recovered (the level that corresponds to the original contour). This procedure simply reverses the steps in the Laplacian pyramid generation. From equation 6.4 we have that:

$$\mathcal{GP}(i) = \mathcal{LP}(i) - EXPAND(\mathcal{GP}(i+1)) \tag{6.6}$$

Figure 6.3 presents an example of the reconstruction of the Gaussian pyramid for a leaf and a brain white matter contour respectively.

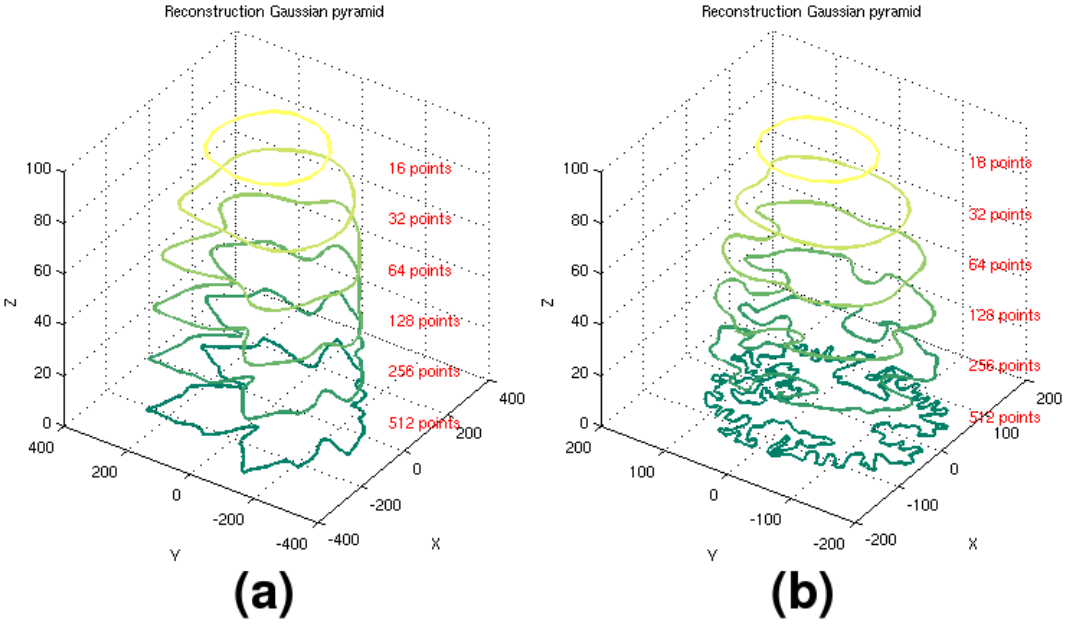


FIGURE 6.3: Example of the reconstruction of the Gaussian pyramid for: **(a)** leaf and **(b)** brain white matter contour.

6.3.2 Deriving shape information from Laplacian pyramids

Intuitively, the idea is that the shape variation is encoded in the Laplacian pyramid, so each level encodes the shape variation among the different levels of the Gaussian pyramid.

Having a set of shapes $\Phi = \mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_n$, the first step as in any other shape model is to align them. Then, to derive shape information, each level of the Laplacian pyramid $\mathcal{LP}$ is collapsed to a longer vector where the different levels

are concatenated $\mathcal{LV}$:

$$\mathcal{LV} = [\mathcal{GP}(l), \mathcal{L}_{(l-1)}, \mathcal{L}_{(l-2)}, \ldots, \mathcal{L}_{0)}]. \tag{6.7}$$

where $\mathcal{L}$ represents a level from $\mathcal{LP}$.

Then, a covariance matrix $\mathbf{C}$ is created by the outer product of each $\mathcal{LV}$:

$$\mathbf{C} = \frac{1}{n}\sum_{i=1}^{n}[\mathcal{LV}_i][\mathcal{LV}_i]^T \tag{6.8}$$

Then using equation 2.12 over $\mathbf{C}$ we obtain the corresponding eigenvectors and eigenvalues by PCA. As before, the eigenvalues or *modes* of variation effectively capture the variability of the set (refer to section 2.4.4.3 for more information).

6.3.3 Experimental results

The idea of the following experiment is to explore the compactness of the eigenplots derived from PCA. The data set used was leaves of the type Macrophyllum (see figure 2.6).

The following plots, reflect the results for each of the mentioned data sets for 4 different levels of the Gaussian pyramid and with different set sizes: 20 (Figure 6.4), 40 (Figure 6.5), 60 (Figure 6.6) and 80 (Figure 6.7) leaves were used.

In section 6.5 an evaluation of these results will be given. The next section shows the results from comparing the LSM with the HSM.

6.4 Comparative assessment: LSM and Laplacian Hierarchical Shape Model

In this section experiments were carried out to compare the proposed LSM and the novel Hierarchical Shape Model (HSM). For consistency, the same data sets for

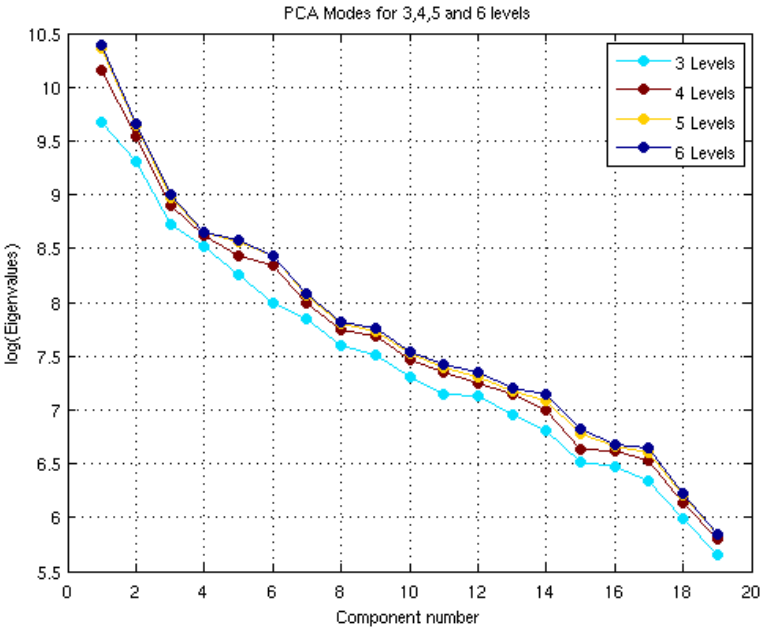


FIGURE 6.4: 20 leaves set: Log plot of eigenvalues against number of principal modes.

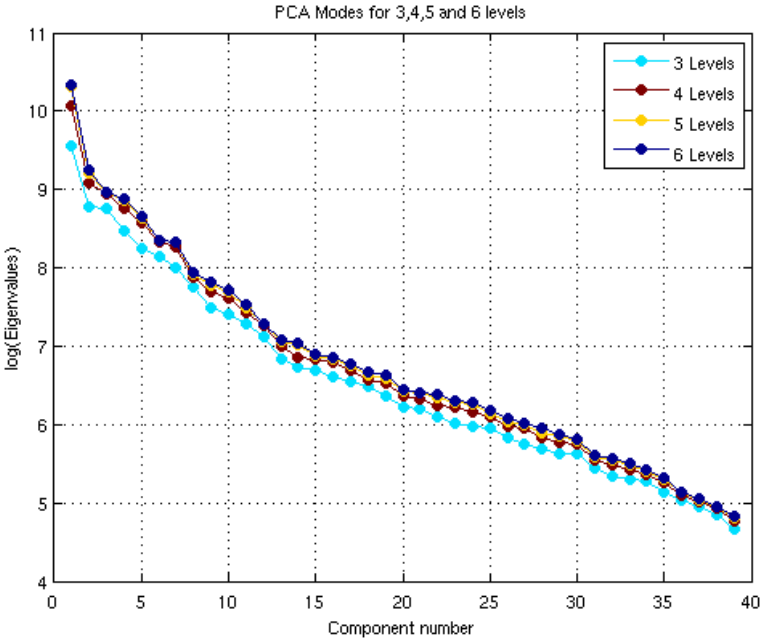


FIGURE 6.5: 40 leaves set: Log plot of eigenvalues against number of principal modes.

the experiments in Chapter 5 have been used. Here, the classes identified by the Diffusion Map clustering are compared against the HSMs. For both experiments, different levels of the Gaussian pyramid were used, and the idea was to compare the compactness of the eigenvalues from both models. In each of the following

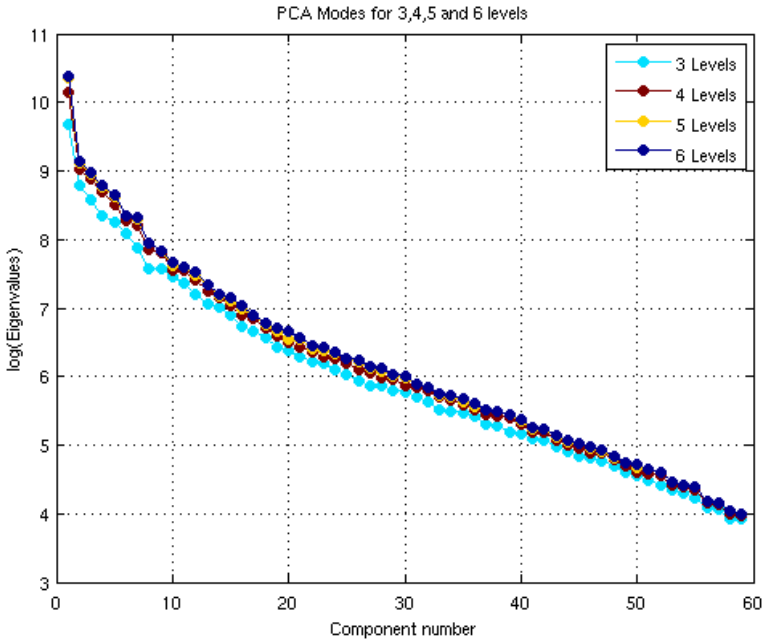


FIGURE 6.6: 60 leaves set: Log plot of eigenvalues against number of principal modes.

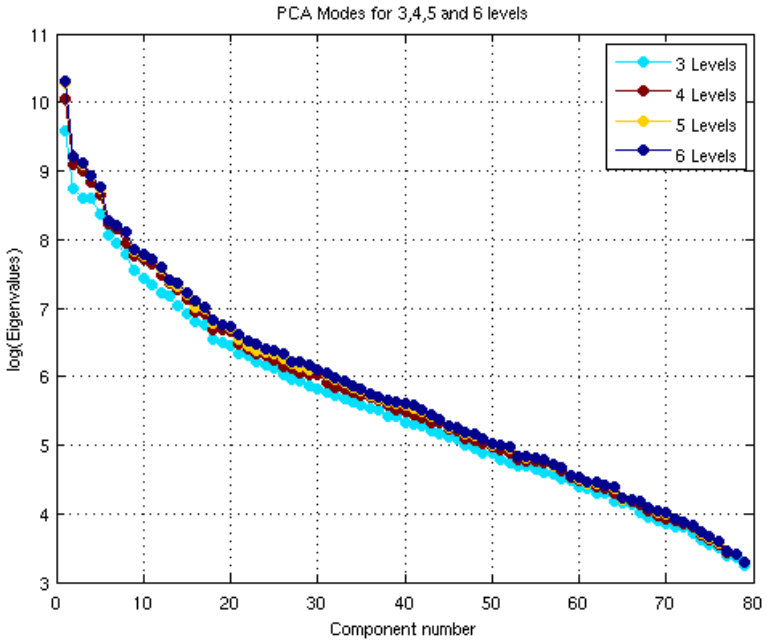


FIGURE 6.7: 80 leaves set: Log plot of eigenvalues against number of principal modes.

figures the resulting eigenmodes for the HSM are plotted in shades of green and for the LSM in red.

The first data set contains 50 leaves of the type Quercus Kelloggii. Figure 6.8

corresponds to the HSM using 3 Levels of the Gaussian pyramid, figure 6.9 to 4 levels, figure 6.10 to 5 and finally figure 6.11 to 6 levels.

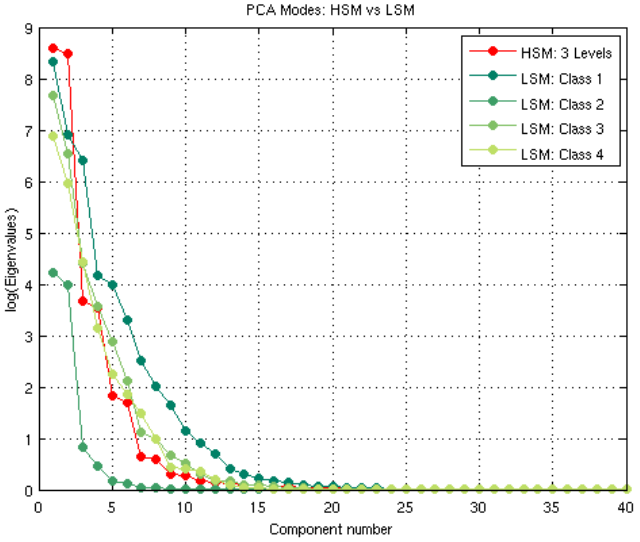


FIGURE 6.8: Kelloggii leaves set, 3 levels of the Gaussian pyramid: in red the plot for the HSM eigenmodes and in shades of green the different classes of the LSM, plot of eigenmodes against number of principal modes.

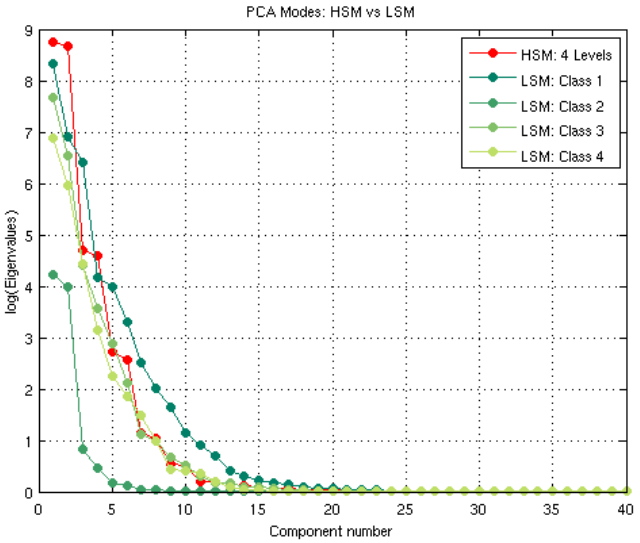


FIGURE 6.9: Kelloggii leaves set, 4 levels of the Gaussian pyramid: in red the plot for the HSM eigenmodes and in shades of green the different classes of the LSM, plot of eigenmodes against number of principal modes.

Second data set contains 60 brain white matter contours, the same shape set used for the experiments in Chapters 4 and 5. Again as in the previous data set, each

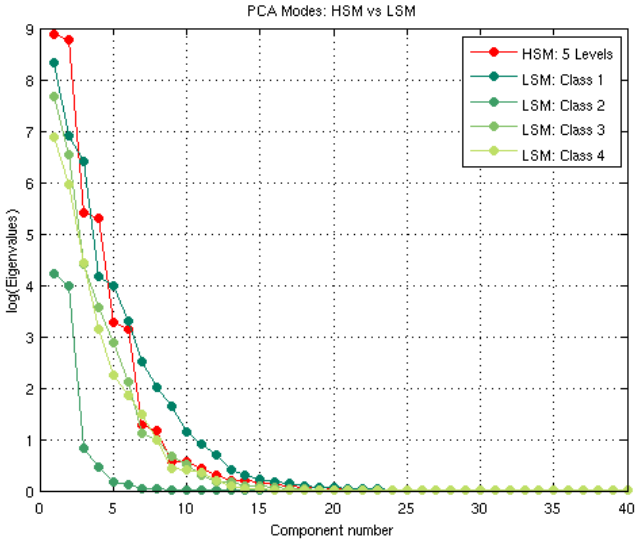


FIGURE 6.10: Kelloggii leaves set, 5 levels of the Gaussian pyramid: in red the plot for the HSM eigenmodes and in shades of green the different classes of the LSM, plot of eigenmodes against number of principal modes.

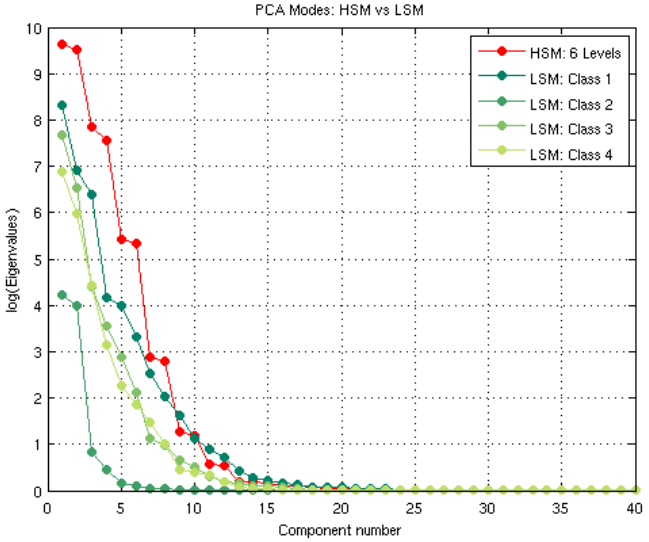


FIGURE 6.11: Kelloggii leaves set, 6 levels of the Gaussian pyramid: in red the plot for the HSM eigenmodes and in shades of green the different classes of the LSM, plot of eigenmodes against number of principal modes.

figure (6.12 to 6.15) corresponds to the HSM using 3, 4, 5 and 6 Levels of the Gaussian pyramid accordingly.

In the next section an evaluation of these results is given.

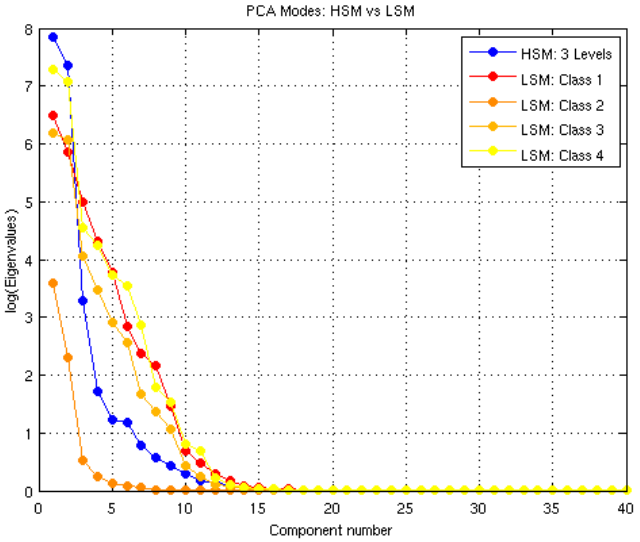


FIGURE 6.12: Brain white matter set, 3 levels of the Gaussian pyramid: in blue the plot for the HSM eigenmodes and in warm colours the different classes of the LSM, plot of eigenmodes against number of principal modes.

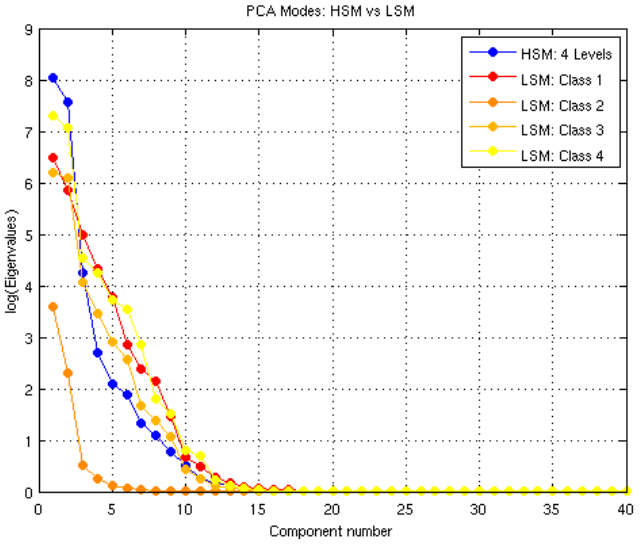


FIGURE 6.13: Brain white matter set, 4 levels of the Gaussian pyramid: in blue the plot for the HSM eigenmodes and in warm colours the different classes of the LSM, plot of eigenmodes against number of principal modes.

6.5 Evaluation and Discussion

The resulting plots of section 6.3.3 (figures 6.4 to 6.7) show that even using different levels of the Laplacian pyramid and different sizes of the training sets, the plots do not present significative variation in their compactness. Hence, from the

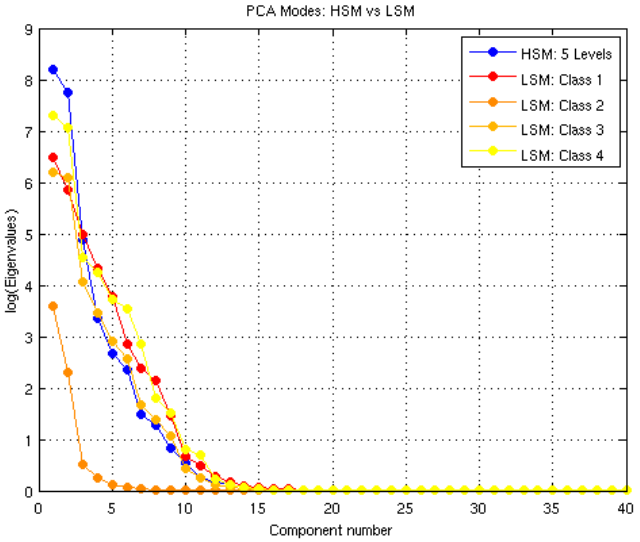


FIGURE 6.14: Brain white matter set, 5 levels of the Gaussian pyramid: in blue the plot for the HSM eigenmodes and in warm colours the different classes of the LSM, plot of eigenmodes against number of principal modes.

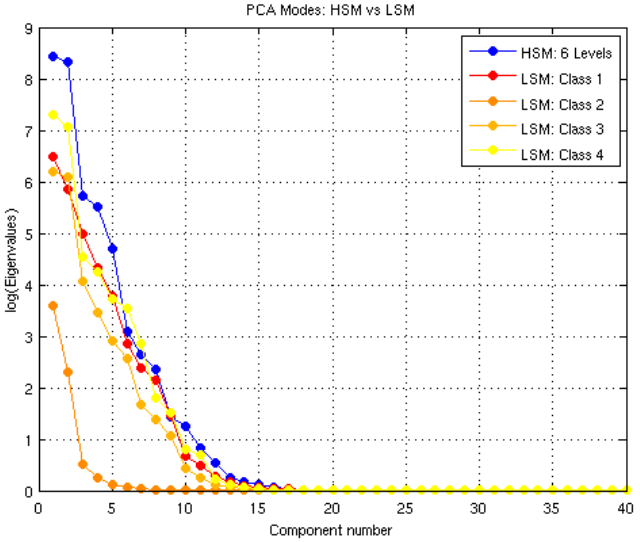


FIGURE 6.15: Brain white matter set, 6 levels of the Gaussian pyramid: in blue the plot for the HSM eigenmodes and in warm colours the different classes of the LSM, plot of eigenmodes against number of principal modes.

experiments it is possible to conclude that the compactness of the method is consistent, and therefore using few levels or a small set of shapes will not affect the performance of the method.

Results from leaf contours (figures 6.8 to 6.11) and from white matter shapes

(figures 6.12 to 6.15) illustrate that, in most cases, the HSM appears not to be more compact than the LSM since is not possible to find significative variation in the compactness of the eigenmode plots. Even though one can note that in the construction of the HSMs there is more information coded for each shape, the accuracy in deriving shape information does not always outperform the proposed LSM. This can be graphically assessed in the resulting plots, where none of the plots using different levels of the HSM were more compact than the ones produced by the LSM. This means that the HSMs can be as good and compact as the LSM, and consequently, this shows that the new method is worthy for performing shape analysis but work in a different way than the proposed LSM. Additionally, with this experiments and the experiments from Chapter 5, it is possible to conclude that the proposed LSM is a compact and reliable method to perform shape analysis.

At the moment there is the scope for extending the presented technique in two ways. The first is to adjust it to Local shape analysis by taking parts of the contour at different scales of smoothing according to the CSS, and build its corresponding Laplacian pyramids. The other direction will be to extend it to three dimensions using surface patches. In [114], it is shown that by convolving local parameterisations of the surface with 2D Gaussian filters iteratively it is possible to obtain smoothed versions of the patches. This should be enough to derive the Gaussian and Laplacian pyramids needed for the proposed HSM.

6.6 Summary

In this chapter a new shape model that derives shape information in a hierarchical fashion was presented. Hierarchical Shape Analysis can be regarded as a technique able to derive and quantify correlated behaviour between any number of structures. To create a multilevel analysis, the method employs the same idea as in Burt and Adelson's method [23], i.e. that of using an iterative process to encode the image generates a pyramid data structure, which is equivalent to sampling an image with Laplacian operators of many sizes, hence Laplacian pyramids. In this case, the

idea is that the shape variation is encoded in the Laplacian pyramid, so each level encodes the shape variation among the different levels of the Gaussian pyramid. The basic principles behind the method were explained along with examples to illustrate its construction. Similar to other experiments in this work, brain and leaf data sets were used to obtain results of the new model. From the experiments, it is possible to conclude that the compactness of the method is consistent, and when using few levels or a small set of shapes should not affect the performance of the method.

Likewise, this new method has been used to establish a comparison with the LSM proposed in previous chapters. The experiments used the classes identified by the Diffusion Map clustering, compared against the HSMs resulting modes. Results from leaf contours and from white matter shapes illustrate that, in most cases, the HSM appears not to be more compact than the LSM since is not possible to find significative variation in the compactness of the eigenmode plots. This means that the HSMs can be as good and compact as the LSM, and consequently, this shows that the new method is worthy for performing shape analysis but works in a different way than the proposed LSM.

Further extensions of the proposed LSM are discussed in the following chapter, together with a summary and conclusions of this research work.

Chapter 7

Conclusions

7.1 Thesis Summary

The work described in this thesis has been concerned with the development of a statistical shape model to analyse local variation in shapes. The proposed method is based on a Point Distribution Model, so the objective has been to analyse the correlation of the shape of parts of certain structures. In this case, natural shapes (leaves and squids) and brain structures were considered as the research data. With the purpose of overcoming some of the aforementioned problems that normally arise when using Active Shape Models, here an extension of the work of Bhalerao and Wilson [15] was reviewed. In that work, a major conclusion was that the localisation of the shape space to contour parts implies that periodic and repetitive structural variation can be modelled using a single training sample. Here, solutions to cope with the issue of what should be a good partition of a given contour into local parts were presented. From all the results in this work it is possible to conclude that local shape models are efficient in modelling the statistical variation of shape of natural structures.

Chapter 1 had the purpose of setting the scene and motivate the work. First, a general introduction to medical image analysis was given. Here, the importance

of Computational Anatomy as a new discipline with its objective of creating algorithmic tools to help in the analysis of the substructures of the human brain was explained. Next, some background on fractal brain structure was introduced since this is used as main driving direction for this research. Furthermore, the importance of fractal geometry in many natural objects was given by the examination of a series of state of the art research papers that show that it is possible to regard the brain as a natural fractal. The next section hightlighted the need for local shape models compared with the global ones. As one of the objectives was the partitioning of contours, a review of previous work in this area was presented. Then, spectral clustering was introduced as a general method to construct sets of contour partitions useful for shape analysis. Also in this section, a statement of purpose was given and the thesis contributions, materials and outline presented.

Chapter 2 was mainly dedicated to the explanation of basic background of shape and statistical shape analysis. The chapter starts with an introduction and then in the next section, basic but important concepts such as shape, landmarks, shape descriptors, shape spaces and shape signatures were explained. Then, in the following section Morphometry was introduced, followed by a brief introduction to Shape Analysis. In the next section, first Statistical Shape Analysis was described and then four basic parts of any SSA method are then explained: Procrustes methods, PCA, Eigenshape analysis and PDM. The last one is described in a more complete way since it comprises most of the aforementioned concepts and because it is the shape descriptor used in this work. Hence, the next part of this section, contour localisation or contour partitioning was explained along with a formal definition of partitioning. Next, registration or the pose removal problem was explained, along with an explanation of the Procrustes method. After that, a description of how statistical variation in the shapes can be characterised by Principal Components Analysis. PCA is a statistical technique aimed at reducing the dimensionality of a data set, but it also useful to summarise information, in this case, to express the shape variability of a data set. In the following part of the section, it was explained how shape reconstruction can be carried out by retaining the pose parameters obtained in the registration process and a novel, alternative

way to visualise results was created. This consisted of presenting the reconstructed partitions blended back into the smoothed version of the contour to which they belong. Finally, a chapter summary was given.

Fractals, and in this case, space filling curves are considered important mathematical descriptions of the appearance and shape of common objects in nature. In Chapter 3 their properties and how the property of self-similarity can be used in shape analysis was explained. This chapter presented a new method for modelling fractal curves, such as the boundary of brain white matter, and partitioning such curves into segments with equal fractal dimension. For a given number of contour points and a required set of partitions, the solution space is very large. A Bayesian framework with reversible-jump Markov Chain Monte Carlo was developed with a sampler based on the Metropolis-Hastings test. An introduction to fractal objects was carried out in the first part of this chapter, and in particular space filling curves, and the fractal dimension. The rest of the chapter was devoted to reviewing Bayesian statistics concepts, with particular reference to, Monte Carlo methods, Markov Chains and the Metropolis-Hastings algorithm, which is used to explore the solution space and find an appropriate solution.

Monte Carlo methods are often used to simulate physical and mathematical systems. Based on repeated computation and random or pseudo-random numbers, Monte Carlo methods are suited to calculation by computers. Such methods tend to be used when it is impossible to compute an exact result with a deterministic algorithm. The implemented algorithm was explained in detail, and results were presented on simple contours (animal silhouettes) and space-filling brain contours. The chapter concluded by discussing an application of the algorithm, and its use for building local statistical shape models.

Chapter 4 reviewed the Curvature Scale Space technique as a powerful and general shape analysis descriptor that has been developed comprehensively during the last 20 years. The method consists of describing curves at various levels of detail using features that are invariant with respect to transformations that do not change the shape of the curves. The first part of this chapter stated the principles of the

method, where the main idea is to find a representation by convolving curves with a Gaussian function to extract the curvature zero-crossing points of the resulting curves. Devised as a useful and natural way to set partitions on a contour, the rest of the chapter explained how the Curvature Scale Space Representation is used in this research. A tool built using CSS was presented and is explained in Appendix B.

Due to the complexity of a high dimensional data set, it is often desirable to represent this data using as few parameters as possible, whilst preserving the relevant information. It is assumed that the original data exists on a lower dimensional manifold, that is embedded in a high dimensional feature space. Under this assumption, it may be possible to learn the underlying manifold that generates the data by using approaches based on graph-theory and differential geometry. It has been shown that the eigenfunctions of Markov matrices can be used to construct coordinates called Diffusion Maps that generate efficient representations of complex structures, by embedding the data points into a lower dimensional space, giving an insight into the geometry of the dataset. In Chapter 5 all these concepts were explored to achieve dimensionality reduction, and also improve the clustering of data sets for use in shape analysis. The first part of this chapter introduced Fourier descriptors as a method of representing a given shape contour. Then we continued with the definition of manifolds, manifold learning, spectral clustering, an explanation of diffusion maps and finally, how a model can be built using the curvature zero-crossing points from CSS. Results on different data sets were given, with particular attention paid to the leaf and brain data sets. Next, a discussion of the results is given along with a demonstration of the compactness of the proposed LSM by building a set of local (linear) shape models and then comparing each model against a single global one. Finally, possible ways to improve the obtained results have been given.

Finally, Chapter 6 introduced a new SSM that is based on an image encoding technique that uses Laplacian operators, more specifically Gaussian and Laplacian pyramids. Since the method extracts and quantifies correlated behaviour between the different levels of the Laplacian pyramid created for each contour of

the set, we regard this as Hierarchical Contour Shape Analysis. In this chapter an introduction to hierarchical shape analysis was given, including a review of relevant work in the field. Similarly, details about the construction of the Laplacian pyramids and the way to derive the shape information encoded on them were explained. From the experiments it is possible to conclude that this method is consistent, since the size of the data sets does not affect its compactness. This technique was used to perform a comparative assessment between this model and the LSM. Results of this comparison indicate that, in most cases, the HSM appears not to be more compact that the LSM by looking at the eigenmode plots. Hence, such shape models are worthy for performing shape analysis but in a different way.

7.1.1 Original Contributions

The main contributions of this work have been:

- The creation of a method cast in the Bayesian framework combined with fractal analysis to estimate a set of contour partitions. A Markov Chain Monte Carlo method was used along with the box counting dimension to characterise fractal dimension.

- The utilisation of the Curvature Scale Space descriptor to obtain a set of zero-crossing points that was used to create consistent partitions in contours.

- A method for shape clustering useful in shape analysis was presented. Such a model combines the CSS representation and Spectral clustering to create clusters of closed and non-closed contours in an unsupervised way.

- A novel Hierarchical Shape Analysis method based on the concept of the Gaussian and Laplacian pyramids was proposed. The method is able to extract in a hierarchical fashion, the relevant shape variation and is used to compare the performance of the LSM.

7.2 Discussion

This thesis provides a contribution to the problem of modelling *local* versus *global* variation of shape. Local SSM approaches like in [16] have been demonstrated to be effective in the characterisation and analysis of the variability of shape, based on pose-alignment and an SSM. The novel contribution of this work is the creation of a local statistical shape model that provides a way to cut-up a self similar contour, obtain a set of meaningful shapes, obtain the shape variation and reconstruct the contours. The presented method is based on the use of a SSM, specifically a PDM, and the Curvature Scale Space representation of shapes.

Initially, one of the pursued aims of the LSM was to follow a semi-automatic methodology through partial automation of different tasks of the model. The author considers the model general enough to be able to represent a broad range of shapes so it is useful in a wide range of medical or biological applications. To some extent the proposed model has some robustness because it adopts a scale space representation that is invariant to noise. All these are desirable characteristics for the construction of effective models for medical image analysis [105].

7.2.1 Contour Localisation

This work introduces two different approaches to answer what should be a good partition of any given contour into local parts and how this may be determined in a simple and efficient way. The first approach is based on fractal dimension and the Metropolis-Hastings algorithm to determine how to break up a contour in partitions. Fractals and in particular space filling curves defy shape analysis because of the relative complexity of the shapes. Therefore it is hard to establish meaningful correspondences between shapes and require a much larger training size. Despite this, the proposed solution is not stable enough to produce a best set of partitions with fixed number of points but it is, to the knowledge of the author, the first attempt to solve the problem by exploiting the fractal nature of the brain. The second approach presented looked for a solution that overcomes

the issues of the MCMC approach. In this case the CSS representation was used to find a natural way to create partitions of a contour by local variation given by the zero-crossings of the CSS algorithm. Although other approaches have used Scale Space representations for such purposes, the proposed method is a simple, yet effective, way to localise the contour. Moreover, since brain contours and other natural structures like leaves exhibit self-similarity, using the CSS method its possible to establish (at certain scales) possible partitions derived from the extrema points that present local correspondences. This provides a natural way to identify consistent parts in the contour by their local variation.

7.2.2 Shape Clustering

Once a robust enough technique for contour localisation was defined, the next question to address was how could the partitions be grouped in an effective way to create suitable sets for shape analysis. Here again we explored two different solutions. The first one involved a supervised clustering performed with the help of a tool with a graphical interface to facilitate the work flow and visualise results directly. Although it is useful to select any partition from the generated data set of shapes and obtain a set of similar figures that keeps the shape space relatively compact, the need for interaction is a major drawback of any shape model. Hence, another approach was investigated, based on spectral clustering techniques. In this case Diffusion Maps were used to create corresponding clusters of shapes. The method used Fourier transforms of the centroidal distance function (shape signature), and therefore only half of the Fourier coefficients were needed. This results in loss of fine detail when analysing the shape, but as the shapes in question were obtained through the CSS technique they were already smoothed versions of the original partitions. This implied that no meaningful local details were lost. Similarly, another concern of the method arose in relation to the use of non-closed contours. In the reported experiments of the spectral clustering literature only closed contours are used, but this work presents fairly successful results using non-closed contours derived form the contour localisation. Therefore the proposed

method based on Fourier analysis and Diffusion maps was found to be adequate to find clusters of shapes useful for the local shape analysis in an unsupervised manner.

7.3 Limitations and Further Research

A limitation of the presented method is the need to re-sample the partitions to have the same length in order to align and perform PCA. Unless the original contour has many points, any small local part may not have enough to do simple piece-wise-linear re-sampling. Here, a basis representation would help. In [16], Legendre functions to both represent and align parts have been used. Similarly, in [115] a strongly related work was presented. There, the CSS technique was used in conjunction with Hermite curves for automatic fitting of digitised contours at multiple scales. Additionally, in [168], a B-Spline curve representation is used to model the shapes, and then smooth them by increasing the degree of the curve. This is used to construct the CSS image with an explicit representation. More recently Manousopoulos *et al.* [101] presented a method for fractal interpolation that provides a way to describe data that have an irregular or self-similar structure. Other curve representations might be useful as well, but the mentioned approaches seem to be the most promising alternatives to extend the work developed in this research.

7.3.1 From 2D to 3D

For quantitative morphologic assessment in medicine, volume is the most common choice. Volume is a coarse measure that strives to capture shape differences that can improve the diagnostic accuracy. Assessment of individual brain structures is often based on volumetric measurements. Volume changes can be regarded as intuitive features as they might explain atrophy or dilation due to illness. On the other hand, structural changes at specific locations are not sufficiently reflected

in volume measurements [153]. Because most existing methods use complex and difficult to reproduce shape descriptors, the experts prefer more easily and robustly extractable area and volume measures in their diagnosis [167]. Even though quantitative shape features such as volumetric measurements present problems in modelling changes at specific locations, they have been widely used in statistical shape analysis of diverse anatomical structures in previous years. In the particular case of the training data for SSMs in the medical field, the most common choice are segmented volumetric images. Then, depending on the chosen segmentation method, the initial representation might be binary or fuzzy voxel data; or surface meshes [68].

Hence, it might be natural to extend the proposed method to 3D in order to assess structural changes at specific locations of volume measurements. Therefore, a proposal for further research is to extend the method to surfaces to be properly validated with clinical data. However, it is not clear at this time how the local partitioning could be easily extended to surface patches. Presumably, the simplest and most generic method used to represent shapes is a set of points distributed across the surface, the aforementioned landmarks, but there are other possibilities to represents shapes. A typical one is the use of medial models or skeletons introduced by Blum [18] and used to represent objects by their centre lines an the corresponding radii. Pizer *et al.* presented a medial model representation for two dimensions in [125], and later extended to 3D in an approach termed M-rep [124]. Another popular representation is the one proposed by [21] namely spherical harmonics (SPHARMs) where a set of basis functions can be used to describe closed surfaces of spherical topology. Alternatively, Davatzikos *et al.* presented a shape description using wavelets [38]. In this approach, the statistical properties of the wavelet transform of a deformable contour are analysed via PCA and used as priors for contour deformation. Only some of the most popular 3D representations have been introduced here, but for details on these and other proposed ones refer to [68] and [184].

A possible solution can be envisaged using the CSS technique extended to surfaces. In [113] a multi-scale smoothing of 3D surfaces is presented. Complete triangulated

models of 3D objects are constructed and then described at multiple scales. This is achieved by convolving local parameterisations of the surface with 2D Gaussian filters iteratively. The smoothing gradually eliminates surface noise and small details [114]. Based on the 2D idea, similarly here, something could be done to find partitions on surfaces, that is to use extrema curvature points to obtain regions of interest. Moreover, the latter can be combined with other approaches, such as Fractal Dimension. In [79] an analysis of the cortex geometry is given, where the fractal dimension is calculated in 3D maps of the cortical gray matter. Hence, it seems possible to combine both approaches to produce a patch partitioning method.

It is important to mention that even though methods can be generalised or extended to the 3D case, many methods have only been applied to 2D so far. This is because even though such methods are very successful in 2D, they are not technically feasible in 3D [68].

Clearly, our approach has a number of applications in shape modelling of natural shapes, such as in biology and medical imaging. Particularly, the development of this model could provide a way to analyse brain images and perform brain morphometrics. Moreover, it might contribute also as a useful method in the construction of population-based brain atlases. These atlases have different applications such as image labelling or pathology detection either in individuals or groups. Research might lean to recognition of specific patterns of anatomic alterations due to diseases (Alzheimer's) or dementias (schizophrenia, epilepsy, ADHD, or autism). So the provided method could be adapted for clinical diagnosis software for assessing changes in local shape variation of anatomical structures, such as, white/gray matter. Similarly, the model might be adapted for the problem of image database retrieval where the objective can be to discover images which contain objects similar to query objects, in this case brain sections.

Appendix A

Publications by the Author

- Valdes-Amaro, D. A. and Bhalerao, A. (2008). Local Shape Modelling for Brain Morphometry using Curvature Scale Space. In McKenna, S. and Hoey, J., editors, *Proceedings of the 12th Annual Conference on Medical Image Understanding and Analysis 2008*, pages 64-68. British Machine Vision Association.

- Valdes-Amaro, D. and Bhalerao, A. (2008). To Boldly Split: Partitioning Space Filling Curves by Markov Chain Monte Carlo Simulation. In Gelbukh, A. F. and Morales, E. F., editors, *MICAI 2008: Advances in Artificial Intelligence, 7th Mexican International Conference on Artificial Intelligence, Atizapan de Zaragoza, Mexico, October 27-31, 2008, Proceedings*, volume 5317 of Lecture Notes in Computer Science, pages 543-553. Springer.

- Valdes-Amaro, D. and Bhalerao, A. (2009). Unsupervised Clustering Using Diffusion Maps for Local Shape Modelling. In Moreno-Daz, R., Pichler, F., and Arencibia, A. Q., editors, *12th International Conference on Computer Aided Systems Theory, EUROCAST 2009, Las Palmas de Gran Canaria, Spain, February 15-20, 2009, Proceedings*, volume 5717 of Lecture Notes in Computer Science, pages 342-349. Springer.

Appendix B

CSS Interface

The following describes the usage of the graphical interfaces created to facilitate obtaining results from the methodology described in Chapter 4. Different windows were created corresponding to the three main steps of our methodology (Figure 4.5).

The first window (Figure B.1) is where the new contour to be analysed is displayed smoothed, and a slider is provided to establish the smoothness coefficient. Six action buttons are included in this window. The *Smooth Contour* button performs the smoothness over the input contour once a coefficient is provided. Then the smoothness is performed and the contour is adjusted to its new shape, so the zero-crossing points are displayed in red over it. The second button, *Select Points*, is used to select the points from which a section of the contour is going to be analysed, called here the *reference partition*. It is only possible to use the red points on the contour since those represent meaningful information about the geometry of the smoothed contour. The *Search* button performs a search of the partition with less MSE in each contour of the input set. The next two options are similar in the way they perform the analysis, but they differ in the size of the shape set they work on. The option of *Slice Analysis* opens a new window that displays all the available shapes of the set, the user selects one, a ranking of partitions is created for it and then one or more partitions can be selected to be analysed. The option of *All Slice Analysis* uses all the shapes on the set to create a rank and then select

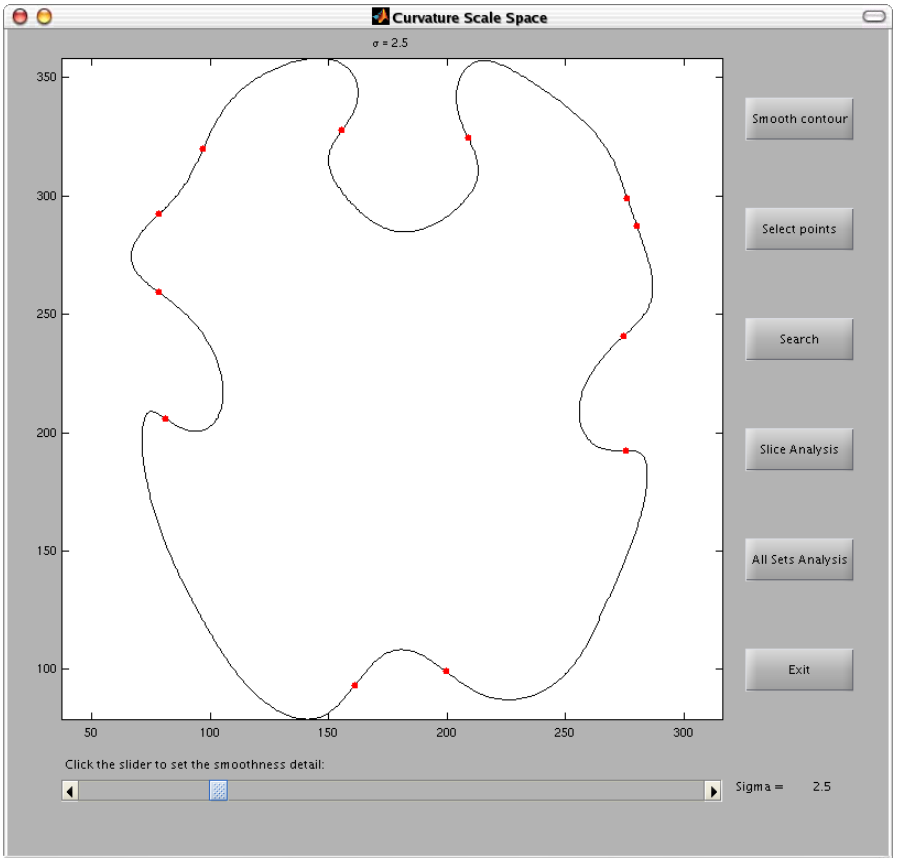


FIGURE B.1: Initial window where the user can smooth the contour and choose other analysis options using the interface buttons

only a partition to be analysed. Both will be explained later on the section with more detail. Finally, the *Exit* button closes the application.

The *Slice Analysis* (Figure B.2) window where the available slices are displayed to be selected, a slider is provided to navigate the set. Once a shape is selected, in this case a brain slice, the ranking plot is created and presented, and on the left hand of this plot there is a slider provided to set an error threshold, useful to select a number of partitions that are shown in the adjacent plot window. Here the reference partition is shown in gray, and the rest of the partitions are drawn in multiple colours. Two smaller plot boxes are given, where the selected partitions displayed on the second plot window are shown in its original position over the original contour and on the smoothed version for a better visual control. In addition, we provide the user with the option to select a specific partition from the ranked set, and if selected, it will be displayed in red only with the reference

one. As before, such a partition will be plotted in its original contour position and in the smoothed version.

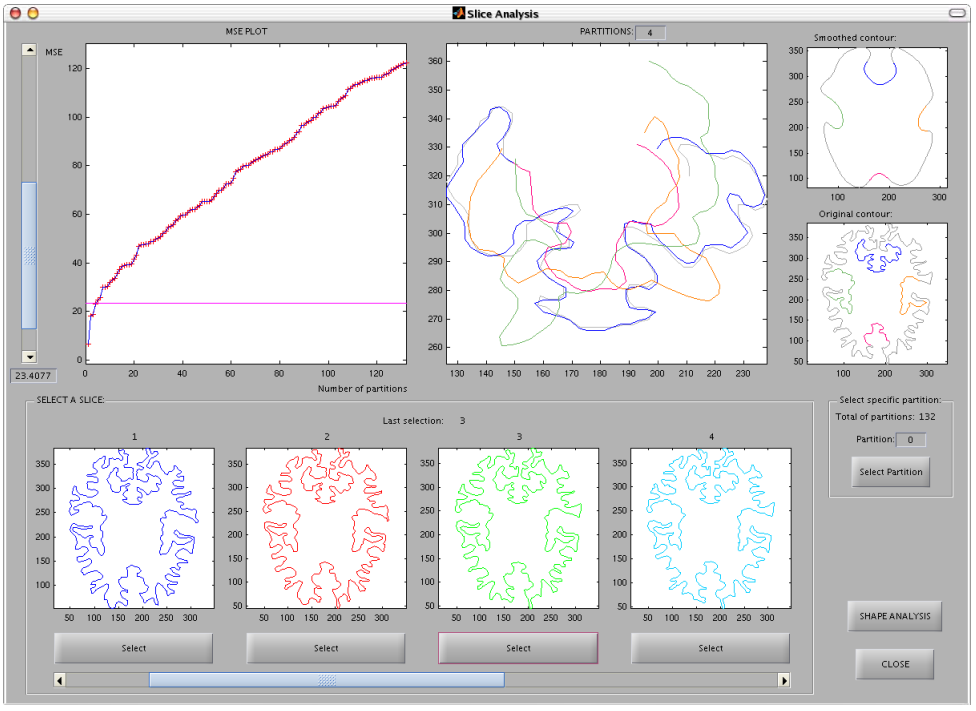


FIGURE B.2: Slice analysis window

The option of *All Slice Analysis* (Figure B.3) displays a window where the main feature is that of the ranking plot is created by using all the partitions of all the available contours. Here the objective is select one of the partitions to be analysed, so this is displayed on the adjacent plot window in red and as well the reference partition is plotted in gray. Likewise the current partition is displayed in its original context, first in the original contour and then in the smoothed version in the smaller boxes provided. Once a partition is select the statistical analysis can be carried out.

The third part of the system is the Statistical Analysis windows. When the user has selected to use only a slice (Figure B.4) or a all of them (Figure B.5), depending on the selection of the second step, then is possible to perform the statistical analysis. The purpose here is to display an interactive reconstruction of the shapes after a PCA. A slider is provided to select the number of modes for the reconstruction or as well, a single mode can be selected to analyse its contribution to the reconstruction.

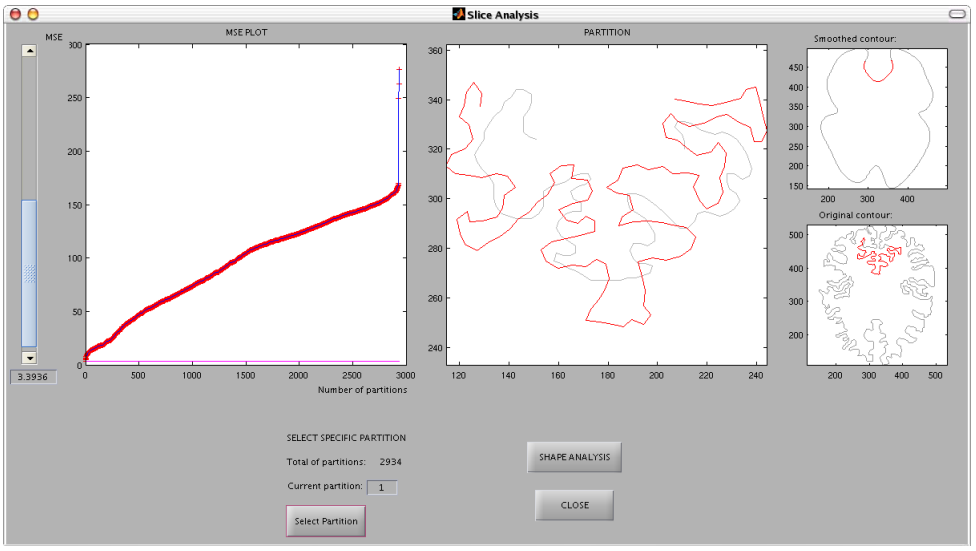


FIGURE B.3: All slice analysis window

A couple of plot windows are provided to give a graphical representation of the eigenvalues derived from the covariance matrix.

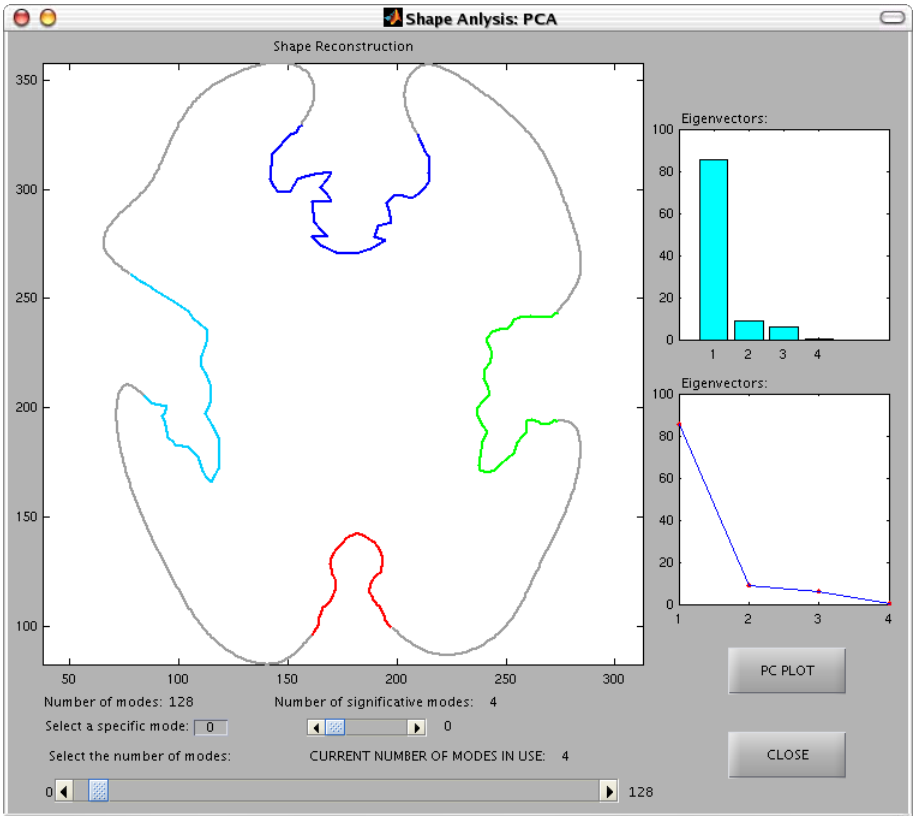


FIGURE B.4: PCA window for the case when only one slice is selected.

An important feature of this analysis and of this work is the way the reconstructions are presented. Here the idea is to produce a new way of visualise results by presenting the reconstructed partitions blended back into the smoothed version of the contour they belong to. The blending is achieved by using squared cosine window functions as detailed in section 2.7.

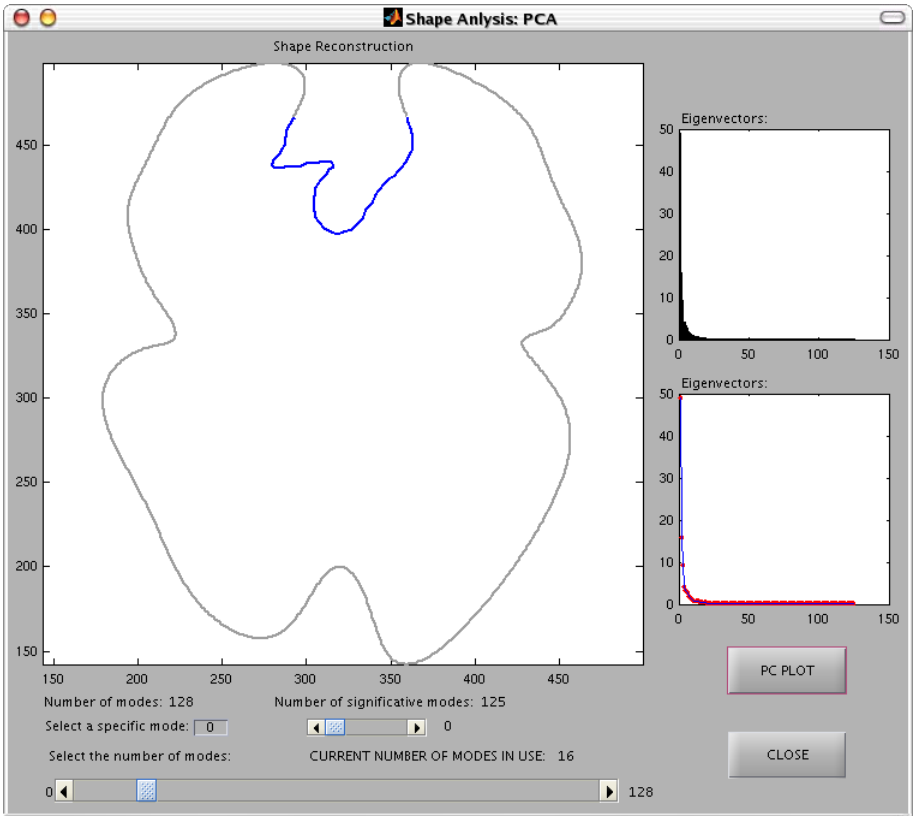


FIGURE B.5: PCA window for the case when only one slice is selected.

Bibliography

[1] S. Abbasi, F. Mokhtarian, and J. V. Kittler. Enhancing CSS-based shape retrieval for objects with shallow concavities. *Image and Vision Computing*, 18(3):199–211, Feb 2000.

[2] D. C. Adams, F. J. Rohlf, and D. E. Slice. Geometric morphometrics: Ten years of progress following the 'revolution'. *Italian Journal of Zoology*, 71(1):5–16, 2004.

[3] A. A. Afifi. *Computer-aided multivariate analysis*. Chapman and Hall, London, 2004.

[4] Mark S. Aldenderfer and Roger K. Blashfield. *Cluster Analysis*. Sage Publicatons, 1984.

[5] S. G. Alves, M. L. Martins, P. A. Fernandes, and J. E. H. Pittella. Fractal patterns for dendrites and axon terminals. *Physica A: Statistical Mechanics and its Applications*, 232(1-2):51–60, 1996.

[6] T. W. Anderson. Harold Hotelling's Research in Statistics. *The American Statistician*, 14(3):17–21, 1960.

[7] Haruo Asada and Michael Brady. Curvature Primal Sketch. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):2–14, 1986.

[8] J. Ashburner, J. G. Csernansky, C. Davatzikos, N. C. Fox, G. B. Frisoni, and P. M. Thompson. Computer-assisted imaging to assess brain structure in healthy and diseased brains. *Lancet Neurology*, 2:79–88, 2003.

[9] J. Ashburner, C. Hutton, R. Frackowiak, I. Johnsrude, C. Price, and K. Friston. Identifying global anatomical differences: Deformation-based morphometry. *Human Brain Mapping*, 6:348–357, 1998.

[10] B. Aubert-Broche, M. Griffin, G. B. Pike, A. C. Evans, and D. L. Collins. Twenty new digital brain phantoms for creation of validation image data bases. *IEEE Transactions on Medical Imaging*, 25:1410–14163, 2006.

[11] D. Avnir, O. Biham, D. Lidar, and O. Malcai. Is the geometry of nature fractal? *Science*, 279(5347):39–40, 1998.

[12] Orit Baruch and Murray H. Loew. Segmentation of two-dimensional boundaries using the chain code. *Pattern Recognition*, 21(6):581–589, 1988.

[13] Mikhail Belkin and Partha Niyogi. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, 15(6):1373–1396, 2003.

[14] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape Matching and Object Recognition Using Shape Contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[15] Abhir Bhalerao and Roland Wilson. Local Shape Modelling Using Warplets. In Heikki Kälviäinen, Jussi Parkkinen, and Arto Kaarna, editors, *Image Analysis, 14th Scandinavian Conference, SCIA 2005, Joensuu, Finland, June 19-22, 2005, Proceedings*, volume 3540 of *Lecture Notes in Computer Science*, pages 439–448. Springer, 2005.

[16] Abhir Bhalerao and Roland Wilson. Warplets: An image-dependent wavelet representation. *International Conference on Image Processing, ICIP*, 2:490–493, 2005.

[17] R. E. Blanton, J. G. Levitt, P. M. Thompson, K. L. Narr, L. Capetillo-Cunliffe, A. Nobel, J. D. Singerman, J. T. McCracken, and A. W. Toga. Mapping cortical asymmetry and complexity patterns in normal children. *Psychiatry Research - Neuroimaging*, 107(1):29–43, 2001.

[18] Harry Blum. Biological shape and visual science. *Journal of Theoretical Biology*, 38(2):205–287, February 1973.

[19] F. L. Bookstein. *Morphometric Tools for Landmark Data: Geometry and Biology*. Cambridge University Press, Cambridge, 1991.

[20] Paul Bourke. Fractals, chaos, url: http://local.wasp.uwa.edu.au/∼pbourke/fractals/, 2009.

[21] Ch. Brechbühler, G. Gerig, and O. Kübler. Surface Parametrization and Shape Description. In *SPIE Workshop on Visualization in Biomedical Computing*, pages 80–89, 1992.

[22] X. Bresson, P. Vandergheynst, and J. P. Thiran. Multiscale Active Contours. *International Journal of Computer Vision*, 70(3):197–211, 2006.

[23] Peter J. Burt and Edward H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, COM-31,4:532–540, 1983.

[24] S. X. Cadrin. Advances in morphometric identification of fishery stocks. *Reviews in Fish Biology and Fisheries*, 10(1):91–112, 2000.

[25] J. H. Challis. A procedure for determining rigid body transformation parameters. *Journal of Biomechanics*, 28:733–737, 1995.

[26] Siddhartha Chib and Edward Greenberg. Understanding the Metropolis-Hastings Algorithm. *The American Statistician*, 49(4):327–335, Nov 1995.

[27] Ronald Coifman and Stphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21:5–30, 2006.

[28] Timothy F. Cootes, G. Edwards, and C.J. Taylor. Comparing Active Shape Models with Active Appearance Models. In *Proceedings of the British Machine Vision Conference, BMVC 1999, University of Nottingham, September 13-16, 1999.*, pages 173–182. BMVA Press, 1999.

[29] Timothy F. Cootes, G. J. Edwards, and C. J. Taylor. Active Appearance Models. In *5th European Conference on Computer Vision*, volume 1407, pages 484–498, Berlin, 1998. Springer.

[30] Timothy F. Cootes, A. Hill, C. J. Taylor, and J. Haslam. Use of active shape models for locating structures in medical images. *Image and Vision Computing*, 12(6):355–365, 1994.

[31] Timothy F. Cootes and C. J. Taylor. Active Shape Models: Smart Snakes. In *British Machine Vision Conference*, pages 267–275, 1992.

[32] Timothy F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Training models of shape from sets of examples. In *Proc. British Machine Vision Conference*, pages 266–275, Berlin, 1992. Springer.

[33] Timothy F. Cootes and C.J. Taylor. Combining Point Distribution Models with Shape Models Based on Finite Element Analysis. *Image and Vision Computing*, 13:419–428, 1995.

[34] Timothy F. Cootes, C.J. Taylor, D. H. Cooper, and J. Graham. Active Shape Models-Their Training and Application. *Computer Vision and Image Understanding*, 61(1):38–59, January 1995.

[35] Isabelle Corouge, Sylvain Gouttard, and Guido Gerig. A statistical shape model of individual fiber tracts extracted from diffusion tensor MRI. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2004, 7th International Conference Saint-Malo, France, September 26-29, 2004, Proceedings, Part II*, volume 3, pages 671–679, 2004.

[36] Daniel Cremers, Mikael Rousson, and Rachid Deriche. A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape. *Int. J. Comput. Vision*, 72(2):195–215, 2007.

[37] T. M. Cronin. Visualizing concave and convex partitioning of 2d contours. *Pattern Recognition Letters*, 24(1-3):429–443, 2003.

[38] C. Davatzikos, X. Tao, and D. Shen. Hierarchical active shape models, using the wavelet transform. *IEEE Transactions on Medical Imaging*, 22(3):414–423, 2003.

[39] R. H. Davies, C. J. Twining, T. F. Cootes, J. C. Waterton, and C. J. Taylor. 3D Statistical Shape Models Using Direct Optimisation of Description Length. In *European Conference on Computer Vision 2002*, volume 3, pages 3–20, 2002.

[40] Thomas G. Dietterich. Isolated leaves dataset, oregon state university web resource, url: http://web.engr.oregonstate.edu/ tgd/leaves/, 2002.

[41] Katarina Domijan and Simon Wilson. A bayesian method for automatic landmark detection in segmented images. In *Proceedings of the workshop Machine Learning Techniques for Processing Multimedia Content, Bonn, Germany, 2005*, 2005.

[42] David L. Donoho and Carrie Grimes. Hessian Eigenmaps: New Locally Linear Embedding Techniques For High-Dimensional Data. Technical report, Department of Statistics, Stanford University, 2003.

[43] J. L. Doob. *Stochastic Processes*. John Wiley & Sons, New York, 1953.

[44] Ian L. Dryden. General Shape and Registration Analysis. In *Stochastic geometry: likelihood and computation*, pages 333–364. Chapman and Hall, 1999.

[45] Ian L. Dryden and Kanti V. Mardia. *Statistical Shape Analysis*. Wiley, Chichester, 1998.

[46] G. H. Dunteman. *Principal Component Analysis*, volume 14. Sage Publications, 1989.

[47] Gerald Edgar. *Measure, Topology, and Fractal Geometry*. Springer, second edition, 2008.

[48] Patrick Etyngier, Renaud Keriven, and Jean-Philippe Pons. Towards Segmentation Based on a Shape Prior Manifold. In *Scale Space and Variational Methods in Computer Vision*, pages 895–906, 2007.

[49] Kenneth Falconer. *Fractal Geometry: Mathematical Foundations and Applications*. John Wiley & Sons, Chichester, second edition, 1990.

[50] Jens Feder. *Fractals*. Plenum Press, New York, 1988.

[51] Richard P. Feynman. *The Feynman Lectures on Physics (volume 2)*. Addison-Wesley, Reading, Massachusetts, 1964.

[52] B. Fischl, A. Liu, and A. M. Dale. Automated manifold surgery: Constructing geometrically accurate and topologically correct models of the human cerebral cortex. *IEEE Transactions on Medical Imaging*, 2:70–80, 2001.

[53] S. L. Free, S. M. Sisodiya, M. J. Cook, D. R. Fish, and S. D. Shorvon. Three-dimensional fractal analysis of the white matter surface from magnetic resonance images of the human brain. *Cerebral Cortex*, 6(6):830–836, 1996.

[54] K. Fukunaga and W. L. G. Koontz. Applications of the Karhunen-Loeve expansion to feature selection and ordering. *IEEE Transactions on Computers*, C-19:311–318, 1970.

[55] G. David Garson. Factor analysis, url: http://www2.chass.ncsu.edu/garson/pa765/factor.htm, 2009.

[56] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall, London, 1995.

[57] J. J. Gerbrands. On the relationships between SVD, KLT and PCA. *Pattern Recognition*, 14(1-6):375–381, 1981.

[58] Guido Gerig, Martin Styner, Martha E. Shenton, and Jeffrey A. Lieberman. Shape versus size: Improved understanding of the morphology of brain structures. In Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen, editors, *Medical Image Computing and Computer-Assisted Intervention MICCAI*

2001, 4th International Conference Utrecht, The Netherlands, October 1417, 2001 Proceedings, volume 2208 of *Lecture Notes in Computer Science*, pages 24–32. Springer, 2001.

[59] Manel Giuli. Terra Virtual, url: http://personal.telefonica.terra.es/web/terravirtual, 2005.

[60] Colin Goodall. Procrustes Methods in the Statistical Analysis of Shape. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(2):285–339, 1991.

[61] John C. Gower. Generalized Procrustes Analysis. *Psychometrika*, 40:33–51, 1975.

[62] John C. Gower and Garmat B. Dijksterhuis. *Procrustes Problems*. Oxford Univeristy Press, 2004.

[63] U. Grenander and M. I. Miller. Computational anatomy: An emerging discipline. *Quarterly of Applied Mathematics*, 56:617–694, 1998.

[64] Geoffrey R. Grimmett and David R. Stirzaker. *Probability and Random Processes*. Oxford Science Publications, second edition, 1992.

[65] D.J. Hawkes, D. Barratt, J.M. Blackall, C. Chan, P.J. Edwards, K. Rhode, G.P. Penney, J. McClelland, and D.L.G. Hill. Tissue deformation and shape models in image-guided interventions: a discussion paper. *Medical Image Analysis*, 9(2):163–175, 2005.

[66] Xiaofei He, Wei-Ying Ma, and Hong Jiang Zhang. Learning an image manifold for retrieval. In *Proceedings of the 12th ACM International Conference on Multimedia*, pages 17–23. ACM, 2004.

[67] B. Heidari, F. Madeh Khaksar, and D. FitzPatrick. Automatic Landmark Detection on Epicondyles of Distal Femur in X-Ray Images. In *4th European Conference of the International Federation for Medical and Biological Engineering*, volume 22, pages 533–536. Springer, 2009.

[68] Tobias Heimann and Hans-Peter Meinzer. Statistical shape models for 3D medical image segmentation: A review. *Medical Image Analysis*, 13(4):543–563, August 2009.

[69] Andrew Hill, Christopher J. Taylor, and Alan D. Brett. A Framework for Automatic Landmark Identification Using a New Method of Nonrigid Correspondence. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(3):241–251, 2000.

[70] Paul G. Hogel, Sidney C. Port, and Charles J. Stone. *Introduction to Statistical Theory*. Houghton, Boston, first edition, 1971.

[71] Heike Hufnagel, Xavier Pennec, Jan Ehrhardt, Heinz Handels, and Nicholas Ayache. Point-Based Statistical Shape Models with Probabilistic Correspondences and Affine EM-ICP. In Alexander Horsch, Thomas Martin Deserno, Heinz Handels, Hans-Peter Meinzer, and Thomas Tolxdorff, editors, *Bildverarbeitung für die Medizin 2007, Algorithmen, Systeme, Anwendungen, Proceedings des Workshops vom 25.-27. März 2007 in München*, pages 434–438. Springer, 2007.

[72] Camille Izard, Bruno Jedynak, and Craig E. L. Stark. Spline-Based Probabilistic Model for Anatomical Landmark Detection. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2006, 9th International Conference, Copenhagen, Denmark, October 1-6, 2006, Proceedings, Part I*, volume 4190 of *Lecture Notes in Computer Science*, pages 849–856. Springer, 2006.

[73] Keith A. Johnson. Neuroimaging primer, url: http://www.med.harvard.edu/aanlib/hms1.html, 1999.

[74] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, second edition, 1986.

[75] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active Contour Models. *International Journal of Computer Vision*, 1(4):321–331, 1987.

[76] Hannu Kauppinen, Tapio Seppänen, and Matti Pietikäinen. An Experimental Comparison of Autoregressive and Fourier-Based Descriptors in 2D Shape Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):201–207, 1995.

[77] D. G. Kendall, D. Barden, T. K. Carne, and H. Le. *Shape and Shape Theory*. Wiley, Chichester, 1999.

[78] Jeong-Sik Kim, Soo-Mi Choi, Yoo-Joo Choi, and Myoung-Hee Kim. Global and Local Shape Analysis of the Hippocampus Based on Level-of-Detail Representations. In Jun Zhang, Ji-Huan He, and Yuxi Fu, editors, *Computational and Information Science, First International Symposium, CIS 2004, Shanghai, China, December 16-18, 2004, Proceedings*, volume 3314 of *Lecture Notes in Computer Science*, pages 504–509. Springer, 2004.

[79] V. G. Kiselev, K. R. Hahn, and D. P. Auer. Is the brain cortex a fractal? *NeuroImage*, 20(3):1765–1774, 2003.

[80] J. Koikkalainen, T. Tolli, K. Lauerma, K. Antila, E. Mattila, M. Lilja, and J. Lotjonen. Methods of artificial enlargement of the training set for statistical shape models. *IEEE Trans. Medical Imaging*, 27(11):1643–1654, November 2008.

[81] D. Kontos, V. Megalooikonomou, N. Ghubade, and C. Faloutsos. Detecting discriminative functional mri activation patterns using space filling curves. *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, 1:963–966, 2003.

[82] Steven H. Koslow and Shankar Subramaniam. *Databasing the Brain: From Data to Knowledge (Neuroinformatics)*. Wiley, 2005.

[83] Stéphane Lafon and Ann B. Lee. Diffusion Maps and Coarse-Graining: A Unified Framework for Dimensionality Reduction, Graph Partitioning, and Data Set Parameterization. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 28(9):1393–1403, September 2006.

[84] Peter M. Lee. *Bayesian Statistics*. Arnold, London, third edition, 2004.

[85] Subhash R. Lele and Joan T. Richtsmeier. *An Invariant Approach to Statistical Analysis of Shapes*. Chapman and Hall/CRC, 2001.

[86] Tony Lindeberg. Scale-space. In *Encyclopedia of Computer Science and Engineering*, 2008.

[87] Seymour Lipschutz and Marc Lipson. *Linear Algebra*. Schaum's Outline Series. McGraw-Hill, New York, NY, USA, 3d edition, 2001.

[88] J. Z. Liu, L. D. Zhang, and G. H. Yue. Fractal dimension in human cerebellum measured by magnetic resonance imaging. *Biophysical Journal*, 85(6):4041–4046, 2003.

[89] G. P. Lohmann. Eigenshape analysis of microfossils: A general morphometric procedure for describing changes in shape. *Mathematical Geology*, 15(6):659–672, 1984.

[90] Sven Loncaric. A Survey of Shape Analysis Techniques. *Pattern Recognition*, 31:983–1001, 1998.

[91] Jyrki Lötjönen, Kari Antila, E. Lamminmäki, Juha Koikkalainen, Mikko Lilja, and Timothy F. Cootes. Artificial enlargement of a training set for statistical shape models: Application to cardiac images. In Alejandro F. Frangi, Petia Radeva, Andrés Santos, and Monica Hernandez, editors, *Functional Imaging and Modeling of the Heart, Third International Workshop, FIMH 2005, Barcelona, Spain, June 2-4, 2005, Proceedings*, volume 3504 of *Lecture Notes in Computer Science*, pages 92–101. Springer, 2005.

[92] J. P. Louvet. Fractals, url: http://fractals.iut.u-bordeaux1.fr/jpl/jpl01a.html, 2003.

[93] Yi Lu, Steven Schlosser, and Michael Janeczko. Fourier descriptors and handwritten digit recognition. *Machine Vision and Applications*, 6(1):25–34, 1993.

[94] E. Luders, K. L. Narr, P. M. Thompson, D. E. Rex, L. Jancke, H. Steinmetz, and A. W. Toga. Gender differences in cortical complexity. *Nature Neuroscience*, 7(8):799–800, 2004.

[95] N. MacLeod. Generalizing and extending the eigenshape method of shape space visualization and analysis. *Paleobiology*, 25(1):107–138, 1999.

[96] N. MacLeod. Geometric morphometrics and geological shape-classification systems. *Earth-Science Reviews*, 59(1-4):27–47, 2002.

[97] N. Macleod and K. D. Rose. Inferring locomotor behavior in paleogene mammals via eigenshape analysis. *American Journal of Science*, 293-A:300–355, 1993.

[98] B. B. Mandelbrot. *The fractal geometry of nature*. Freeman, Oxford, 1982.

[99] B. B. Mandelbrot, P. Pfeifer, O. Biham, O. Malcai, D. A. Lidar, and D. Avnir. Is nature fractal? *Science*, 279(5352):783–786, 1998.

[100] P. Manousopoulos, V. Drakopoulos, and T. Theoharis. Fractal Active Shape Models. In *Proceedings of the 12th International Conference on Computer Analysis of Images and Patterns, CAIP 2007, Vienna, August 2007.*, pages 645–652, 2007.

[101] P. Manousopoulos, V. Drakopoulos, and T. Theoharis. Curve fitting by fractal interpolation. In Marina Gavrilova and C. J. Kenneth Tan, editors, *Transactions on Computational Science I*, volume 4750 of *Lecture Notes in Computer Science*, pages 85–103. Springer Berlin / Heidelberg, 2008.

[102] Kantilal Vardichand Mardia, John T. Kent, and John M. Bibby. *Multivariate Analysis*. Probability and Mathematical Statistics. Academic Press, London, 1979.

[103] David Marr. Early processing of visual information. *Philosophical Transactions of the Royal Society of London*, B-275:483–524, 1976.

[104] Wendy L. Martinez and Angel R. Martinez. *Exploratory Data Analysis with MATLAB*. Chapman & Hall/CRC, 2004.

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

[106] G. McLachlan and D. Peel. *Finite Mixture Models*. Wiley, 2000.

[107] Yang Mingqiang, Kpalma Kidiyo, and Ronsin Joseph. A Survey of Shape Feature Extraction Techniques. In Peng-Yeng Yin, editor, *Pattern Recognition*, pages 43–90. IN-TECH, 2008.

[108] F. Mokhtarian. Silhouette-based occluded object recognition through curvature scale space. *Machine Vision and Applications*, 10:87–97, 1997.

[109] F. Mokhtarian and S. Abbasi. Affine curvature scale space with affine length parametrisation. *Pattern Analysis and Applications*, 4(1):1–8, 2001.

[110] F. Mokhtarian, S. Abbasi, and J. V. Kittler. Robust and efficient shape indexing through curvature scale space. In *Proc. British Machine Vision Conference*, pages 53–62, 1996.

[111] F. Mokhtarian and A. Mackworth. Scale base description and recognition of planar curves and two-dimensional shapes. *IEEE PAMI*, 8:34–43, 1986.

[112] F. Mokhtarian and A. K. Mackworth. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(8):789–805, aug 1995.

[113] Farzin Mokhtarian and Miroslav Bober. *Curvature Scale Space Representation: Theory, Applications, and MPEG-7 Standardization*. Kluwer Academic Publishers, Norwell, MA, USA, 2003.

[114] Farzin Mokhtarian, Nasser Khalili, and Peter Yuen. Estimation of error in curvature computation on multi-scale free-form surfaces. *International Journal of Computer Vision*, 48(2):131–149, 2002.

[115] Farzin Mokhtarian, Yoke Khim Ung, and Zhitao Wang. Automatic fitting of digitised contours at multiple scales through the curvature scale space technique. *Computers & Graphics*, 29(6):961–971, 2005.

[116] D. F. Morrison. *Multivariate statistical methods*. McGraw-Hill, London, second edition, 1976.

[117] J. Neumann, H. Samet, and A. Soffer. Integration of Local and Global Shape Analysis for Logo Classification. *Pattern Recognition Letters*, 23(12):1449–1457, Oct 2002.

[118] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849–856. MIT Press, 2001.

[119] T.F. Nonnenmacher, G.A. Losa, and E.R. Weibel. *Fractals in Biology and Medicine*. Birkhauser Verlag, Boston, 1993.

[120] Petrus Johannes Van Otterloo. *A Contour-Oriented Approach to Shape Analysis*. Prentice Hall, 1991.

[121] Sasirekha Palaniswamy, Neil A. Thacker, and Christian Peter Klingenberg. Automatic identification of morphometric landmarks in digital images. In *British Machine Vision Conference*, pages 870–879, 2007.

[122] Theodosios Pavlidis. A review of algorithms for shape analysis. *Computer Graphics and Image Processing*, 7:243–258, 1978.

[123] M. Peura and J. Iivarinen. Efficiency of simple shape descriptors. In *VF97*, pages 443–451, 1997.

[124] S. M. Pizer, P. T. Fletcher, S. C. Joshi, A. Thall, J. Z. Chen, Y. Fridman, D. S. Fritsch, A. G. Gash, J. M. Glotzer, M. R. Jiroutek, C. L. Lu, K. E. Muller, G. Tracton, P. A. Yushkevich, and E. L. Chaney. Deformable M-reps for 3D medical image segmentation. *International Journal of Computer Vision*, 55(2-3):85–106, November 2003.

[125] S. M. Pizer, D. S. Fritsch, P. A. Yushkevich, V. E. Johnson, and E. L. Chaney. Segmentation, registration, and measurement of shape variation via image object shape. *IEEE Transactions on Medical Imaging*, 18(10):851–865, October 1999.

[126] Nasir. M. Rajpoot, Muhammad. Arif, and Abhir. H. Bhalerao. Unsupervised learning of shape manifolds. In *British Machine Vision Conference*, pages 312–321, 2007.

[127] A. Rao, P. Aljabar, and D. Rueckert. Hierarchical statistical shape analysis and prediction of sub-cortical brain structures. *Medical Image Analysis*, 12:55–68, 2008.

[128] R. A. Reyment. Multivariate morphometrics and analysis of shape. *Journal - International Association for Mathematical Geology*, 17(6):591–609, 1985.

[129] W. Richards and D. D. Hoffman. Codon constraints on closed 2-d shapes. *Computer Vision, Graphics, and Image Processing*, 31(3):265–281, 1985.

[130] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. Springer, Berlin, 1999.

[131] F. J. Rohlf and L. F. Marcus. A revolution in morphometrics. *Trends in Ecology and Evolution*, 8(4):129–132, 1993.

[132] Amy Ross. Procrustes Analysis. Technical report, Department of Computer Science and Engineering, University of South Carolina, SC 29208, 2004.

[133] Sam T. Roweis and Lawrence K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2326, 2000.

[134] Todd Rowland. Manifold - from mathworld–a wolfram web resource, url: http://mathworld.wolfram.com/manifold.html, 2009.

[135] Sylvia Rueda and Mariano Alcañiz Raya. An Approach for the Automatic Cephalometric Landmark Detection Using Mathematical Morphology and Active Appearance Models. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2006, 9th International Conference, Copenhagen, Denmark, October 1-6, 2006, Proceedings, Part I*, volume 4190 of *Lecture Notes in Computer Science*, pages 159–166. Springer, 2006.

[136] Hans Sagan. *Space-Filling Curves*. Springer-Verlag, New York, NY, 1994.

[137] Cesare Sangiorgi, Andrew C.Collop, and Nicholas H.Thom. A non-destructive impulse hammer for evaluating the bond between asphalt layers in a road pavement. In *Proceedings of the International Symposium on Non-Destructive Testing in Civil Engineering (NDT-CE)*, 2003.

[138] Lawrence K. Saul and Sam T. Roweis. Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds. *Journal of Machine Learning Research*, 4:119–155, 2003.

[139] K. Schaefer, P. Mitteroecker, P. Gunz, M. Bernhard, and F. L. Bookstein. Craniofacial sexual dimorphism patterns and allometry among extant hominids. *Annals of Anatomy*, 186(5-6):471–478, 2004.

[140] J. A. Schnabel and S. R. Arridge. Active Shape Focusing. *Image and Vision Computing*, 17(5-6):419–429, 1999.

[141] Daniel Sharvit, Jacky Chan, Huseyin Tek, and Benjamin B. Kimia. Symmetry-based indexing of image databases. *Journal of Visual Communication and Image Representation*, 9:366–380, 1998.

[142] D. Shen, E. H. Herskovits, and C. Davatzikos. An adaptive-focus statistical shape model for segmentation and shape modeling of 3-d brain structures. *IEEE Transactions on Medical Imaging*, 202:257–270, 2001.

[143] D. Shen, E. H. Herskovits, and C. Davatzikos. An adaptive-focus statistical shape model for segmentation and shape modeling of 3-D brain structures. *IEEE Transactions on Medical Imaging*, 202(4):257–270, 2001.

[144] Dennis E. Slice, Fred L. Bookstein, F. James Rohlf, and Leslie F. Marcus. A Glossary for Geometric Morphometrics, url: http://life.bio.sunysb.edu/morph/glossary/gloss1.html, 1998.

[145] C. G. Small. *The Statistical Theory of Shape*. Springer, New York, 1996.

[146] Lindsay I. Smith. A tutorial on Principal Components Analysis, url: http://www.cs.otago.ac.nz/cosc453/, 2002.

[147] T. G. Smith and T. N. Behar. Comparative fractal analysis of cultured glia derived from optic nerve and brain demonstrate different rates of morphological differentiation. *Brain Research*, 634(2):181–190, 1994.

[148] Richard Socher. Manifold Learning and Dimensionality Reduction with Diffusion Maps. Seminar report, Saarland University, 2008.

[149] Z. Soltys, M. Ziaja, R. Pawlinski, Z. Setkowicz, and K. Janeczko. Morphology of reactive microglia in the cerebral cortex: Fractal analysis and complementary quantitative methods. *Journal of Neuroscience Research*, 63(1):90–97, 2001.

[150] Richard Souvenir and Robert Pless. Manifold Clustering. In *International Conference on Computer Vision*, pages 648–653, 2005.

[151] G. Strang. *Linear Algebra and Its Applications*. Harcourt Brace Jovanovich, Orlando, FL, third edition, 1988.

[152] M. Styner, G. Gerig, J. Lieberman, D. Jones, and D. Weinberger. Statistical Shape Analysis of Neuroanatomical Structures Based on Medial Models. *Medical Image Analysis*, 7:207–220, 2003.

[153] Martin A. Styner, I. Oguz, S. Xu, C. Brechbhler, D. Pantazis, J.J. Levitt, M.E. Shenton, and G. Gerig. Framework for the statistical shape analysis of brain structures using SPHARM-PDM. In *Open Science Workshop at MICCAI*, 2006.

[154] Martin A. Styner, Kumar T. Rajamani, Lutz peter Nolte, Gabriel Zsemlye, Gabor Szekely, Chris J. Taylor, and Rhodri H. Davies. Evaluation of 3d correspondence methods for model building. In Christopher J. Taylor and J. Alison Noble, editors, *Information Processing in Medical Imaging, 18th International Conference, IPMI 2003, Ambleside, UK, July 20-25, 2003, Proceedings*, volume 2732 of *Lecture Notes in Computer Science*, pages 63–75. Springer, 2003.

[155] Jon Sullivan. PDPhoto: Free Public Domain Photo Database, url: http://pdphoto.org/, 2009.

[156] Cho-Huak Teh and Roland T. Chin. On the Detection of Dominant Points on Digital Curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-11(8):859–872, 1989.

[157] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, 2000.

[158] D. W. Thompson. *On Growth and Form*. Dover, 1917.

[159] P. Thompson and A. Toga. A framework for computational anatomy, 2002.

[160] P. M. Thompson, C. Schwartz, R. T. Lin, A. A. Khan, and A. W. Toga. Three-dimensional statistical analysis of sulcal variability in the human brain. *Journal of Neuroscience*, 16(13):4261–4274, 1996.

[161] P. M. Thompson, R. P. Woods, M. S. Mega, and A. W. Toga. Mathematical/computational challenges in creating deformable and probabilistic atlases of the human brain. *Human Brain Mapping*, 9(2):81–92, 2000.

[162] Elke Thönnes. *Monte Carlo methods*. Department of Statistics, University of Warwick, 2008.

[163] A. Tsai, A. Willsky, W. Wells, E. Grimson, and C. Tempany. Mutual information in coupled multi-shape model for medical image segmentation. *Medical Image Analysis*, 8(4):429–445, 2004.

[164] T. Ubukata. A morphometric study on morphological plasticity of shell form in crevice-dwelling pterioida (bivalvia). *Biological Journal of the Linnean Society*, 79(2):285–297, 2003.

[165] Ferdinand van der Heijden, Robert Duin, Dick de Ridder, and David M. J. Tax. *Classification, Parameter Estimation and State Estimation: An Engineering Approach Using MATLAB*. John Wiley & Sons, Chichester, 2004.

[166] B. C. Vemuri and A. Radisavljevic. Multiresolution stochastic hybrid shape models with fractal priors. *ACM Transactions on Graphics*, 13(2):177–207, 1994.

[167] Jingnan Wang, A. Ekin, and G. de Haan. Shape analysis of brain ventricles for improved classification of Alzheimer's patients. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 2252–2255, 2008.

[168] Y. Wang and E. K. Teoh. 2D Affine-Invariant Contour Matching Using B-Spline Model. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(10):1853–1858, October 2007.

[169] Yongmei Wang and Lawrence H. Staib. Boundary Finding with Prior Shape and Smoothness Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:738–743, 2000.

[170] Eric W. Weisstein. Fractal - from mathworld–a wolfram web resource, url: http://mathworld.wolfram.com/fractal.html, 2009.

[171] Bruce J. West. *Fractal Physiology and Chaos in Medicine*. World Scientific, Singapore, 1990.

[172] Andrew P. Witkin. Scale-Space Filtering. In *8th Int. Joint Conf. Artificial Intelligence*, volume 2, pages 1019–1022, 1983.

[173] Todd Wittman. MANI fold Learning Matlab Demo, url: http://www.math.umn.edu/ wittman/mani/.

[174] Daniel M. Wuescher and Kim L. Boyer. Robust contour decomposition using a constant curvature criterion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(1):41–51, 1991.

[175] Rui Xu and Don Wunsch. *Clustering*. Wiley, New Jersey, 2008.

[176] H. Xue, L. Srinivasan, S. Jiang, M. Rutherford, A. D. Edwards, D. Rueckert, and J. V. Hajnal. Automatic segmentation and reconstruction of the cortex from neonatal MRI. *NeuroImage*, 38:461–477, 2007.

[177] Dragomir Yankov and Eamonn J. Keogh. Manifold Clustering of Shapes. In *ICDM*, pages 1167–1171. IEEE Computer Society, 2006.

[178] Peng Yu, P. Ellen Grant, Yuan Qi, Xiao Han, Florent Ségonne, R. Pienaar, E. Busa, J. Pacheco, N. Makris, Randy L. Buckner, Polina Golland, and Bruce Fischl. Cortical Surface Shape Analysis Based on Spherical Wavelets. *IEEE Trans. Medical Imaging*, 26(4):582–597, April 2007.

[179] Paul Yushkevich, Stephen M. Pizer, Sarang Joshi, and J. S. Marron. Intuitive, Localized Analysis of Shape Variability. In Michael F. Insana and Richard M. Leahy, editors, *17th International Conference Information Processing in Medical Imaging, IPMI 2001 Davis, CA, USA, June 1822, 2001 Proceedings*, volume 2082 of *Lecture Notes in Computer Science*, pages 402–408. Springer, 2001.

[180] Lihi Zelnik-Manor and Pietro Perona. Self-Tuning Spectral Clustering. In *NIPS*, 2004.

[181] Dengsheng Zhang and Guojun Lu. A Comparative Study on Shape Retrieval Using Fourier Descriptors with Different Shape Signatures. In *Proceedings of International Conference on Intelligent Multimedia and Distance Education (ICIMADE01)*, pages 1–9, 2001.

[182] Dengsheng Zhang and Guojun Lu. A comparison of Shape Retrieval Using Fourier Descriptors and Short-Time Fourier Descriptors. In *Proceedings of the Second IEEE Pacific Rim Conference on Multimedia*, pages 855–860, London, UK, 2001. Springer-Verlag.

[183] L. Zhang, J. Z. Liu, D. Dean, V. Sahgal, and G. H. Yue. A three-dimensional fractal analysis method for quantifying white matter structure in human brain. *Journal of Neuroscience Methods*, 150(2):242–253, 2006.

[184] Lisha Zhang, Manuel Jo ao da Fonseca, and Alfredo Ferreira. Survey on 3D Shape Descriptors. Technical report, Project DecorAR, Fundação para a Ciência e Tecnologia, 2004.

[185] Zheen Zhao, Stephen R. Aylward, and Eam Khwang Teoh. A novel 3D Partitioned Active Shape Model for Segmentation of Brain MR Images. In James S. Duncan and Guido Gerig, editors, *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2005, 8th International Conference, Palm Springs, CA, USA, October 26-29, 2005, Proceedings, Part I*, volume 3749 of *Lecture Notes in Computer Science*, pages 221–228. Springer, 2005.

[186] Zheen Zhao and Eam Khwang Teoh. A novel framework for automated 3D PDM construction using deformable models. volume 5747, pages 303–314. SPIE, 2005.