BUILDING OPTIMAL 3D SHAPE MODELS

Allan Reinhold Kildeby

LYNGBY 2002
MASTERS THESIS
IMM-2002-62

IMM

Printed by IMM, Technical University of Denmark

Dedicated to my wife, Martine.

Preface

This project has been prepared at the Section for Image Analysis, Department of Informatics and Mathematical Modelling, IMM, at The Technical University of Denmark, in partial fulfillment of the requirements for the degree Master of Science in Engineering, M.Sc.Eng.

The general framework of the thesis is statistics and shape analysis. It is implied that the reader has a basic knowledge within these areas and is familiar with concepts such as deformable template models and minimum description length, MDL.

Lyngby, September 2002

Allan Reinhold Kildeby

Acknowledgements

A considerable part of the work reported in this project could not have been completed without the support and encouragement from colleges and friends. For this effort I would like to thank the following.

Ph.D. student Rasmus Reinhold Paulsen, Oticon A/S for helping me getting started on the project, constant encouragements, mathematical guidance and for keeping the mobile phone line continuously open.

Ph.D. student Rhodri Davies, Imaging Science and Biomedical Engineering, Faculty of Medicine, University of Manchester for useful hints and explanations on the subject of this thesis.

Ph.D. student Henrik Aanæs, Section of Image Analysis, IMM for helping me solve key geometrical problems.

Ph.D. student Mikkel B. Stegmann, Section of Image Analysis, IMM for guidance on development issues.

Dr. Klaus Baggesen Hilger, Section of Image Analysis, IMM for fruitful discussions on mathematical subjects.

Dr. Rasmus Larsen, Section of Image Analysis, IMM for guidance and knowledge in the field of statistical image analysis, and for being my mentor during the project.

On a personal note I would like to thank my good friend Andreas Sand for dragging me onto the golf course and getting my mind off project work in times of despair.

Finally I would like to send my heartful thoughts to my wife Martine, without whos understanding, patience and love, this project would never have succeeded.

Abstract

This thesis presents a general approach towards automated 3D statistical shape model building through the utilization of spherical mapping and Minimum Description Length, based on algorithms proposed by Angenent et al. [1] and Davies et al. [14].

A thorough treatment and discussion of the theoretic foundation involved in conformally mapping 3D surfaces to the unit sphere is given. The basic algorithm is extended through the imposing of an area-preservation criteria.

The theoretical foundation behind Minimum Description Length shape modelling is presented and discussed, followed by several extensions to the basic algorithm. Extensions include employment of the spherical map derived, robust landmark positioning and simplification of objective function, all of which have been included in a high performance C++ framework.

Experimental results on both synthetical and biological training data reveal the potential of and difficulties in composing unique spherical maps as well as in building a fully automated shape model, while retaining specificity, generality and compactness.

It is concluded that automated statistical shape learning successfully can accomplish compact and general shape models, through the use of spherical maps, though this approach to automated 3D model building is still fairly unexplored.

Keywords: Deformable Template Models, Point Distribution Models, Principal Component Analysis, Shape Analysis, Shape Alignment, Finite Elements Models, Spherical Mapping, Conformal Mapping, Area-preserving Mapping, Minimum Description Length, Automated Shape Learning.

Resumé

I denne afhandling præsenteres en generel metode til automatiseret etablering af statistiske formmodeller, under anvendelse af sfæriske afbildninger og Minimum Description Length. Metoden er baseret på algoritmer præsenteret af Angenent et al. [1] samt Davies et al. [14].

Der gives en grundig introduktion til og diskussion af det teoretiske fundament for konforme afbildninger af 3D former på enhedskuglen. Algoritmen udvides ved tilføjelsen af et areal-bevarende kriterie.

Den teoretiske baggrund for Minimum Description Length formmodellering præstenteres og diskuteres, efterfulgt af en række udvidelser til formuleringen. Disse udvidelser omfatter integration af den udledte arealbevarende afbildning, robust landmark positionering samt simplificering af objekt funktion, alt sammen inkluderet i et højtydende C++ bibliotek.

Eksperimentielle resultater, for både syntetiske såvel som biologiske former, afslører potentialer og svagheder i konstruktionen af unikke sfæriske afbildninger, såvel som i automatiseret konstruktion af formmodeller der samtidig skal være både specifikke, generelle og kompakte.

Det konkluderes, at automatiseret statistisk formindlæring succesfuldt kan opnå generelle og kompakte modeller, under anvendelsen af sfæriske afbildninger, på trods af at denne tilgang til automatiseret formindlæring stadig er relativt uudforsket.

Nøgleord: Deformable Template Models, Point Distribution Models, Principal komponent analyse, Formanalyse, Formregistrering, Finit-element modeller, Sfærisk afbildning, Konform afbildning, Arealbevarende afbildning, Minimum Description Length, Automatiseret formindlæring.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"Theory attracts practice as the magnet attracts iron."
– Karl Friedrich Gauss

This thesis documents a study on core problems within 3D statistical shape modelling, namely automated building of corresponding shape representations.

Common for deformable template models is the use of prior knowledge retrieved from statistical analysis of shape and in some cases texture parameters. Prior to such analysis, a set of corresponding common variables must be established between all members of the data set. Generally the acquisition of such landmarks is a cumbersome and tedious task, and will in most cases require an educated specialist within the specific field of application.

Throughout the last decade, a vast array of 2D shape modelling and automated model building techniques has emerged. As computational power has increased, the application horizon is broadened thus 3D shape models have been the next logical step in the evolution of statistical shape modelling. However the concurrent increase in complexity involved in landmarking large data sets by hand is seldom linear, thus the need for robust, efficient and compact automated model building has become an inherent requirement.

1.1 Motivation and Objectives

The Minimum Description Length shape model was proposed by Davies et al. [12] in 2001 as a sophisticated automated model building approach, capable of solving the ever present correspondence problem. Relying on spherical mapping of a training set, the model constitutes a novel approach on statistical shape learning. Due to this fact, and the overall elegance of such combination of techniques, work in this area constituted a suitable and challenging subject for a master thesis. Thus the main objectives set forth were:

- Discuss, document and explore the spherical mapping and MDL approaches to shape modelling.
- Design extensions to the original algorithms.
- Evaluate the spherical mapping and MDL approaches through relevant examples.

As an additional objective the solution aimed at providing a platform for further development in C++.

1.2 Thesis Overview

The thesis is divided into 4 parts, each requiring knowledge from the preceding parts.

Part I: Theory. Presenting the mathematical foundation of spherical mapping and the MDL approach, with emphasis on adjustments and extensions.

Part II: Implementation. Walk-through of development details regarding the resulting SM^2-API.

Part III: Experimental Results. Evaluating performance of algorithms and implementation.

Part IV: Discussion. Discussion of the achieved results and presentation of possible new extensions.

In addition to the above listed parts, the report is supplemented by appendices on programmatic issues as well as detailed results.

1.3 Mathematical Notation

Mathematical notations conform to the following conventions.

Scalars are typeset in italic lowercase.

$$s = 1.0$$

Scalar functions are typeset in italic lowercase.

$$f(s) = s - s$$

Vectors are typeset in non-italic bold lowercase.

$$\mathbf{v} = [a, b, c]^T$$

Vector functions are typeset in non-italic bold lowercase.

$$\mathbf{f}(\mathbf{v}) = \mathbf{v} - \mathbf{v}$$

Matrices are typeset in non-italic bold capitals.

$$\mathbf{M} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Matrix functions are typeset using bold non-italic capitals.

$$\mathbf{f}(\mathbf{M}) = \mathbf{M} - \mathbf{M}$$

Part I

Theory

Chapter 2

Spherical Mapping

"All human knowledge thus begins with intuitions, proceeds thence to concepts, and ends with ideas."
– Emmanual Kant

2.1 Overview

In the following chapter, an approach to transformation of complicated 3D shapes into simpler 2D representations using conformal mapping is given. By mapping into 2D coordinates, the parameter requirement needed to represent each shape in a data set is reduced, since each coordinate can be represented by latitude and longitude.

Various approaches on deriving a spherical map have been presented [1], [5], [48]. As an example, Brechbühler et al. [5] employs a similar approach to the one described in the following chapter. By posing the problem as one of Dirichlets boundary conditions, i.e. with a known solution to the mapping function on the boundary, a Laplacian equation of the form $\nabla^2 f = 0$ is derived. The problem is solved using a finite differences method. A similar approach is known from heat conduction problems, where the temperature of some object is known at at fixed number of points, and the task here being to derive a complete map of temperatures across the object in question.

The algorithm proposed in this chapter is based primarily on the work conducted in the field of spherical mapping by Angenent et al. For further details on derivation of equations on conformal maps and finite elements, please refer to [1].

An effort has been put into making treatment of key concepts rich on examples.

2.2 Conformal Maps

2.2.1 Overview

Consider the mapping function

$$w = f(z) \quad , \quad f : D \to D \tag{2.1}$$

of a complex variable z denoted by

$$z = x + iy \tag{2.2}$$

If the function is defined in the z-plane of domain D, then each point will correspond to a point in the w-plane, denoted by

$$w = u + iv \tag{2.3}$$

In this sense there is a mapping between the complex z- and w-planes of domain D. If the mapping preserves angles between oriented curves in magnitude as well as sense, the mapping is conformal [30]. In other words $f(z)$ is conformal in a point z_0 if

$$f'(z_0) \neq 0 \tag{2.4}$$

This means, that the angle between oriented tangents in $f(z_0)$ equals the angle between corresponding curves in z_0. Furthermore the mapping is one-to-one or injective if different points in the z-plane corresponds to different points in the w-plane.

> **Definition 2.2.1.** *A conformal map preserves measures of angles between oriented curves at all but a finite number of points.*

As stated in definition 2.2.1, a map is conformal except at critical points, where $f'(z) = 0$. Such is the case with $f(z) = z^2$ where $f'(z) = 2z \neq 0$ for all $z \neq 0$.

If a map is conformal in all points, as is the case with $f(z) = e^z$ where $f'(z) = e^z \neq 0$ for all z, we say the map is conformal "everywhere".

The true value of conformal mapping lies in the ability to transform complicated boundary conditions into simpler boundaries when solving partial differential equations [19].

2.2.2 Uniformization

Riemanns mapping theorem in complex surface geometry states, that a surface of genus 0, i.e. a surface with no self-intersections or holes, can be conformally mapped to the unit sphere [29]. This fact, though enforcing a requirement on shapes to be topologically equivalent with a sphere, forms the base of solving the spherical mapping problem.

Keeping the above in mind, coordinates (x, y, z) of any given 3D shape can be represented using a complex z-coordinate as $(x, y, x + iy)$. This allows the shape to be interpreted as a *Riemann surface*, hence proving the existence of a spherical conformal map.

Let $\Sigma \subset \mathbb{R}^3$ be the surface of any given shape with topological equivalence with a sphere, and let p be a fixed point on Σ. Let S^2 denote the unit sphere in $\mathbb{R}^3$, and let N be the north pole. Using this notation, a conformal map can be expressed as:

$$f : \Sigma \backslash \{p\} \rightarrow S^2 \backslash \{N\} \tag{2.5}$$

Furthermore the partial differential equation, PDE

$$\Delta z = \left(\frac{\partial}{\partial u} - i \frac{\partial}{\partial v} \right) \delta_p \tag{2.6}$$

where (u, v) are conformal coordinates on Σ and δ_p is Dirac's delta, is a solution to the mapping problem. Since the solution to (2.6) is defined explicitly on the boundary, a smooth unit sphere, of domain Σ, the equation forms a *Dirichlet problem* [30].

Though (2.6) is defined on smooth manifolds, an approximation is necessary to solve the discrete problem numerically.

2.2.3 Finite Element Approximation

Overview

The concept of *finite element models* or FEM is an approach to solve boundary conditioned partial differential equations, by approximation of local regions with piecewise linear functions, thus transforming the problem into a system of ordinary linear equations. In the case of conformally mapping 3D surfaces to the unit sphere, solving the continuous mapping problem would not be feasible due to immense computational requirements.

A basic idea behind FEM is to model a given system as a physical problem [24]. This means setting up physical rules such as stiffness, forces and restitution coefficients, a set of properties defined in the specific environment. In general a finite element system can be written as:

$$\mathbf{K}\mathbf{d} = \mathbf{f} \tag{2.7}$$

where $\mathbf{K}$ is the symmetric stiffness matrix, $\mathbf{f}$ is the force vector and $\mathbf{d}$ the displacement vector. Symmetry in $\mathbf{K}$ exists since the matrix expresses element-to-element interactions, thus $\mathbf{K}_{a,b} = \mathbf{K}_{b,a}$. Assuming the inverse $\mathbf{K}^{-1}$ exists, the obvious solution to (2.7) is just $\mathbf{d} = \mathbf{K}^{-1}\mathbf{f}$. However calculating the inverse is only feasible if the modelled system has a low number of degrees of freedom. In most cases the FEM system can be solved by employing an iterative approach, such as the *conjugate gradient* algorithm [37].

Approximation of Conformal Map

If Δz is the set of piecewise linear functions on domain Σ, denoted by $PL(\Sigma)$, then for any function f, smooth in the neighborhood of a point p, it is given that

$$\int\int_{\Sigma} f\left(\frac{\partial}{\partial u} - i\frac{\partial}{\partial v}\right)\delta_p dS = -\left(\frac{\partial f}{\partial u} - i\frac{\partial f}{\partial v}\right)|_p \tag{2.8}$$

Let $\sigma = ABC$ be the triangle in whose interior p lies. D is the orthogonal projection of C on AB and f_x is the force contribution acting on vertex x in σ.

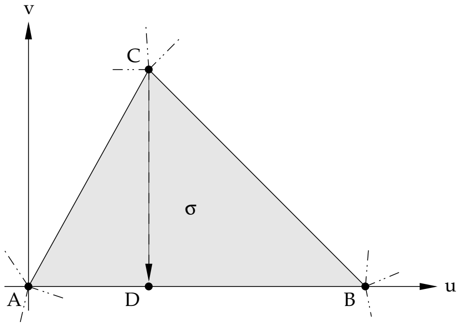


Figure 2.1: Approximating $\frac{\partial f}{\partial u}$ and $\frac{\partial f}{\partial v}$ on triangle σ in triangulated domain Σ.

$\frac{\partial f}{\partial u}$ respectively $\frac{\partial f}{\partial v}$ are the derivatives along the u- and v-axis. In physical terms this could be expressed as the forces acting along a specified axis. Let forces along u be approximated by a linear combination of forces acting on line segment AB. In the same manner forces acting along v can be approximated by a linear combination of the line from C to a point D on and orthogonal to AB. More specifically one has

$$\begin{aligned} \frac{\partial f}{\partial u} &= \frac{f_B - f_A}{\|B - A\|} \qquad (2.9)\\ \frac{\partial f}{\partial v} &= \frac{f_C - f_D}{\|C - D\|} \qquad (2.10) \end{aligned}$$

D can be expressed as

$$D = A + \theta(B - A) \tag{2.11}$$

where

$$\theta = \frac{\langle C - A, B - A\rangle}{\|B - A\|^2} \tag{2.12}$$

Through the linear nature of f and relations (2.9), (2.10), (2.11) and (2.12) inserted into (2.8) we derive

$$\begin{aligned} \int\int_{\Sigma} f\left(\frac{\partial}{\partial u} - i\frac{\partial}{\partial v}\right)\delta_p dS &= \frac{f_A}{\|B - A\|} - \frac{f_B}{\|B - A\|}\\ &\quad + i\frac{f_C - (f_A + \theta(f_B - f_A))}{\|C - D\|} \qquad (2.13) \end{aligned}$$

By restricting to $PL(\Sigma)$, seeking $z \in PL(\Sigma)$, then for all $f \in PL(\Sigma)$ we have

$$\int\int_{\Sigma} \nabla z \cdot \nabla f dS = \left(\frac{\partial f}{\partial u} - i\frac{\partial f}{\partial v}\right)|_p \tag{2.14}$$

where ∇z is the gradient with respect to the induced metric on Σ.

Let $P, Q \in \Sigma$ be any pair of vertices on the triangulated unit sphere, and let ϕ_P be a continuous function, linear on each triangle, conforming to

$$\phi_P(P) = 1, \tag{2.15}$$
$$\phi_P(Q) = 0, \quad Q \neq P \tag{2.16}$$

thus forming a basis for $PL(\Sigma)$. What remains is to derive an expression for z such that

$$z = \sum_P z_P \phi_P \tag{2.17}$$

By combining (2.17) and (2.14) it is given that

$$\sum_P z_P \int\int_\Sigma \nabla z \cdot \nabla f dS = \frac{\partial \phi_Q}{\partial u}(p) - i\frac{\partial \phi_Q}{\partial v}(p) \tag{2.18}$$

for all Q and fixed p.

Building the Stiffness Matrix

Based on the formulation derived in the previous section, a linear system of equations is formulated, thus forming a solution to the conformal mapping problem. In matrix terms (2.18) becomes

$$\mathbf{K}_{P,Q} = \int\int \nabla \phi_P \cdot \nabla \phi_P dS \tag{2.19}$$

where $\mathbf{K}$ is the stiffness matrix and P, Q is any pair of vertices on Σ. Consider two adjacent triangles, σ_A and σ_B, sharing a common edge on a triangulated mesh Σ as illustrated in figure 2.2.

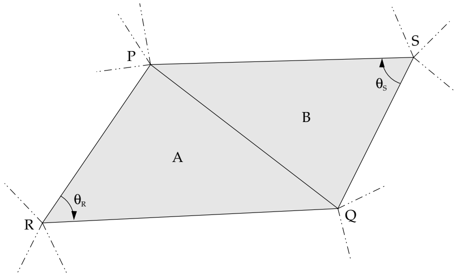


Figure 2.2: Relation between angles of opposite vertices on adjacent triangles forms base of the stiffness matrix $\mathbf{K}$.

Using triangular elements when solving Laplaces equation with a finite element model, the following relation exists between vertices $\mathbf{v}_R$ and $\mathbf{v}_S$:

$$\mathbf{K}_{P,Q} = -\frac{1}{2}(\cot\theta_R + \cot\theta_S), \quad P \neq Q \tag{2.20}$$

Diagonal elements are formed by

$$\mathbf{K}_{P,P} = -\sum_{P \neq Q} \mathbf{K}_{P,Q} \tag{2.21}$$

Based on a triangle $\sigma = ABC$ we compose an expression for $a_Q - ib_Q = \frac{\partial \phi_Q}{\partial u}(p) - i\frac{\partial \phi_Q}{\partial v}(p)$ as

$$a_Q - ib_Q := \begin{cases} 0 & , \quad Q \notin \{A, B, C\} \\ \frac{-1}{\|B-A\|} + i\frac{1-\theta}{\|C-D\|} & , \quad Q = A \\ \frac{1}{\|B-A\|} + i\frac{\theta}{\|C-D\|} & , \quad Q = B \\ i\frac{-1}{\|C-D\|} & , \quad Q = C \end{cases} \tag{2.22}$$

A solution to the complex system of equations can thus be derived by solving the system for the real respectively the imaginary parts. This forms the following two linear systems of equations

$$\mathbf{Kx} = \mathbf{a} \tag{2.23}$$

$$\mathbf{Ky} = -\mathbf{b} \tag{2.24}$$

Solving the system of equations produces a complex mapping plane as illustrated in figure 2.3.

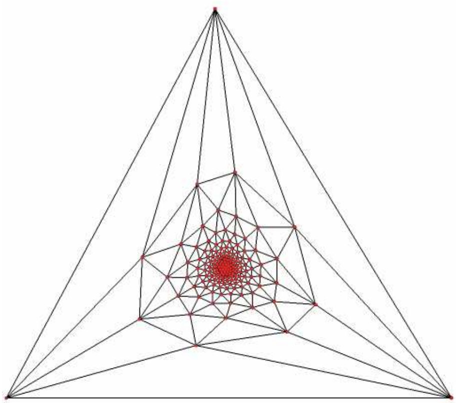


Figure 2.3: The 2D complex plane of a conformally mapped sphere with uniformly distributed points. The largest triangle being $\sigma = ABC$ in which p resides.

Ensuring Uniqueness

Since the map is only unique up to scale and translation, we ensure uniqueness by translation to the center of mass, $\overline{\mathbf{v}}$ defined as

$$\overline{\mathbf{v}} = \frac{1}{n_v} \sum_{i=1}^{n_v} \mathbf{v_i} \tag{2.25}$$

where n_v is the number of vertices and in turn rescale to let half the vertices lie within a unit circle with origo at the center of mass.

Inverse Stereographic Projection

We introduce *inverse stereographic projection* as a means of wrapping a complex 2D surface onto the unit sphere, thus composing the conformal map. This is achieved by employing the transformation

$$f(z) = \left(\frac{2x}{(1+r^2)}, \frac{2y}{(1+r^2)}, \frac{2r^2}{(1+r^2)} - 1 \right) \tag{2.26}$$

$$r^2 = x^2 + y^2 \tag{2.27}$$

where $z = x + iy$. To clarify, the concept of stereographic projection is presented graphically in figure 2.4.

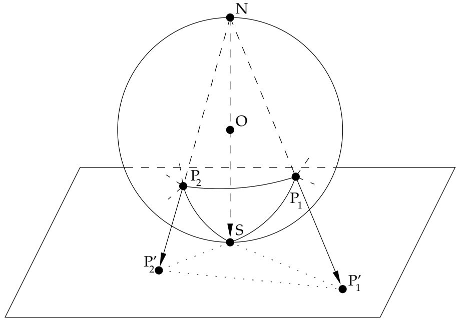


Figure 2.4: Stereographic projection of sphere onto plane. P_i' denotes projected points, P_i denotes original points on the sphere with origo at O and N/S are north/south poles.

2.2.4 Summary of Algorithm

Based on the definitions and equations described in the preceding sections, the algorithm of conformal mapping complies to the following rules:

Algorithm 1 Conformal mapping of a shape to the unit sphere.

```
Require: Shape is topologically equivalent to a sphere.
1: Compute $\mathbf{K}$, $\mathbf{a}$ and $\mathbf{b}$.
2: Compute the planar complex map, $\mathbf{C}$ by solving $\mathbf{Kx} = \mathbf{a}$ and $\mathbf{Ky} = -\mathbf{b}$.
3: Use inverse stereographic projection, $f_{ISP} : \mathbf{C} \rightarrow \mathbf{S}$ to project $\mathbf{C}$ onto the unit sphere, $\mathbf{S}$.
```

In figure 2.5 this approach has been utilized to produce a conformal map of a sphere with uniformly distributed points. In this case, the specific choice of object, displays the area distortion problem encountered in conformal maps.

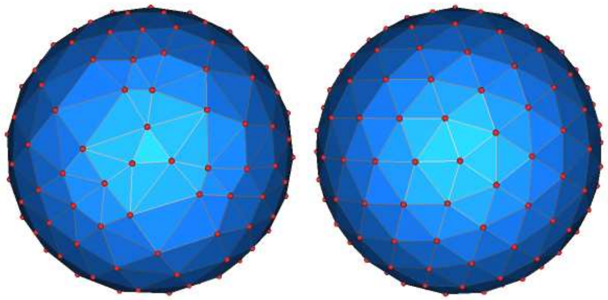


Figure 2.5: Conformal map of a sphere with uniformly distributed points, seen from opposite poles.

2.3 Area-preserving Maps

2.3.1 Overview

The area distortion introduced through conformal mapping stems from a fundamental fact of mapping theory:

> **Definition 2.3.1.** *No mapping between two surfaces exists which is both conformal and area-preserving.*

To overcome this problem a novel optimization scheme is proposed. Depending on geometry of the shape being conformally mapped, some level of area-distortion is introduced. The initial goal of mapping a shape to the unit sphere was to reduce the total number of parameters needed to represent each shape. A requirement of such a mapping procedure must be, that it retains information of size and shape, in other words areas and angles. Though a fairly simple approach, the optimization scheme presented in the following sections aims at optimizing each vertex position based on a least squares error measurement of area differences between each pair of original and conformally mapped shapes.

2.3.2 Local Optimization

Consider a local region around vertex **v** of a triangulated conformal map on the unit sphere, as shown in figure 2.6.

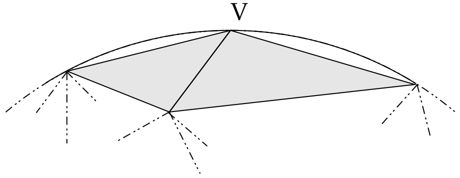


Figure 2.6: The local mesh region connected to vertex **v** is considered to be planar.

Assume the region around vertex **v** is planar. Though this is an approximation, the nature of errors introduced by this assumption is of the form

$$\theta \approx sin(\theta) \tag{2.28}$$

for $\theta \to 0$ where θ is defined as the angle between the optimal plane and a vector drawn from the point of intersection between plane and triangle edge to vertex **v**, measured orthogonally to the plane, and illustrated in figure 2.7.

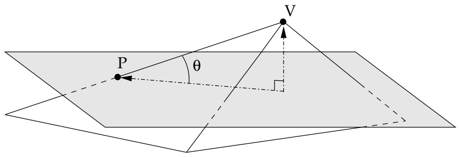


Figure 2.7: Assuming planar conditions around vertex **v** introduces an error, related to angle θ.

Optimal plane

The transformation problem is posed as one of fitting a plane to the set of 3D points connected by edges to vertex **v**. Residing on a sphere, points express a higher degree of variation along the surface, rather than along an axis orthogonal to the surface. This fact allows the axis along which the least variation is observed to be estimated using *principal component analysis*, PCA [8].

For a more detailed description of principal component analysis and the eigenvalue problem please refer to 3.6.

Consider the vector

$$\mathbf{x} = [x, y, z]^T \tag{2.29}$$

as a vector of point coordinates with mean value based on N observations calculated by

$$\overline{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x_i} \tag{2.30}$$

Let the covariance matrix be defined as

$$\mathbf{D} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T \tag{2.31}$$

Thus principal axes are now given by the eigenvectors, $\mathbf{P}$, of the covariance matrix, $\mathbf{D}$, as

$$\mathbf{DP} = \mathbf{P\Lambda} \tag{2.32}$$

where Λ is a diagonal matrix of eigenvalues, λ_i, corresponding to the eigenvectors, $\mathbf{p}_i$, in columns of $\mathbf{P}$ [18].

Sorting eigenvalues in descending order and extracting the eigenvector corresponding to the smallest eigenvalue yields the normal $\mathbf{n}$ of the optimal plane, minimizing distances from point to plane measured orthogonally to the plane.

The rotation matrix, $\mathbf{R}$ which rotates the optimal plane normal, $\mathbf{n}$, onto the z-axis is derived. Given $\mathbf{R}$, vertex $\mathbf{v}$ along with points connected by edges to $\mathbf{v}$ are transformed. By disregarding the z-coordinates of rotated points, the local optimization problem can be approximated in 2D.

Vertex Optimization

Once transformed into 2D, a set of linear equations describing the relations between areas of triangles on the original and mapped mesh can be derived. Thus a solution to the optimal vertex placement is given by the objective function

$$\min_f \sum_{\sigma=1}^{n}(A_{m,\sigma} - A_{o,\sigma})^2 \tag{2.33}$$

where $A_{m,\sigma}$ is the area of triangle σ on the spherical mapping, locally projected onto the plane and $A_{o,\sigma}$ is the area of the corresponding triangle on the original shape.

Let A_σ denote the area of a triangle, $\sigma = ABC$, defined as

$$A_\sigma = \tfrac{1}{2}|\mathbf{AB}||\mathbf{CD}| \tag{2.34}$$

where $\mathbf{D}$ is the projection of $\mathbf{C}$ onto and orthogonal to $\mathbf{AB}$.

Consider a triangle, σ, as illustrated in figure 2.8, with normal, $\mathbf{n}$, of the base line, $\mathbf{b}$. Let $\mathbf{l}$ be a line through origo and parallel to $\mathbf{n}$.

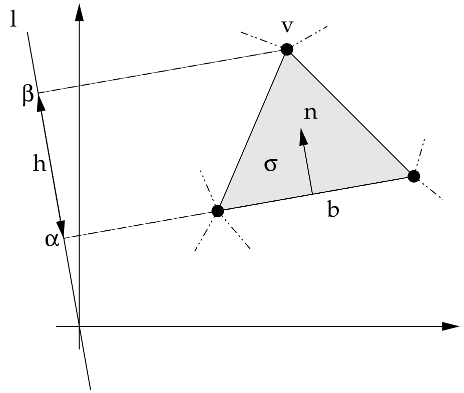


Figure 2.8: Calculation of triangle area using orthogonal projection of points onto a line through origo and parallel to normal $\mathbf{n}$.

Any point $\mathbf{x}$ on $\mathbf{l}$ can thus be expressed as

$$\mathbf{n}^T \cdot \mathbf{x} = \alpha \tag{2.35}$$

where α is the distance to origo, measured along $\mathbf{l}$. Rewriting (2.35) into *homogenous coordinates* one has

$$\begin{aligned} \mathbf{n}^T \cdot \mathbf{x} - \alpha &= 0 && (2.36)\\ \begin{bmatrix} \mathbf{n} \\ -\alpha \end{bmatrix}^T \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} &= 0 && (2.37) \end{aligned}$$

In the same manner the distance, β, at which vertex $\mathbf{v}$ is projected onto $\mathbf{l}$ and measured from origo, is defined as

$$\begin{aligned} \mathbf{n}^T \cdot \mathbf{v} - \beta &= 0 && (2.38)\\ \beta - \alpha &= h && (2.39) \end{aligned}$$

where h is the height of triangle σ measured along $\mathbf{l}$. From (2.37), (2.38) and (2.39) we derive

$$\begin{aligned} \mathbf{n}^T \cdot \mathbf{v} - \alpha &= h && (2.40)\\ \begin{bmatrix} \mathbf{n} \\ -\alpha \end{bmatrix}^T \begin{bmatrix} \mathbf{v} \\ 1 \end{bmatrix} &= h && (2.41)\\ \mathbf{n}_w^T \cdot \mathbf{v}_w &= h && (2.42) \end{aligned}$$

where $\mathbf{n}_w^T$ denotes the normal in homogenous coordinates with $w = \alpha$ and $\mathbf{v}_w$ is vertex $\mathbf{v}$ also defined in homogenous coordinates with $w = 1$.

By inserting (2.42) into (2.34) we derive a novel expression for the area, $A_{m,\sigma}$ of any mapped triangle σ as

$$A_{m,\sigma} = \tfrac{1}{2}\mathbf{n}_w^T \cdot \mathbf{v}_w \tag{2.43}$$

For each vertex $\mathbf{v}$, n_{ve} occurrences of expression (2.43), where n_{ve} is the number of edges connected to v, form a linear system of equations to be solved in the least squares sense.

$$\|\mathbf{D}_m \mathbf{v} - \mathbf{A_o}\|^2 \tag{2.44}$$

Here $\mathbf{D}_m$ is defined as

$$\mathbf{D}_m = \frac{1}{2}\begin{bmatrix} n_{w,1,x} & n_{w,1,y} & \alpha_1 \\ n_{w,2,x} & n_{w,2,y} & \alpha_2 \\ \vdots & \vdots & \vdots \\ n_{w,n_{ve},x} & n_{w,n_{ve},y} & \alpha_{n_{ve}} \end{bmatrix} \tag{2.45}$$

By solving equation (2.44), we derive the best position for vertex $\mathbf{v}$, based on area distortion. Since the optimal position is defined in 2D, rotation and normalization form the final steps.

2.3.3 Global Optimization

The local optimization scheme approximates the best vertex position by a set of linear equations. Due to the nature of the local optimization scheme, each step decreases the local and global objective. In other words, the most likely step for each iteration is chosen. In this manner, probabilities are maximized, thus yielding a *maximum a posteriori*, MAP, estimate of the triangulated surface, $\overline{\Sigma}$. This approach constitutes a *deterministic relaxation* scheme, better known as *iterated conditional modes*, ICM [2] as opposed to *simulated annealing* which constitutes a *stochastic relaxation* scheme.

The global minimization approach conforms to the following simple rules.

Algorithm 2 Minimization of area distortion.

```
repeat
    Rescale the spherical map such that the sum of areas of all triangles
    equals the sum of areas of all triangles on the original shape.
    For all vertices, and in a randomized manner, optimize vertex posi-
    tions using the local optimization scheme.
    Calculate the residual between optimization steps.
until convergence
```

Convergence is achieved as the residual between successive iterations is below some specified threshold.

Although the residual may never converge towards 0, due to the errors imposed by the assumption $\theta = sin(\theta)$, a rough estimate of the global minimum, ε, using the 2D approximation can be obtained by:

$$\varepsilon = \frac{1}{2n_e} \sum_{v=1}^{n_v} \sum_{e=1}^{n_{ve}} \theta_{ve} \tag{2.46}$$

where n_e is the total number of edges, n_v is the total number of vertices, n_{ve} is the number of edges at vertex v, and θ_{ve} is the angle between the e^{th} edge and the optimal plane at vertex v.

Reducing Artifacts

Though the novel optimization scheme minimizes area distortion, a potential hazard exist: If the initial conformal map expresses a high degree of area distortion, particularly if the number of highly distorted triangles is little, $n < 0.1 \cdot n_v$, the local vertex optimization scheme produces large steps, extending triangle edges beyond recovery. This effect will propagate to the remaining triangles, thus forcing the algorithm into a local minima. To avoid this type of artifact, a weight, ω, has been added to the local optimization scheme.

Consider the 2D local optimization problem. Let $\mathbf{v}_o$ be the optimal position for a vertex $\mathbf{v}$, and let $\mathbf{T}$ be the translation vector describing the transformation $\mathbf{v} \rightarrow \mathbf{v}_o$, thus one has

$$\mathbf{v}_o = \mathbf{v} + \omega \mathbf{T} \tag{2.47}$$

Figure 2.9 displays the effect of weight factor ω on area distortion achieved by the optimization scheme introduced.

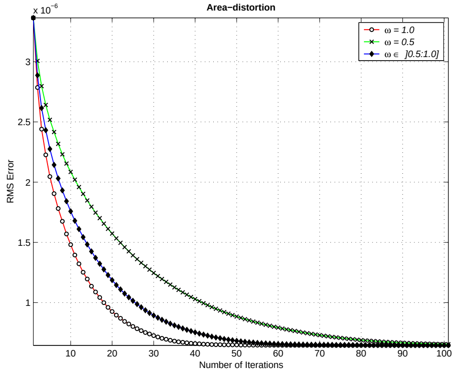


Figure 2.9: RMS Error for different values of weight ω as a function of the optimization step.

Cross validation has revealed $\omega \in [0.5 : 1.0]$ as the best weight factor interval when used in combination with the MDL model derived in succeeding chapters.

Figure 2.10 illustrates the effect of employing the area distortion minimization scheme suggested.

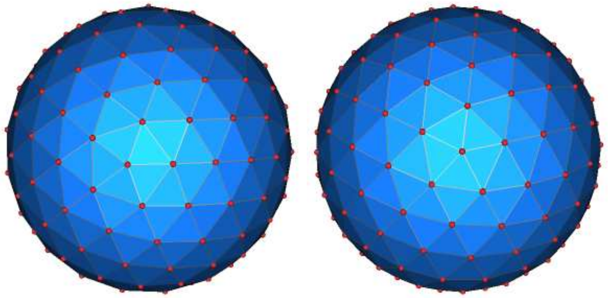


Figure 2.10: Optimized area-preserving map, based on conformal map of a sphere with uniformly distributed points, seen from opposite poles.

2.4 Summary

Throughout this chapter, a method for reducing the parameter space of a set of shapes based on spherical mapping have been introduced. Key mathematical concepts, aided by illustrative examples, explain the framework for employing such a method with little loss[1] of general shape characteristics, through the utilization of a novel area-preservation criteria.

[1] Some degree of loss is inevitable due to the nature of combining conformal and area-preserving maps, see definition 2.3.1.

Chapter 3

Statistical Shape Models

“There are three kinds of lies: lies, damned lies, and statistics.”
– Benjamin Disraeli

3.1 Overview

The following chapter provides the fundamental concepts and techniques for building a statistical shape model. Starting off with a definition of shapes and landmarks, different approaches on aligning a set of landmarked shapes, with and without landmark correspondence, are presented. A method for modelling shape variations seen across a given data set - based on point distribution models, PDM - concludes this chapter.

3.2 Shapes and Landmarks

To clarify the concept of statistical shape analysis, one needs to define the term *shape*. In everyday language a shape is referred to as "the appearance of an object". The Merrian-Webster dictionary explains the concept of shape as

> *- The visible makeup characteristic of a particular item* [34].

Though it describes the meaning of shape in common terms, this does not suffice in the mathematical understanding of the term. To clarify we adopt the definition of D. G. Kendall [16]:

> **Definition 3.2.1.** ***Shape*** *is all the geometrical information that remains when location, scale and rotational effects are filtered out from an object.*

This definition implies that a shape is invariant to *Euclidean similarity transformations*.

In the same manner, we define a shape representation as a finite number of points positioned along the outline of an object. These points constitute *landmarks*, and can be divided into 3 categories, again adopting the notation from Kendall [16].

- **Anatomical landmark** A point assigned by an expert that corresponds between organisms in some biologically meaningful way.
- **Mathematical landmark** A point located on an object according to some mathematical or geometrical property of the figure, e.g. at a point of high curvature or at an extreme point.
- **Pseudo landmark** Constructed point on an object, located either around the outline or in between anatomical or mathematical landmarks.

3.3 Point Distribution Models

Having defined the required primitives, a convenient framework for statistically analyzing and modelling variations across similar shapes, proposed by Cootes and Tayler [10], is employed. This approach constitutes *point distribution models*. The model utilizes knowledge acquired through statistical analysis of shape variation, based on an aligned data set. The model building procedure is one of three steps. These constitute:

1. **Shape capture:** Establishment of a set of corresponding landmarks across the data set.

2. **Shape alignment:** Alignment of the data set using the Procrustes distance metric.

3. **Shape analysis:** Principal component analysis of variation seen across the data set.

The force of this approach lies in its ability to produce a compact model.

In the following sections, the steps involved in building a PDM are explained in detail.

3.4 Obtaining Landmarks

Generally the acquisition of landmarks is a cumbersome and tedious task. In most cases the landmarking effort requires a trained expert within the specific field, since the problem of positioning landmarks in a manner consistent with the medical or biological variation one seeks to analyze most likely relies on forehand knowledge. Furthermore the task becomes increasingly difficult as dimensionality increases.

While no "golden" automated or semi-automated approach exists, several good attempts to construct a such has carried out.

As an example, Thirion [46] employs a method for extracting an extremal mesh, based on crest lines[1] derived using the marching lines algorithm [47]. Extremal points are calculated as the point of intersection between two or more crest lines, based on gradients. Though this approach simplifies subsequent alignment, the matching of crest lines is not straightforward.

Another example authored by Wang et al. utilizes a hierarchical refinement process, starting off with a sparse representation of landmarks. For each iteration, the most likely landmark candidates are selected through evaluation of measures on shortest path, curvature and surface normals at local patches of a triangulated mesh [49].

[1] Lines of maximal curvature along an object.

Brett and Taylor [6] suggest an approach based on a decimation scheme in which a sparse polygonal model of the object in question is derived. For each iteration, vertices are removed, and selection is determined through a distance metric preserving sharp edges and thin structures. The resulting sparse triangulated meshes can thus be aligned using iterative closest point, see 3.5.1.

In chapter 4 an approach to "circumvent" the landmarking process is presented. This method relies on establishment of correspondence through automated generation of landmarks based on spherical mappings of a given set of shapes.

3.5 Aligning a Set of Shapes

A prerequisite of statistical shape analysis is the establishment of a common *shape space*, a space in which variation across any given set of shapes can be measured and evaluated. From [16] we adopt the definition

> **Definition 3.5.1.** *The **shape space** is the set of all possible shapes. Formally, the shape space $\sum_m^k$ is the orbit space of the non-coincident k point set configurations in $\mathbb{R}^m$ under the action of the Euclidean similarity transformations.*

In other words we wish to filter out effects of scale, translation and rotation, as stated in definition 3.2.1. This is achieved through shape alignment.

Alignment algorithms can generally be divided into two categories:

- Correspondence based
- Non-correspondence based

The most common methods in each of the two categories are the Procrustes [20] and iterative closest point or ICP [3] alignment algorithms. Extensions to both methods exists. As an example Larsen and Eiríksson [31] combines Procrustes alignment with different norms, the l_2-norm being utilized in ordinary Procrustes alignment. As an example of extending ICP, Zhang [51] introduced the ability to estimate 3D motion through successive ICP alignment procedures.

The Euclidean Distance Metric

As a common measurement of "goodness", i.e. how well a specific point is aligned, we employ the *Euclidean distance metric* defined as

$$d(\mathbf{v}_1, \mathbf{v}_2) = \|\mathbf{v_1} - \mathbf{v}_2\| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (3.1)$$

where $\mathbf{v}_i$ is a vertex represented as $\mathbf{v}_i = [x_i, y_i, z_i]^T$, and d is the general distance metric. Extensions to more complicated primitives, such as lines and planes, will not be documented further in this thesis, though it should be noted, they can be derived using simple geometry.

3.5.1 Iterative Closest Point

The iterative closest point algorithm is especially useful when alignment of non-corresponding data sets is required. Employing a least squares approach on inter-point distance minimization, a corresponding set of landmarks is derived. For each set of corresponding landmarks, a transformation vector is calculated based on unit quarternions [23]. ICP was developed by Besl and McKay and has become a widely used approach on shape alignment.

Let $\mathbf{p}$ be a point on shape $\mathbf{P}$, and let $\mathbf{X}$ be the model shape, to which $\mathbf{P}$ should be aligned. Then the distance metric, d, between $\mathbf{p}$ and $\mathbf{X}$ is defined as

$$d(\mathbf{p}, \mathbf{X}) = \min_{\mathbf{x} \in \mathbf{X}} \|\mathbf{x} - \mathbf{p}\| \quad (3.2)$$

where $\mathbf{x}$ is any point on $\mathbf{X}$. Furthermore, let $\mathcal{C}$ be the closest point operator such that

$$\mathbf{Y} = \mathcal{C}(\mathbf{P}, \mathbf{X}) \quad (3.3)$$

where $\mathbf{Y}$ is the resulting set of closest points on $\mathbf{X}$. Finally let $\mathbf{q}$ be the unit quarternion which minimizes distances between corresponding landmarks in $\mathbf{P}$ and $\mathbf{Y}$ in a least squares manner, by scale, translation and rotation. The least squares registration is then given by

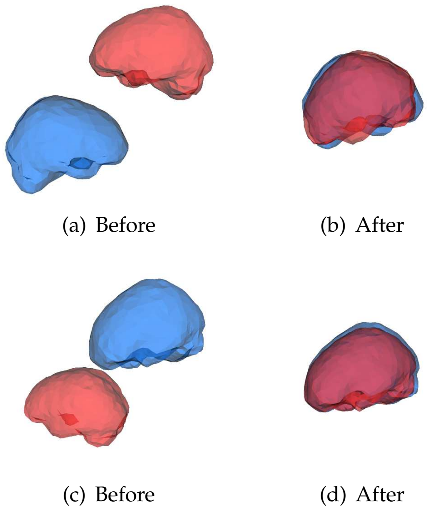
$$(\mathbf{q}, d) = \mathcal{Q}(\mathbf{P}, \mathbf{Y}) \quad (3.4)$$

Having briefly defined the fundamentals of ICP, attention is focused on algorithm structure. Given $\mathbf{P}$ and $\mathbf{X}$, the algorithm employs the following set of rules:

Algorithm 3 Iterative closes point alignment.

```
repeat
    Compute the closest points $\mathbf{Y}_k = \mathcal{C}(\mathbf{P}_k, \mathbf{X})$.
    Compute the alignment, $(\mathbf{q}_k, d_k) = \mathcal{Q}(\mathbf{P}_k, \mathbf{Y}_k)$.
    Apply the registration to produce $\mathbf{P}_{k+1} = \mathbf{q}_k(\mathbf{P}_k)$.
until convergence
```

It should be noted, that the algorithm only ensures convergence towards a local minima. To illustrate this concept, examine the two alignment problems posed in figure 3.1.


(a) Before

(b) After

(c) Before

(d) After

Figure 3.1: (a) before and (b) after failing ICP alignment due to convergence towards incorrect minimum. (c) before and (d) after successful ICP alignment.

For further details on ICP, please refer to [3].

3.5.2 Procrustes Alignment

To align a set of shapes with corresponding landmarks the generalized *Procrustes alignment* algorithm [20] is employed. Relying on the Euclidean distance metric between two points, the algorithm aligns a set of shapes to a mean shape estimate.

A key problem in solving the Procrustes alignment problem is to derive the optimal rotation between two sets of N corresponding points. Examples of possible solutions to the rotation matrix problem are singular value decomposition [26] of the correlation matrix, and unit quaternions [23].

The Procrustes algorithm employs an iterative approach, calculating a new mean shape estimate, $\overline{\mathbf{x}}$, defined as

$$\overline{\mathbf{x}} = \frac{1}{n_s} \sum_{i=1}^{n_s} \mathbf{x}_i \tag{3.5}$$

for each new configuration, k, only allowing *rigid body transformations*, i.e. translation and rotation. Upon each new configuration, the set of shapes are re-scaled to unit size. To give meaning to this operation, we employ a common size metric.

Let $S(\mathbf{x})$ be the *centroid size* metric [16, 44] defined by

$$S(\mathbf{x}) = \sqrt{\sum_{i=1}^{N} (x_i - \overline{x})^2 + (y_i - \overline{y})^2 + (z_i - \overline{z})^2} \tag{3.6}$$

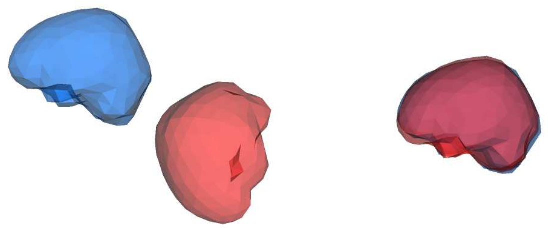
for which the relation $S(a\mathbf{x}) = aS(\mathbf{x})$ is satisfied.

If $\mathbf{X}$ is the set of shapes to be aligned, then the algorithm complies to the following rules:

Algorithm 4 Procrustes alignment.

```
Choose the first shape $\mathbf{x}_0$ as the initial mean shape estimate, $\overline{\mathbf{x}}$.
repeat
    Align the set of shapes, $\mathbf{X}$, to the mean shape estimate.
    Re-calculate the mean shape estimate based on the new shape configuration.
    Re-scale mean shape to unit size and zero rotation.
until convergence.
```

Convergence is achieved as the residual between successive iterations is below some specified threshold, and usually within 2 iterations [4].



(a) Before

(b) After

Figure 3.2: (a) before and (b) after Procrustes alignment.

3.6 Modelling Shape Variations

A common and well defined problem in shape analysis is one of extracting information regarding inter-point correlation between a set of shapes. Suppose we have a set of n_s shapes, each with N points, aligned within a common coordinate system. From the set of shapes a model can be derived, able to reproduce any instance of the shape set, as well as synthesizing new shapes similar to the ones found in the shape set used in the model building process.

Several attempts on capturing variations seen across a set of shapes, based on statistical variation analysis, have been described. Methods include maximum autocorrelation factors, MAF [45], and independent component analysis, ICA [25]. A good review of these methods, though based on 2D shape analysis, can be found in [31].

The goal of any such analysis approach is to derive a compact model of shape variations. In this thesis attention is focused on *principal component analysis*, PCA.

3.6.1 Principal Component Analysis

Consider a 3n-dimensional vector of 3D point coordinates

$$\mathbf{x} = [x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n, z_1, z_2, \ldots, z_n]^T \tag{3.7}$$

with mean value based on n_s observations calculated by

$$\overline{\mathbf{x}} = \frac{1}{n_s}\sum_{i=1}^{n_s}\mathbf{x_i} \tag{3.8}$$

Let the covariance matrix be defined as

$$\mathbf{D} = \frac{1}{n_s}\sum_{i=1}^{n_s}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T \tag{3.9}$$

Eigenvalues for $\mathbf{D}$ are given by

$$|\mathbf{D} - \lambda\mathbf{I}| = 0 \tag{3.10}$$

where $|\cdot|$ denotes the determinant. Principal axes are then given by the eigenvectors of the covariance matrix as

$$\mathbf{DP} = \mathbf{P\Lambda} \tag{3.11}$$

where $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues, λ_i, defined by

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_i & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_{3n} \end{bmatrix} \tag{3.12}$$

Corresponding eigenvectors, $\mathbf{p}_i$, are defined in columns of $\mathbf{P}$ as.

$$\mathbf{P} = \begin{bmatrix} \\ \mathbf{p}_1 & \cdots & \mathbf{p}_{3n} \\ \\ \end{bmatrix} \tag{3.13}$$

Any given shape instance of the data set can thus be derived through the linear combination

$$\mathbf{x} = \overline{\mathbf{x}} + \mathbf{Pb} \tag{3.14}$$

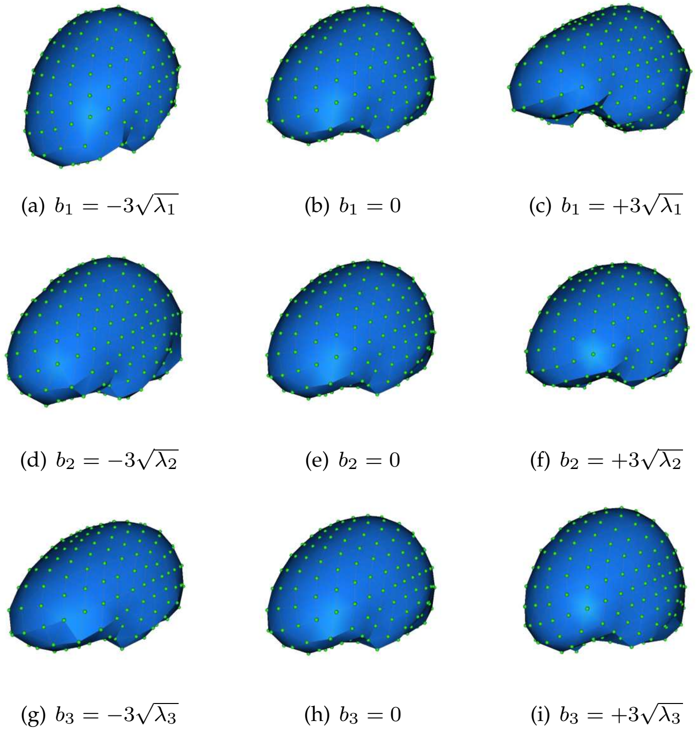
where $\mathbf{b}$ is defined as the shape model parameters given by

$$\mathbf{b} = \mathbf{P}^T(\mathbf{x} - \overline{\mathbf{x}}) \tag{3.15}$$

By sorting eigenvalues and corresponding eigenvectors in descending order the t principal axes, or modes, responsible for a predefined level of variance, i.e. 95%, can be identified. Let the specified quantile, q, be defined as

$$\sum_{i=1}^{t}\lambda_i \geq q\sum_{i=1}^{3n}\lambda_i \tag{3.16}$$

Recall that eigenvectors are axes along which the mean shape expresses variations. To illustrate this fact, figure 3.3 displays the variations found in a given data set along the largest three principal axes.



(a) $b_1 = -3\sqrt{\lambda_1}$ (b) $b_1 = 0$ (c) $b_1 = +3\sqrt{\lambda_1}$

(d) $b_2 = -3\sqrt{\lambda_2}$ (e) $b_2 = 0$ (f) $b_2 = +3\sqrt{\lambda_2}$

(g) $b_3 = -3\sqrt{\lambda_3}$ (h) $b_3 = 0$ (i) $b_3 = +3\sqrt{\lambda_3}$

Figure 3.3: Deformation of a human brain mean shape along the first three principal axes.

Extending PCA

Suppose we wish to model a set of n_s shape vectors, $\mathbf{x}_i$, each consisting of n_p landmarks for which the condition $n_s < n_p$ holds. Since the dimension of the covariance matrix in this case is $n_p \times n_p$, it is desirable to deduct an alternative method to extract eigenvectors $\mathbf{P}$ and corresponding eigenvalues λ, thus reducing the computational effort required to solve the eigenvalue problem. This can be achieved using the following approach.

Consider a $n_s \times n_p$ matrix $\mathbf{W}$ defined as

$$\mathbf{W} = \begin{bmatrix} \mathbf{x}_1 - \overline{\mathbf{x}} \\ \mathbf{x}_2 - \overline{\mathbf{x}} \\ \vdots \\ \mathbf{x}_{n_s} - \overline{\mathbf{x}} \end{bmatrix} \tag{3.17}$$

where $\overline{\mathbf{x}}$ is the mean shape estimate. The $n_s \times n_s$ covariance matrix, $\mathbf{D}$, is then given by

$$\mathbf{D} = \frac{1}{n_s n_p} \mathbf{W}\mathbf{W}^T \tag{3.18}$$

Let a $\mathbf{T}$ be a matrix such that

$$\mathbf{T} = \frac{1}{n_s n_p} \mathbf{W}^T\mathbf{W} \tag{3.19}$$

Finally, let $\mathbf{p}_i$ be the n_s eigenvectors of $\mathbf{T}$ with corresponding eigenvalues λ_i sorted in descending order. Using the above notation, it can be shown that the n_s vectors $\mathbf{W}\mathbf{p}_i$ are eigenvectors of D with corresponding eigenvalues λ_i. Further it can be shown, that all remaining eigenvectors of $\mathbf{D}$ have zero eigenvalues [9].

3.7 Summary

Throughout this chapter the mathematical fundamentals of building and analyzing a statistical shape model, based on Procrustes alignment and principal component analysis, have been described. A set of definitions form the environment, in which the model resides.

Chapter 4

Minimum Description Length Shape Models

"The art of doing mathematics consists in finding that special case which contains all the germs of generality."
– David Hilbert

4.1 Overview

In the following chapter a complete approach on building an optimal 3D shape model is presented, partly based on the mathematical foundation served in earlier chapters, partly on new definitions described here. Utilizing the concept of minimal description length, MDL, the shape model, derived using ordinary principal component analysis, is optimized.

By all means, constructing a fully automated shape model is the ultimate objective in statistical shape analysis. However, this has proven to be a difficult task, primarily due to the need for consistent automated landmark extraction. Several attempts, mainly based on combinations of known landmark extraction and shape analysis methods, have been made to construct such models [27, 6, 7].

As an example, Kaus et al. [27] employs a method for automated model building by constructing a triangulated template mesh. By deforming the template mesh, corresponding landmarks are found in segmented volumetric images utilizing a deformation energy measurement. Based on the corresponding set of landmarks, a standard point distribution model is established.

The algorithm proposed in this chapter is based primarily on the work conducted in the field of minimum description length statistical shape modelling by Davies et al. For further details please refer to [13, 11, 15, 14, 11].

4.2 Minimum Description Length

The problem is posed as one of finding a set of optimal parameterizations, $\mathbf{\Phi}_i$ for shape $\mathbf{S}_i$ in a set of shapes, $\mathbf{S}$. Using principal component analysis, this constitutes deriving the optimal set of landmarks for a given training set, in a sense that enforces generalization ability and compactness. This is achieved by employing a *minimum description length* approach, in which the optimization problem is treated as one of minimizing the cost of transmitting model and model parameters [40].

The MDL principle stems from an idea of transmitting a data set as en encoded message, thus the transmission must contain information on both model and model parameters in some encoded form. In shape terms this constitutes sending a mean shape estimate, a set of parameterizations describing the deformation of the mean shape for each member of the data set being transmitted, and in the process evaluating the cost of transmission. In this manner, a balance between model complexity and quality of fit is expressed.

4.2.1 Model Definition

Consider the set of n_s shapes each consisting of n_p-dimensional shape vectors, $\mathbf{x}_i$, conforming to a multivariate Gaussian distribution[1]. To transmit a shape $\mathbf{x}_i$, a coordinate system is chosen which is aligned with the principal axes of the data set. In this manner, the $n_s - 1$ mutually orthogonal eigenvectors, $\mathbf{P}$, sorted by descending eigenvalues, span the subspace which contains the data set being transmitted. In other words $\mathbf{x}_i$ is defined as

$$\mathbf{x}_i = \overline{\mathbf{x}} + \sum_{m=1}^{n_s-1} \mathbf{p}_m b_{m,i} \tag{4.1}$$

In addition, the linear combination allows for the position of any point to be modelled as a one-dimensional Gaussian distribution. Each shape to be transmitted can now be composed as

$$\mathbf{y}_i \equiv \mathbf{p}_m^T(\mathbf{x}_i - \overline{\mathbf{x}}) \tag{4.2}$$

It is assumed that transmitting the mean shape estimate, $\overline{\mathbf{x}}$, and eigenvectors, $\mathbf{P}$, requires a fixed code length.

4.2.2 Description Length

Due to the fact that eigenvectors are mutually orthogonal, the total description length, $\mathcal{L}_{total}$, can be decomposed as the sum over $n_s - 1$ dimensions of shape space:

$$\mathcal{L}_{total} = \sum_{m=1}^{n_s-1} \mathcal{L}_m = \mathcal{L}_{parameters} + \mathcal{L}_{data} \tag{4.3}$$

This requires an expression able of calculating the description length of a 1D data set using the *Gaussian distribution* model

[1] Based on the assumption, that the position of point v on some shape $\mathbf{x}_a$ tends to be more probable in the same neighborhood of a corresponding point v on a similar shape $\mathbf{x}_b$, thus conforming to a Gaussian type distribution.

$$\rho(y, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{y^2}{2\sigma^2}\right) \tag{4.4}$$

Since the MDL principle is restricted to transmitting a finite data set, Y, a quantization parameter, Δ, is imposed upon data values such that $\hat{Y} \rightarrow Y$ for $\Delta \rightarrow 0$. Thus quantized data values can be represented to an accuracy of Δ as

$$y \rightarrow \hat{y}, \quad \hat{y} = n\Delta, \quad n \in \mathbb{Z} \tag{4.5}$$

Assuming all points, $x_{i,v}$, in the original shape set, $\mathbf{x}$, reside within a strictly bounded region given by

$$-\frac{r}{2} \leq x_{i,v} \leq \frac{r}{2}, \quad \forall v = 1, \ldots, n_p, \quad \forall i = 1, \ldots, n_s \tag{4.6}$$

Then the corresponding set of border conditions in shape space is given by

$$|y_{m,i}| \leq R \quad , \quad \forall (m, i) \tag{4.7}$$

where $R = r\sqrt{n_p}$. Variance of the quantized data is calculated by

$$\sigma_m = \sqrt{\frac{1}{n_s} \sum_{i=1}^{n_s} \hat{y}_{m,i}^2} \tag{4.8}$$

4.2.3 The Objective Function

By utilizing *Shannon's codeword length* [42], the code length needed to code a data value, $\hat{y}$, using a probabilistic model, $P(\hat{y})$, can be calculated as

$$\mathcal{L} = -\log P(\hat{y}) \tag{4.9}$$

Using the Gaussian model defined in (4.4), the probability, $P(\hat{y})$, for a data value, $\hat{y}$, in the interval $\hat{y} \pm \frac{\Delta}{2}$ is given by

$$P(\hat{y}) = \int_{\hat{y}-\frac{\Delta}{2}}^{\hat{y}+\frac{\Delta}{2}} \rho(k, \sigma) dk \tag{4.10}$$

This expression can be approximated by

$$P(\hat{y}) \approx \frac{\Delta}{\hat{\sigma}\sqrt{2\pi}} \exp\left(-\frac{\hat{y}^2}{2\hat{\sigma}^2}\right) \tag{4.11}$$

The approximation introduces a minimum error term for all values $\hat{\sigma} \geq 2\Delta$, thus a lower bound of $\sigma_{min} = 2\Delta$ is chosen. Since $\hat{\sigma}$ can take any value within the data range, assuming no prior knowledge, $\sigma_{max} = \frac{R}{2}$ is chosen as upper bound. Though this setup requires the definition of 2 separate expressions, dependant upon whether the actual data value lies within or below the boundaries. In addition, a third expression is needed to evaluate the case where all data values are below the quantization parameter. This yields three different cases:

- **Case 1:** $\sigma > \sigma_{min}$
- **Case 2:** $\sigma \leq \sigma_{min} \wedge Y > \Delta$
- **Case 3:** $Y \leq \Delta$

Thus the total description length, including parameters, for each principal direction becomes

$$\mathcal{L}_m = \begin{cases} \mathcal{L}^{(1)}(\sigma_m, n_s, R, \Delta) & , \quad \sigma_m \geq \sigma_{min} \\ \mathcal{L}^{(2)}(\sigma_m, n_s, R, \Delta) & , \quad \sigma_m < \sigma_{min} \wedge Y > \Delta \\ 0 & , \quad Y \leq \Delta \end{cases} \tag{4.12}$$

where case 3 corresponds to transmitting the mean value and no additional variation. Since the mean value is always transmitted, the contribution from data in case 3 is zero. Utilizing equation (4.3) the total description length can be calculated as

$$\begin{aligned} \mathcal{L}_{total} &= \sum_{m=1}^{n_s-1} \mathcal{L}_m & (4.13) \\ &= \sum_{p=1}^{n_g} \mathcal{L}^{(1)}(\sigma_p, n_s, R, \Delta) + \sum_{q=n_g+1}^{n_s-1} \mathcal{L}^{(2)}(\sigma_q, n_s, R, \Delta) & (4.14) \end{aligned}$$

where n_g is defined as the number of principal directions for which the first case holds. One could say that $\mathcal{L}^{(1)}$ is the cost of transmitting n_g significant directions while $\mathcal{L}^{(2)}$ is the penalty of disregarding the remaining $n_s - n_g - 1$ directions.

Applying the definitions of $\mathcal{L}^{(1)}$ and $\mathcal{L}^{(2)}$ from [14] and replacing $\mathcal{L}_{total}$ with F the full objective function becomes.

$$\begin{aligned} F &= f(\cdot) + \sum_{m=1}^{n_g} (n_s - 2) \ln \sigma_m + \\ &\sum_{m=n_g+1}^{n_s-1} \left((n_s - 2) \ln \sigma_{min} + \frac{n_s + 3}{2} \left[\left(\frac{\sigma_m}{\sigma_{min}} \right)^2 - 1 \right] \right) \qquad (4.15) \end{aligned}$$

where $f(\cdot)$ is a function depending only on the constant terms (Δ, R, n_s), thus evaluating to a constant value.

However the objective function defined in (4.15) requires a substantial computational effort to evaluate due to the need for quantizing the complete data set and re-calculating the shape to be transmitted using equation (4.2). In addition, the part of (4.15) dependant upon σ_m contains terms similar to $\sum \ln \lambda_m$. To simplify the objective function, we can use the following definition as an initial MDL estimate

$$F \approx f(\cdot) + \sum_{m=1}^{n_s-1} \ln(\hat{\lambda}_m + \epsilon) \tag{4.16}$$

where $\hat{\lambda}_m$ is the quantized eigenvalues and ϵ is a normalizing constant forcing the functional to have a minimum value when $\lambda_m \to 0$. This functional is derived from

$$F_D = \ln\left(\prod_m \lambda_m\right) = \sum_m \ln \lambda_m \tag{4.17}$$

which effectively measures the "volume" that the training set occupies in shape space. A similar objective function was proposed by Kotcheff et al. [28]. Since the minimization scheme proposed here requires an iterative solution approach, minimizing the simple objective function and in turn applying the full objective function, yields a much faster implementation. However, only using the simple version and depending on the properties of covariance matrix $\mathbf{D}$ this assumption will favor small eigenvalues, regardless of the proportion of variance contribution, which could result in an non-optimal model. This is primarily due to the inability to differentiate between significant and less significant modes of variation.

For further details on derivation of the full objective function, please refer to [14].

4.3 Parameterization

A shape model is built by casting the correspondence problem as one of defining a parameterization, ϕ_i, for each shape, $\mathbf{x}_i$, which minimizes the objective function, F. If each shape in the training set is topologically equivalent with a sphere, we can employ the following approach

1. Map each shape, $\mathbf{x}_i$ in the training set, $\mathbf{x}$ to the unit sphere using the spherical mapping technique described in chapter 2.

2. Uniformly distribute a fixed number of landmarks, n_l, across each spherical map, using a common north pole.

3. Use barycentric coordinates to wrap the uniform set of landmarks back to the original shapes, creating n_s corresponding sets of n_l landmarks.

Once corresponding landmarks are established, these can be aligned and analyzed using techniques described in the previous chapter.

4.3.1 Initial Registration

Due to the nature of mapping a shape from one domain to another, and utilizing the new domain to compose corresponding landmarks, a initial registration of spherical mappings must be obtained. In the method described in chapter 2 this constitutes registering and aligning the original set of shapes using the ICP alignment algorithm. To obtain a good alignment result, the set of original shapes are roughly aligned by hand, using a sparse registration[2]. Once roughly aligned, the registration obtained in this step is disregarded, since the minimization scheme employed will seek to achieve better correspondence based on manipulation of each set of landmarks.

A suggestion on how to avoid this initial registration step is proposed in chapter 9. For further discussion the initial registration explanation given here will suffice.

4.3.2 Barycentric Coordinates

By mapping points to the spherical surface, a reverse coordinate transform using *barycentric coordinates* will allow a point positioned in an arbitrary triangle τ' on the spherical to mapped in the same relative position in triangle τ on the original triangulated mesh S. This is achieved through a barycentric coordinate transformation [50].

An example of such reverse mapping using a barycentric coordinate transformation is illustrated in figure 4.1 where a set of isolines on the spherical map has been projected back onto the original shape.

[2] This initial registration is a requirement regardless of the mapping algorithm used, since orientation and pose of the spherical map is dependant upon the choice of north pole.

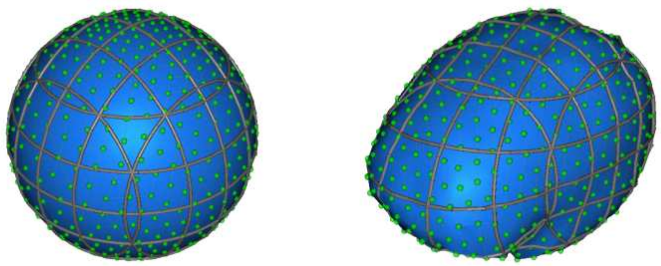


Figure 4.1: Barycentric coordinate transformation of isolines on the spherical map onto the original shape. *Left:* Spherical map with isolines. *Right:* Isolines mapped onto original shape.

The triangulated mesh of a shape can be expressed as

$$S_i = S_i(\theta, \psi) \tag{4.18}$$

where (θ, ψ) are the polar spherical coordinates for all vertices on S_i. Moving vertex, v, on the original mesh corresponds moving v on the spherical map. Thus we have a mapping between vertices given by

$$S_i \to S'_i \quad , \quad \theta \to \theta' \quad , \quad \psi \to \psi' \tag{4.19}$$

where $S_i(\theta, \psi) = S'_i(\theta', \psi')$ and $\theta' = \phi_{i,\theta}(\theta, \psi)$, $\psi' = \phi_{i,\psi}(\theta, \psi)$. Here $\phi_i(\phi_{i,\theta}, \phi_{i,\psi})$ is a parameterization describing only the i^{th} shape in the training set.

4.3.3 Equidistant Points on the Unit Sphere

With no assumptions on surface geometry, i.e. curvature and point densities, the general shape model should have a number of evenly distributed landmarks. Based on the spherical map, this is achieved by placing a number of equidistant points on the spherical surface, and projecting these onto the original shape, using the approach described in the preceding section.

Distributing a fixed number of points across a sphere is a complete field of research, and no "golden" method exists capable of achieving this goal, though several good mesh subdivision techniques come close [17, 22, 32, 52].

Linear Subdivision

The simplest approach on achieving approximated uniformly distributed points on a sphere is to employ the *linear subdivision* method. As most other subdivision methods, this approach is an iterative refinement process.

For each iteration, triangles are divided into four new triangles, by placing a vertex halfway along each edge and forming connections between new vertices. If the initial triangular mesh is a octahedron with eight equivalent equilateral triangles, the number of vertices added by each iteration is $2 + 4^{(k+1)}$, where k is the recursive level. Assuming the center of mass to be origo, hence normalizing vectors from origo to each new vertex, done for each iteration, will project points onto the unit sphere, thus producing an approximated uniform distribution of points on the unit sphere surface.

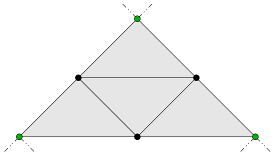


Figure 4.2: Linear subdivision of triangles. Circle = existing vertices, solid = new vertices.

Though a simple approach to the distribution problem, the algorithm fails to incorporate geometric information from surrounding triangles, hence the resulting subdivided mesh is very much dependant upon the starting conditions. Due to this fact, attention is focused on a different approach.

Modified Butterfly Subdivision

The *modified butterfly subdivision* [52] scheme, based in the original butterfly subdivision method by Dun et al. [17], is a somewhat different approach than linear subdivision, most importantly by incorporating local geometry measures such as smoothness.

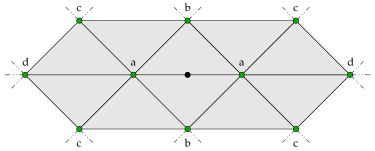


Figure 4.3: 10-point stencil used in modified butterfly subdivision of triangles. Circle = existing vertices, solid = new vertex.

The modified butterfly subdivision scheme employs a 10 point neighborhood stencil to determine the position of new vertices. In this manner, new vertices are positioned along existing edges. Using the notation from figure 4.3 a set of vertex weights are defined and the position of new vertices is calculated as the sum of weighted neighboring vertex positions. Special rules exist for vertices of *valence* other than six[3]. A constant term, w, determines "curvedness" of the resulting mesh. Using the modified butterfly subdivision scheme ensures $\mathcal{C}^1$ continuity of the resulting subdivided surface.

As with the linear subdivision scheme, the modified butterfly method is an iterative approach, for each iteration producing $2 + 4^{k+1}$ new vertices, and in the same manner points are projected onto the unit sphere.

A good description of subdivision schemes is given in [43].

Subdivision Comparison

Using the methods described in preceding sections, two spheres of approximately uniform point distribution, each consisting of $2 + 4^6 = 4098$ points, have been created from a octahedron with eight equivalent equilateral triangles. In figure 4.4 the obvious differences between the two schemes can be examined. Note the linear subdivisions tendency to cluster points in a blossom-like pattern. This form of artifact is due to the fact, that the projection of linear subdivision onto the sphere will force newly created vertices to slide away from each other, since it does not ensure a $\mathcal{C}^1$ continuous surface, thus expanding the new triangle. Though this artifact is also present in the modified butterfly scheme, the ability to incorporate curvature minimizes this type of distortion.

[3] Valence defines the number of edges connected to a given vertex.

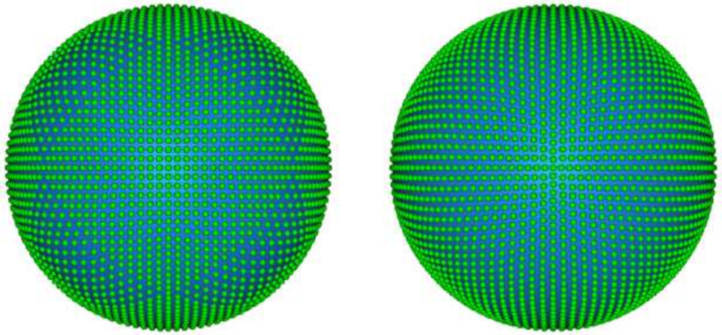


Figure 4.4: Approximating uniform distribution of points on a sphere. *Left:* Linear subdivision. *Right:* Butterfly subdivision.

In subsequent referral to uniformly distributed points on the sphere, the modified butterfly subdivision scheme has been employed.

4.3.4 Manipulating Parameterizations

In order to manipulate the uniformly distributed set of landmarks so as to minimize the objective function, a symmetric landmark transformation is introduced on the spherical polar coordinates. For ease of explanation, consider the arbitrary point P as the point where $\theta = 0$.

Let a rotationally symmetric mapping f be defined as $f : \theta \rightarrow \theta'$ of θ onto θ' and let f be defined such that $f(\pi) = \pi$ and $f(0) = 0$ for $0 < \theta < \pi$. Such mapping can be composed as the cumulative distribution function of some density function defined over the range $0 \leq \theta \leq \pi$.

A *wrapped Cauchy kernel* [33] on the circle centered at $\theta = 0$ has the closed-form indefinite integral:

$$\rho(\theta) = \frac{1}{\pi(1+A)}\left[1 + A\left(\frac{1-\alpha^2}{1+\alpha^2-2\alpha\cos\theta}\right)\right] \tag{4.20}$$

where the width parameter, α, is defined as

$$\alpha \equiv \exp(-a) \quad , \quad a \in \mathbb{R} \tag{4.21}$$

Let A be the kernel amplitude, thus the mapping becomes

$$\begin{aligned} f(\theta) &= \pi\int_0^\theta \rho(s)ds \\ &= \frac{1}{1+A}\left(\theta + A\arccos\left(\frac{(1+\alpha^2)\cos\theta - 2\alpha}{1+\alpha^2-2\alpha\cos\theta}\right)\right) \end{aligned} \tag{4.22}$$

where the constant term $\frac{1}{1+A}$ ensures $f(\theta) = \theta$ when $A = 0$ yielding an unchanged parameterization.

The result of a symmetric θ transformation performed on a sphere with 4098 uniformly distributed points is illustrated in figure 4.5.

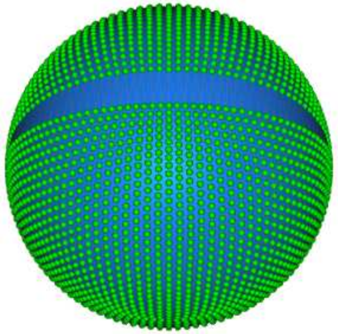


Figure 4.5: Points on the sphere after symmetric θ transformation.

4.3.5 Optimizing Parameterization

As a means of minimizing the objective function, F, we allow the parameterization defined in the previous section to manipulate the set of landmarks. This process forms the basis of an iterative optimization approach.

The idea is to place a number of uniformly distributed kernels across the surface of each set of landmarks, and for each landmark optimize the amplitude, A, while keeping width, a, and reference point, P, fixed. Recall, the landmarks are wrapped back onto the original surfaces using a barycentric coordinate transform, thus for each minimization step, points are realigned, principal component analysis is performed, coordinates are quantized and the objective function is evaluated.

Ideally a minimization scheme would optimize the amplitude of all kernels simultaneously, albeit this is seldom robust nor feasible. Instead a recursive scheme, employing the *Nelder-Mead simplex algorithm* [36], optimizing a single kernel amplitude in turn on each shape is adopted. For each recursive step the kernels added in the corresponding recursive step of subdividing the sphere are optimized. The optimal kernel width is calculated as $a = (\frac{1}{2})^{k+2}$, thus the width is halved for each recursive level.

The Nelder-Mead approach described in this section has been employed in achieving the results presented in subsequent chapters, based on the implementation suggestion given in [38].

4.4 Automated Model Building

In the preceding sections a library of utilities for constructing an optimal shape model based on spherical mapping, barycentric coordinate transform and minimum description length has been presented. To sum up the construction of such a model, the complete algorithmical approach is presented here in pseudo code statements.

Algorithm 5 Minimum description length optimization of statistical shape model.

```
Ensure: $\Delta > 0$
Register and align the set of $n_s$ shapes, $\mathbf{X}$ using ICP$^*$.
Calculate and apply the spherical mapping $f : \mathbf{X} \rightarrow \mathbf{M}$.
Compose a set of $n_s$ spheres, $\mathbf{S}$, with $n_l$ uniformly distributed landmarks.
Compose a sphere, $\mathbf{k}$, with $n_l$ uniformly distributed kernels.
Calculate and apply the barycentric transformation $t : \mathbf{S} \rightarrow \mathbf{L}$ to produce $n_s$ shapes of $n_l$ corresponding landmarks.
Align $\mathbf{L}$ using Procrustes alignment.
Perform principal component analysis.
Evaluate the objective function based on quantized shape data.
repeat
    for all $\mathbf{s}_i$, where $i > 0$, and in random order$^{**}$ do
        for all $k_i$ and in random order do
            Calculate width parameters, $a_r$ and $\alpha$, based on the recursive level $r$
            Calculate optimal kernel amplitude, $A$.
            Apply symmetric theta transformation, $f(\theta)$
        end for
    end for
until convergence
```

4.4.1 Supplemental Notes

*ICP in this case is an example suggestion. The registration method may as such be any method which will produce a set of aligned shapes in some derivation of a least squares measurement, only removing scale, translation and rotation between shapes.

**Randomization is performed only between kernels added in each recursive step of the subdivision algorithm, continuing to the next level once all kernels in the previous level have been optimized, see section 4.3.3.

To prevent landmarks from converging towards a single point, one shape in the set of landmarked shapes is kept fixed throughout the optimization process. Assuming no prior knowledge of shape, this method offers a "best guess" on landmarks, however it does not guarantee optimal landmark positions in terms of ability to describe a given shape.

4.5 Summary

Throughout this chapter, a method for constructing an optimal 3D shape model, based on spherical mapping, landmark parametrization, principal component analysis and minimum description length, has been presented. In the model building process a set of corresponding landmarks is obtained, utilizing a novel area-preserving spherical mapping method, and manipulated to minimize the evaluation of an objective functional. Improvements in landmark distribution enforces generality due to a better shape representation. A simple estimate has been employed to minimize the computational burden while preserving the characteristics of the full objective function.

Part II

Implementation

Chapter 5

Implementation

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

5.1 Overview

Implementation of the previously described techniques and algorithms, resulting in the Spherical Mapping/Shape Modelling Application Programmers Interface, or SM^2-API, has been carried out using Visual C++ 6.0 and targeted at, though not restricted to, the Windows 2000/XP/NT platform. The choice of platform and development environment is partly due the computational requirements involved in implementing algorithms of large scale iterations. In addition, developing in C/C++ offers reuseability for subsequent development along with ease of integration to other software projects.

The main purpose of this chapter is to provide with an overview of the classes and methods in the SM^2-API. Details on usage of the command console interface and XML configuration file are left to the readers own exploration in Appendix B.

5.2 Requirements

Additional functionality was provided by integrating the following 3rd party libraries:

- **Matrix Template Library (MTL):** Sparse/dense Matrix and vector definitions and operations.

- **Iterative Template Library (ITL):** Iterative methods for solving linear system, used in coexistence with MTL.

- **GNU Scientific Library (GSL):** Collection of routines and methods for numerical computing.

- **Visualization Toolkit (VTK):** Collection of methods for 3D graphics, image processing and visualization.

Note, all of the above listed libraries are open source and subjected to public license agreements for personal and non-commercial use.

5.3 Class Overview

This section includes an overview of the different classes contained in the SM^2-API. Each class is commented with a short description of class contents.

`C2DVector`: 2D vector/constant, vector/vector and vector/matrix operations.

`C3DVector`: 3D vector/constant, vector/vector and vector/matrix operations.

`C3DPlane`: 3D plane definition.

`C3DShape`: 3D shape class. Shape/shape operations and spherical mapping.

C3DShapeCollection: 3D shape vector. Multiple shape operations, alignment and PCA procedures.

C3DAnnotation: Core annotation and shape analysis class. Calculation and optimization of MDL model.

C3DNMOptimizer: Nelder-Mead optimization class. Calculation of optimal amplitude of Cauchy kernels.

C3DViewer: Visualization class. OpenGL result viewer.

C3DViewerCallback: Viewer call back class. Handles fixation of back face culling plane in 3D viewer.

C3DUtility: General file I/O utility.

C3DConsole: Console interface. Example wise use of the SM^2-API.

5.4 Console Interface

A console interface has been developed to demonstrate use of the SM^2-API. Most examples and all experiments in the report, were done using this interface, and in addition, the console interface class **C3DConsole** is added to the final distribution.

5.5 Supported File Formats

The current version of the SM^2-API only supports the VTK Polygonal Data file format described in [41].

Part III

Experimental Results

Chapter 6

Experimental design

"Errors using inadequate data are much less than those using no data at all."
– Charles Babbage

6.1 Overview

To assess the performance of the spherical mapping and MDL algorithms, data sets expressing reasonable amount of variability was collected. The training data consists of:

- **Human brains:** 10 triangulated surfaces of human brains, each surface supported by approximately 900 vertices.

- **Synthetic boxes:** 12 triangulated surfaces of synthetically constructed boxes, varying in height, width, depth, bending and extrusion of specific corner, each surface supported by approximately 1200 vertices.

Evaluation of spherical mapping as a stand-alone algorithm was conducted based on Euclidean distances between corresponding points in combination with area-distortion measurement between corresponding triangles.

A higher degree of visual evaluation has been employed in the evaluation process, due to the graphical nature of spherical mappings.

To obtain quantitative results on MDL model performance a set of leave-one-out tests on the standard and combined models were conducted on both training sets. This methodology allows for most shape variation to be captured, composing a good evaluation criteria.

Note, that all computations are performed using relative coordinates, rather than actual pixel values. This ensures consistency across different measures. Data ranges from 0.0 to 1.0.

6.2 Validation Techniques

6.3 Common Definitions

The following error measurement is used throughout the evaluation of both spherical mapping, standard as well as combined MDL model. For this reason, a common definition is given here:

- **RMS Error:** Deviations between two corresponding elements is defined as the root-mean-square error, based on differences between ground truth and the test object.

  Example wise calculation of Euclidian distance between two corresponding landmarks, is done by

  $$d_i(\mathbf{x}_i, \mathbf{x}_{gt,i}) = \sqrt{(x_i - x_{gt,i})^2 + (y_i - y_{gt,i})^2 + (z_i - z_{gt,i})^2} \tag{6.1}$$

  Thus calculating the RMS point-to-point error is accomplished by

  $$d(\mathbf{x}, \mathbf{x}_{gt}) = \frac{1}{n}\sum_{i=1}^{n} d_i(\mathbf{x}_i, \mathbf{x}_{gt,i}) \tag{6.2}$$

  A similar set of expressions can be formed for other types of deviation.

Since we are interested in the actual deviation from ground truth, we can use this relatively simple measurement as a valid scale.

6.3.1 Spherical Mapping

The spherical mapping algorithm aims at minimizing area-distortion between original and mapped shapes in addition to the enforcement of uniqueness. Therefore a quantitative assessment of performance is based on the following measures.

- **RMS Point-to-Point Error:** Error measurement based on inter-point distances between corresponding landmarks, using a mean shape estimate as ground truth.

- **RMS Triangle-to-Triangle Error:** Error measurement based on inter-triangle area differences between corresponding triangles, using the original shape as ground truth.

Point-to-Point error measurements are performed on a set of spherical maps constructed by mapping each shape in the training set a predefined number of times using randomly selected reference triangles, $\sigma = ABC$, and calculating the mean error, while Triangle-to-Triangle error measurements are calculated using the original training set as ground truth.

6.3.2 Minimum Description Length Shape Model

Using a model composed by uniformly distributing a number of landmarks across each shape in the training set, performance of the MDL model can easily be assessed. The reference model is composed using the same approach as in building the MDL shape model, though without the MDL optimization step. This enables the performance gain in the MDL optimization approach to be measured quantitatively as simple differences.

Recall, we wish to construct an optimal shape model, in the sense that it is both variance minimizing and general, i.e. expresses greater ability to describe unseen shapes. To evaluate these properties, the following measurements are employed

- **RMS Error:** Error measurement based on inter-point distances between corresponding landmarks, using a mean shape estimate as ground truth.

- **Cumulative Variance:** Measurement of model compactness, using both a standard MDL model as reference as well as the standard PCA model as ground truth.

6.3.3 Summary

Evaluation of performance has been divided into two categories, one concerning spherical mapping, one concerning the MDL shape model. As a common measurement of "goodness", the RMS error definition is employed in variations depending on the actual objective being measured. For the latter category, the addition of a cumulated variance analysis scheme, composes a compactness measurement.

Chapter 7

Spherical Mapping

"Computers are useless. They can only give you answers."
– Pablo Picasso

7.1 Overview

This section treats the tests conducted on both training sets as a whole.

For each shape in the training set, 100 consecutive maps were constructed, and a mean estimate was calculated from each set of maps. The deviations listed in this section are mean values based on RMS errors calculated between each group of shape and mappings, for each training set. Three different weight factors, ω, were used.

7.2 Results

The basic conformal mapping of a shape of genus 0 to the unit sphere imposes some degree of area-distortion. Furthermore, since the mapping

is conformal, the resulting spherical map is very much dependant upon the choice of north pole, and reference triangle.

From tables 7.1 and 7.2 it can be seen that the minimization scheme introduced, yields a map which displays a significant decrease in overall area distortion while improving on consistency, thus minimizing average point-to-point distances regardless of choice of north pole, p.

Mean Point-to-Point RMS Errors

No.	Data Set	ω	RMS Error
1	Brains	1.0	$3.47e^{-5}$
2	Brains	0.5	$4.20e^{-5}$
3	Brains	[0.5 : 1.0]	$3.75e^{-5}$
4	Brains	-	$8.68e^{-5}$
5	Boxes	1.0	$2.03e^{-5}$
6	Boxes	0.5	$2.94e^{-5}$
7	Boxes	[0.5 : 1.0]	$2.39e^{-5}$
8	Boxes	-	$5.70e^{-5}$

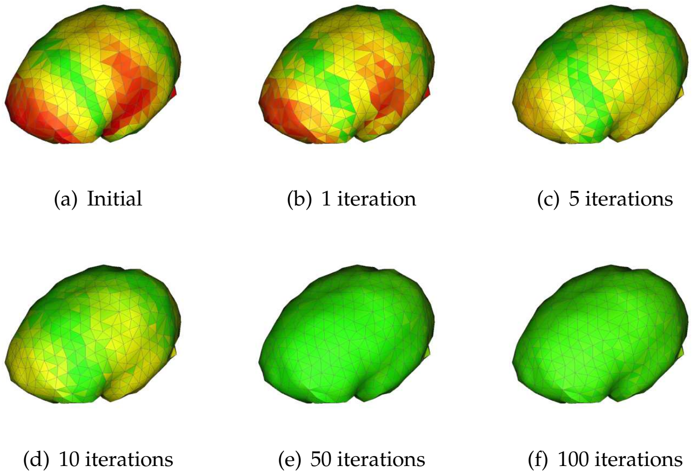
Table 7.1: Mean Point-to-Point RMS errors between the set of mappings and their common mean estimate.

Mean Area RMS Errors

No.	Data Set	ω	RMS
1	Brains	1.0	$1.25e^{-7}$
2	Brains	0.5	$2.86e^{-7}$
3	Brains	[0.5 : 1.0]	$2.29e^{-7}$
4	Brains	-	$17.9e^{-7}$
5	Boxes	1.0	$1.02e^{-7}$
6	Boxes	0.5	$2.63e^{-7}$
7	Boxes	[0.5 : 1.0]	$1.72e^{-7}$
8	Boxes	-	$6.97e^{-7}$

Table 7.2: Mean Area RMS errors between the set of mappings and their respective original shapes.

Behavior of the optimization scheme and the visual effect of area-distortion minimization is best displayed graphically using the illustration in figure 7.1. In this example, a conformal map of a single human brain surface is subjected to area-distortion minimization. The coloration expresses the degree of distortion, with red being more and green less distortion[1].


(a) Initial

(b) 1 iteration

(c) 5 iterations

(d) 10 iterations

(e) 50 iterations

(f) 100 iterations

Figure 7.1: Area distortion minimization per iteration. Red triangles display a high degree of area distortion as opposed to green triangles.

Though an improvement compared to the original algorithm, some issues still remain. First of all since the optimization approach uses a 2D approximation important area-information is lost, thus resulting in a non-optimal minimization of area-distortion. Secondly, since both area-preservation and conformality cannot be realized in the same map, either angle or area information is disregarded, forcing the resulting shape description to be an approximation rather than a complete description. These factors contribute to the non-zero RMS error values.

The algorithm tends to produce large artifacts, if the initial conformal mapping displays large area-distortion. Again, this failure tendency is accredited to the 2D approximation. This could be eliminated by imposing an edge constraint on the minimization scheme.

Differences in weight factor ω allows for non- or semi-optimal steps to be taken in the minimization process. In figure 7.2 a comparison between the

[1] In gray scale terms one seeks a homogenous distribution of "colors".

effect of different types of weights is illustrated.

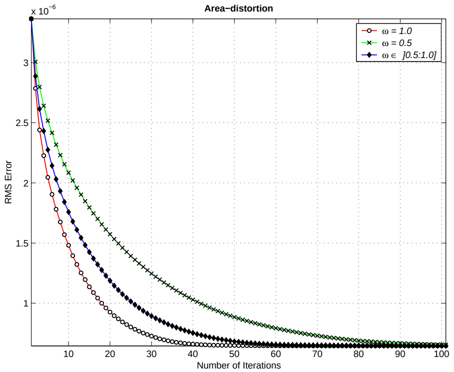


Figure 7.2: RMS Error for different values of weight ω as a function of the optimization step.

Tables 7.1 and 7.2 should also be noted here, where examples of ω parameter usage and corresponding results are listed. An examination of the ω parameters effect on the combined MDL shape model is presented in the succeeding chapter.

7.3 Summary

The spherical mapping algorithm has been successfully used to map both biological as well as synthetical shapes to the unit sphere, an minimize area-distortion on these maps. Albeit issues still remain: The neglecting of 3D information in minimization of area-distortion forcing an error term upon the resulting map.

For additional pictorial documentation of the experiments conducted, refer to appendix A.1.

Chapter 8

Minimum Description Length Model

"Perfect numbers like perfect men are very rare."
– René Descartes

8.1 Overview

The MDL model was quantitatively evaluated using a leave-one-out scheme against a uniformly sampled shape model. The algorithm was run for three levels of recursion, yielding a total of 66 landmarks for each shape. In addition a cumulative variance analysis was performed to evaluate compactness of the resulting model. In all tests, the effect of using weights and area-distortion minimization scheme were examined.

8.2 A Note on Objective Functions

A choice was made early in the implementation process to incorporate the simple version of the objective function (4.16), primarily due to its simplicity and effectiveness. This choice would however prove to have significant effect on the MDL optimization scheme, since it favorizes a regular logarithmic distribution of the principal components as opposed to a distribution composed using the full objective function. This is due to the fact that by applying the same weight to all eigenvalues, less significant principal components may have high influence on evaluation of the objective function. The simple objective function will however prove the effect of using area-distortion minimization of spherical mapping on the actual MDL model.

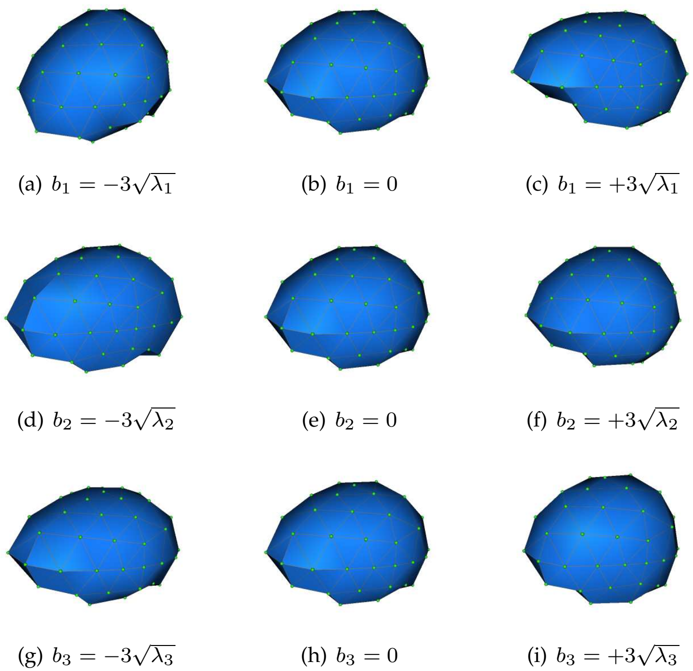
8.3 Results

For reasons of limiting this chapter to a confined space, only images from the human brain data set are presented here. Wherever dissimilarities are present between the two sets of training data, relevant graphs and images from both data sets will be displayed. For further pictorial documentation of both cases, refer to appendix A.2.

A qualitative assessment of the model ability to capture shape variations is shown in figure 8.1, where the three most significant principal components are visualized in shape variations. Variations for the i^{th} component is displayed within the eigenvalue interval of

$$-3\sqrt{\lambda_i} \leq b_i \leq 3\sqrt{\lambda_i} \tag{8.1}$$

Differences in model deformations due to different choices of weight, ω, result in little visible variation across the model, hence no visualization of this effect will be given here, though all four ω values used in the experiments succeeded in capturing variations seen across the training data.



(a) $b_1 = -3\sqrt{\lambda_1}$ (b) $b_1 = 0$ (c) $b_1 = +3\sqrt{\lambda_1}$

(d) $b_2 = -3\sqrt{\lambda_2}$ (e) $b_2 = 0$ (f) $b_2 = +3\sqrt{\lambda_2}$

(g) $b_3 = -3\sqrt{\lambda_3}$ (h) $b_3 = 0$ (i) $b_3 = +3\sqrt{\lambda_3}$

Figure 8.1: Human Brains: Deformations of the mean shape along the first three principal axes.

For a quantitative assessment of model compactness, attention is turned to tables 8.1 and 8.2 in which variances are listed sorted by their contribution to the total variance. A graphical display of tables 8.1 and 8.2 is presented in figure 8.2.

Human Brains

Automatic				
Mode	$\omega = 1.0$	$\omega = 1.0$	$\omega \in [0.5 : 1.0]$	–
1	$186e^{-7}$	$152e^{-7}$	$150e^{-7}$	$163e^{-7}$
2	$65.8e^{-7}$	$72.4e^{-7}$	$78.3e^{-7}$	$68.4e^{-7}$
3	$63.9e^{-7}$	$70.1e^{-7}$	$62.9e^{-7}$	$48.0e^{-7}$
4	$22.1e^{-7}$	$21.4e^{-7}$	$18.5e^{-7}$	$13.2e^{-7}$
5	$13.0e^{-7}$	$11.7e^{-7}$	$14.9e^{-7}$	$9.90e^{-7}$
6	$10.5e^{-7}$	$10.2e^{-7}$	$7.86e^{-7}$	$5.91e^{-7}$
σ_T	$375e^{-7}$	$351e^{-7}$	$345e^{-7}$	$317e^{-7}$
F	84	84	84	81
Uniform				
Mode	$\omega = 1.0$	$\omega = 0.5$	$\omega \in [0.5 : 1.0]$	–
1	$206e^{-7}$	$189e^{-7}$	$167e^{-7}$	$137e^{-7}$
2	$96.1e^{-7}$	$96.3e^{-7}$	$89.6e^{-7}$	$81.0e^{-7}$
3	$65.2e^{-7}$	$64.4e^{-7}$	$63.5e^{-7}$	$50.3e^{-7}$
4	$28.6e^{-7}$	$27.9e^{-7}$	$32.4e^{-7}$	$48.9e^{-7}$
5	$17.4e^{-7}$	$16.5e^{-7}$	$17.5e^{-7}$	$17.2e^{-7}$
6	$12.9e^{-7}$	$12.4e^{-7}$	$12.4e^{-7}$	$10.0e^{-7}$
σ_T	$443e^{-7}$	$420e^{-7}$	$396e^{-7}$	$357e^{-7}$
F	86	85	85	84

Table 8.1: Human Brains: Variance explained by each mode of variation. F is the value of the objective function and σ_T is the total variance.

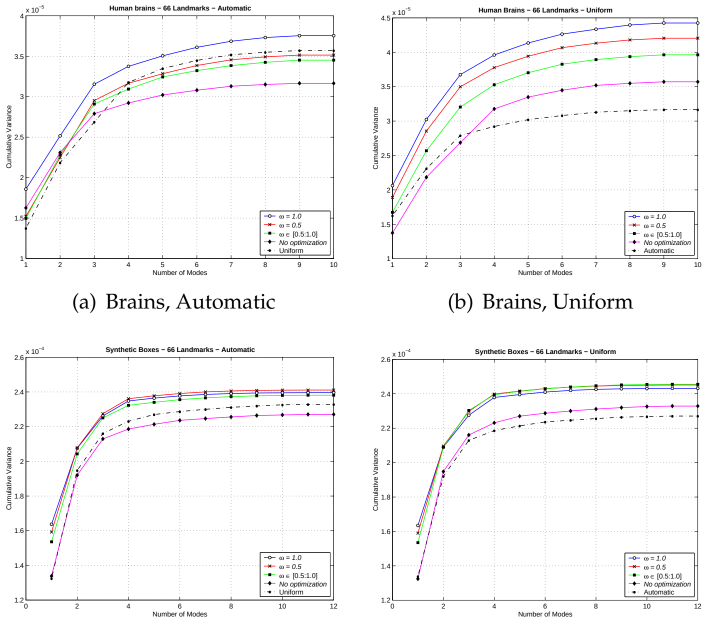
The addition of an area-distortion minimization scheme has introduced an additional component to the shape model, namely that of noise contributed by points sliding across the surface. Since the standard MDL model in this case uses a conformal mapping, each with very similar reference triangles due to a good initial registration using ICP, the addition of noise to the shape model is causing the total variance to climb, resulting in a less compact model. This effect is best displayed graphically in figure 8.2. In this illustration, the black dotted line represents the best value from the counter example, *Automatic* vs *Uniform*.

Synthetic Boxes

Mode	$\omega = 1.0$	$\omega = 1.0$	$\omega \in [0.5:1.0]$	$-$
Automatic				
1	$1638e^{-7}$	$1592e^{-7}$	$1536e^{-7}$	$1339e^{-7}$
2	$438e^{-7}$	$485e^{-7}$	$507e^{-7}$	$581e^{-7}$
3	$186e^{-7}$	$197e^{-7}$	$208e^{-7}$	$209e^{-7}$
4	$86.6e^{-7}$	$86.1e^{-7}$	$72.3e^{-7}$	$56.8e^{-7}$
5	$16.7e^{-7}$	$16.8e^{-7}$	$17.3e^{-7}$	$27.5e^{-7}$
6	$12.5e^{-7}$	$13.2e^{-7}$	$15.3e^{-7}$	$22.6e^{-7}$
σ_T	$2396e^{-7}$	$2411e^{-7}$	$2381e^{-7}$	$2271e^{-7}$
F	69	68	70	73
Uniform				
Mode	$\omega = 1.0$	$\omega = 0.5$	$\omega \in [0.5:1.0]$	$-$
1	$1634e^{-7}$	$1590e^{-7}$	$1535e^{-7}$	$1324e^{-7}$
2	$456e^{-7}$	$506e^{-7}$	$554e^{-7}$	$624e^{-7}$
3	$185e^{-7}$	$202e^{-7}$	$214e^{-7}$	$213e^{-7}$
4	$103e^{-7}$	$101e^{-7}$	$91.4e^{-7}$	$70.7e^{-7}$
5	$17.0e^{-7}$	$17.5e^{-7}$	$19.9e^{-7}$	$38.9e^{-7}$
6	$14.7e^{-7}$	$14.0e^{-7}$	$14.9e^{-7}$	$17.1e^{-7}$
σ_T	$2432e^{-7}$	$2451e^{-7}$	$2455e^{-7}$	$2328e^{-7}$
F	70	69	71	74

Table 8.2: Synthetic Boxes: Variance explained by each mode of variation. F is the value of the objective function and σ_T is the total variance.

Note that in some cases, the uniformly distributed and un-optimized model displays less variance in important principal components, than the optimized model. This is due to the usage of a objective function which favors a logarithmic distribution of eigenvalues. In this manner, small eigenvalues are emphasized since they contribute more to the objective function. Had the objective function imposed a different weight on different types of eigenvalues, a model with less variance across the entire range of principal components could be extracted.



(a) Brains, Automatic

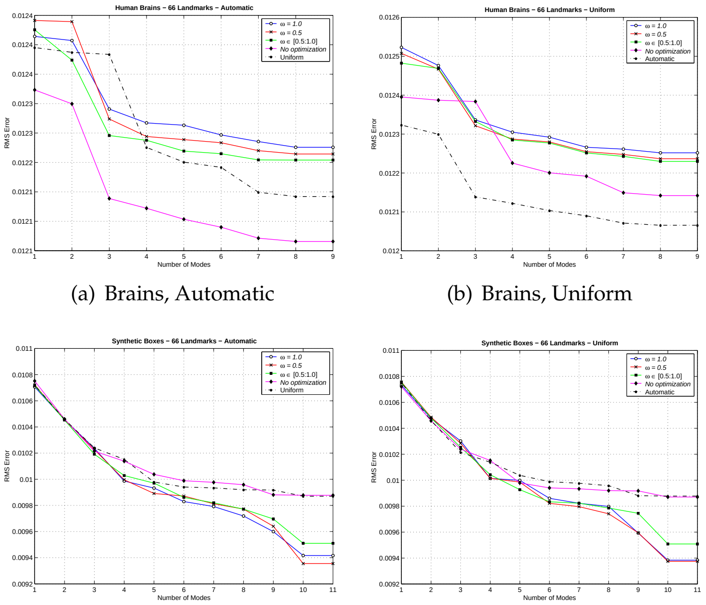
(b) Brains, Uniform

(c) Boxes, Automatic

(d) Boxes, Uniform

Figure 8.2: Cumulative variance described by each principal component using different weight parameters ω. *Left:* MDL optimized model. *Right:* Uniformly sampled model.

The leave-one-out test yielded an assessment of the MDL shape models ability to model unseen examples, based on prior knowledge. This type of assessment is also referred to as *self-contained validation*.



(a) Brains, Automatic

(b) Brains, Uniform

(c) Boxes, Automatic

(d) Boxes, Uniform

Figure 8.3: Mean RMS errors between an unseen example and a synthetical regeneration of the example using mean shape and a deformation vector derived from principal component analysis of the model. *Left:* MDL optimized model. *Right:* Uniformly sampled model.

An apparent difference between the human brain and synthetic box training sets is the result of employing the area-distortion minimization scheme. Again this fact is partly due to the disregard of variation orthogonal to the optimization plane used in the local minimization method. Since the synthetic box training set displays little or no local variation orthogonal to the large surface segments, the amount of information disregarded by the minimization scheme is roughly the same. This on the other hand has more fatal consequences for the human brain training set, which, as must be expected, displays a higher degree of local variations.

Added to this factor, is the ability to distribute landmarks more evenly across each surface, which especially in the box case, enforces a more descriptive model, since corners are better represented with a homogenous point density. It should even be possible to extract a more compact model by hand from the synthetic box training set, just by selecting corner and edge points.

8.3.1 The ω Parameter

All tests were conducted using four different values of ω, the weight by which each minimization step in the area-distortion minimization scheme is affected. Again this value produces a different outcome depending on the type of shapes being spherically mapped. In general it must be said, that this parameter is best set to a random value within a fixed interval with non-zero minimum, if the data set expresses variations orthogonal to the surface, i.e. "points jumping up and down". The random value mimics this effect better than a fixed value, since the fixed value will tend to smoothen out these orthogonal variations. In contrast a fixed value should be used, if the data set being mapped displays little or no variation of the mentioned type. This will force the map to disregard an even amount of information for all points on the same surface segment.

The effects of varying the ω parameter depending on type of training set is best illustrated using the cumulative variance plots in figure 8.2.

8.4 Summary

A series of tests have been conducted using a combination of area-distortion minimized spherical maps and the standard MDL model. The usage of area-distortion removal on maps constructed using the conformal mapping scheme described in chapter 2 have shown to have a significant effect on the resulting model, though dependant on the actual data set being modelled. This is primarily due to errors imposed by area-distortion minimization scheme. In particular the increased uniformization of landmark distribution due to area-distortion removal has proved to yield positive results in synthetic data sets.

For additional pictorial documentation of the experiments conducted, refer to appendix A.2.

Part IV

Discussion

Chapter 9

Propositions for Further Work

"God made the integers, all else is the work of man."
– Leopold Kronecker

9.1 Overview

The following section introduces improvements and suggestions to the original algorithms as well as to the enhancements contributed during the 8 month thesis work, which were either out of scope or out of reach within the given time span.

9.2 Edge Constraints on Area-preserving map

The optimization of a conformal mapping may in some cases fail due to the following facts:

- Transformation of the problem into 2 dimension, thus disregarding the z-component.
- Optimization of the map is based solely on an area measurement, thus disregarding edge length distortion

The argument for transforming the area optimization problem into one of 2D is to simplify the deriviation of a feasible optimization scheme. Recall the 2D transformation scheme from section 2.3.2. If the shape being mapped displays large variation across triangle areas, one could easily imagine a vertex $\mathbf{v}$ and its apparent neighborhood which on the mapped unit sphere will represent large variations orthogonal to the surface, thus deferring from the assumption that

$$\theta \approx \sin(\theta) \tag{9.1}$$

A novel optimization scheme should introduce calculation of areas based on 3D coordinates, deriving a vertex transformation along the surface of a unit sphere. However this would increase complexity since a linear system of equations would no longer suffice, due to the geometrical nature of a sphere.

The latter argument involves introducing a new objective function to measure edge length distortion on triangles in domain Σ. Based on work in the field of conformal maps conducted by Quicken and Brechbühler et al. [39], [5], a new objective function, f, to be minimized is suggested

$$\begin{aligned} \min_{f} \sum_{\sigma\in\Sigma} & \left(\frac{A_{map,\sigma}}{4\pi} - \frac{A_{orig,\sigma}}{\sum_{\sigma\in\Sigma} A_{orig,\sigma}} \right)^2 \\ +\alpha & \left(\left(\frac{\mathbf{a}_\sigma \cdot \mathbf{b}_\sigma}{\|\mathbf{a}_\sigma\| \cdot \|\mathbf{b}_\sigma\|} \right)^2 + \beta \left| \frac{\|\mathbf{a}_\sigma\| \cdot \|\mathbf{a}_\sigma\| - \|\mathbf{b}_\sigma\| \cdot \|\mathbf{b}_\sigma\|}{\|\mathbf{a}_\sigma\| \cdot \|\mathbf{a}_\sigma\| + \|\mathbf{b}_\sigma\| \cdot \|\mathbf{b}_\sigma\|} \right| \right) \end{aligned} \tag{9.2}$$

where $\mathbf{a}_\sigma$ and $\mathbf{b}_\sigma$ are orthogonal legs on triangle σ, $A_{map,\sigma}$ and $A_{orig,\sigma}$ are areas on original and mapped triangle, α and β are arbitrary constants. The first term punishes area distortion, while the second term punishes deviations from right angle and unequal orthogonal edge lengths. Since (9.2) is defined for right angle triangles, a deriviation hereof would be necessary.

9.3 Robust MDL Model

Though the general MDL approach is sensible to initial starting conditions, the algorithm could be extended to allow non-minimizing steps in order to avoid falling into local minima. One could imagine an approach as follows:

Allow the MDL model to propagate from an unaligned starting point by employing a hierarchical Metropolis type minimization scheme [35].

Start off with a small number of landmarks and employ the MDL algorithm.

1. Perform MDL minimization step
2. Calculate the probability for choosing minimization step based on the residual between object function evaluations.
3. If the probability is below 1.0, choose a random number from the interval $[0, 1[$.
4. If the random number generated is higher than the calculated probability, the step is rejected, otherwise accepted.
5. Repeat steps 1-4 until convergence.

Calculating the probability of taking a specific step can be expressed as

$$P(X = x_n) = \frac{1}{z}\exp(-F(x_n)) \tag{9.3}$$

Thus the relative probability of taking a specific step, x_n, given the previous step, x_{n-1} can be defined as

$$P = min\left(1, \frac{P(X = x_n)}{P(X = x_{n-1})}\right) \tag{9.4}$$

where the latter expression is the relation between two successive evaluations of the objective function, F.

$$\begin{aligned} \frac{P(X = x_n)}{P(X = x_{n-1})} &= \frac{\exp(-F(x_n))}{\exp(-F(x_{n-1}))} \\ &= \exp(F(x_{n-1}) - F(x_n)) \\ &= \exp(-\partial F) \end{aligned} \tag{9.5}$$

From (9.4) and (9.5) we construct

$$P = min(1, \exp(-c\,\partial F)) \tag{9.6}$$

Where c is an arbitrary positive constant weight. Once a certain level of convergence is achieved, the rotation and translation between each set of fixed and modelled landmarks could be determined from the result reached with the sparse landmark representation. The final steps include applying the derived transformation to each mapping, and performing the MDL steps in the common sense, now using a dense landmark representation.

The approach suggested in this section was implemented and put through initial tests. Although some adjustments may be required, the results seen so far seem promising.

9.4 Extended Shape Representation

This section is merely a reminder to the fact, that although the method presented derives an optimal shape model, the use of spherical maps restricts to shapes of genus 0, in other words shapes which have topological equivalence with a sphere. A novel mapping scheme, based on cylindrical maps employing a latitude and longitude coordinate representation, could be employed to extend capabilities of the MDL method. Such conformal cylindrical mapping forms the base of a visualization technique used in non-evasive endoscopy [21].

The general MDL approach does not restrict to the spherical case.

Chapter 10

Discussion

"A proof tells us where to concentrate our doubts."
– Morris Kline

10.1 Summary

Throughout the first parts of this thesis, spherical mapping and MDL algorithms have been described and documented in detail, aided by illustrative examples, discussions on each subject and references for further investigation, thus fulfilling the first objective set forth in this thesis.

The original algorithms have been extended and further suggestions on extensions have been presented.

10.1.1 Extensions to Spherical Mapping

An area-preserving optimization scheme, based on deterministic relaxation, has been proposed. The approach employs a local vertex optimization method, simplifying the problem by transformation into 2D and solving a set of linear equations, thus deriving vertex coordinates minimizing

area distortion between mapped and original shapes. The approach has been put through vigorous tests using both biological and synthetical data displaying positive results. Though the algorithm minimizes distortion between corresponding triangles, a certain level of error is imposed on the original shape, due to the transformation into 2D and thereby disregarding important shape information.

10.1.2 Extensions to MDL

Landmark extraction based on uniformly sampled spheres form the base for the set of landmarks derived in the MDL model building process. The landmark sampling method proposed by Davies et al., has the disadvantage of placing points on the sphere in specific grouped patterns, becoming increasingly apparent for each level of recursion. To counter this fact, a new subdivision scheme has been employed, enforcing a smooth distribution across the sphere.

A combination of area-preserving spherical maps and the standard MDL shape model has been introduced and tested. For shapes of certain character, this mapping technique shows promising results due to its ability to distribute landmarks smoothly across a synthetic surface. The combined model was also tested on a training set showing biological variations, which it failed to represent sufficiently precise, primarily due to the disregard of variations seen orthogonal to the surface. This type of variation should be expected on biological objects, and was present at a higher degree than was the case with the synthetic training set.

The use of a simplified objective function has been tested and evaluated. Though the function does not succeed to fully minimize the resulting shape model, due its inability to favor significant rather than less significant components, it provides a good estimate on final optimization using the full objective function as proposed by Davies et al [14].

A scheme for unregistered MDL modelling using a hierarchical Metropolis type sampling scheme has been suggested, although further work in this field is still required.

This concludes the fulfillment of the second and third objectives set forth.

10.2 Conclusion

In the field of shape analysis, consistency and compactness are key figures, in particular as dimensionality of data being processed increases. In this thesis a combination of methods leading to a compact and in an information theoretic sense optimal model has been explored and extended.

The work conducted in the duration of this project has been naturally divided into two separate parts, one regarding spherical mapping techniques, the other shape model optimization. A natural link between these parts exist, due to the utilization of spherical maps in minimum description length shape model building.

In this thesis, a general approach on automated model building has been presented. The theoretical foundation involved in building such model, based on spherical maps, has been thoroughly studied, documented and subsequently extended.

By imposing an area-preservation criteria on the initial conformal spherical map, area-distortion has been significantly minimized, while retaining characteristics of the original shape.

In addition to the incorporation of area-preserving spherical maps, the minimum description length model has been extended by the inclusion of initial registration using iterative closest point, uniformly distributed landmark configuration and simplification of objective functional.

Though results using biological data sets did not yield the sought results, the reasons of failure have been identified for further exploration. In the synthetic case the ability to distribute landmarks uniformly, through improved subdivision and area-preservation, yielded a model capable of better reproduction of unseen shape examples.

Implementation of theoretical algorithms resulted in a high performance C++ class collection, bound together by a console application, and by the introduction of several open source class libraries.

The general conclusion is that automated model building can be achieved using the minimum description length approach, in collaboration with spherical maps as a means of solving the correspondence problem. Of particular importance is the choice of spherical mapping technique, due to a uniqueness requirement.

Bibliography

[1] S. Angenent, S. Haker, A. Tannenbaum, and R. Kikinis. On the Laplace-Beltrami Operator and Brain Surface Flattening. In *IEEE Transactions on Visualization and Computer Graphics*, pages 700–711. IEEE Computer Society, August 1999.

[2] J. Besag. On the Statistical Analysis of Dirty Pictures. *Journal of the Royal Statistical Society*, 48:259–302, 1986.

[3] P. J. Besl and N. D. McKay. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and machine Intelligence*, 14(2):239–258, February 1992.

[4] F. Bookstein. Landmark methods for forms without landmarks: localizing group differences in outline shape. pages 279–289, San Francisco, CA, USA, June 1996. IEEE Computer Society.

[5] C. Brechbühler, G. Gerig, and O. Kübler. Parametrization of closed surfaces for 3-d shape description. *Computer Vision and Image Understanding*, 61:154–170, March 1995.

[6] A. Brett and C. Taylor. A method of automated landmark generation for automated 3d pdm construction. In *Proceedings of the British Machine Vision Conference*, pages 914–923. British Machine Vision Association, 1998.

[7] A. Caunce and C. J. Taylor. 3D Point Distribution Models of the Cortical Sulci. In S. Chandran and U. Desai, editors, *Procedings of the Sixth International Conference on Computer Vision (ICCV-98)*, pages 402–407, New Delhi, January 1998. Narosa Publishing House.

[8] K. Conradsen. *En Introduktion til Statistik*, volume 2B. IMSOR, Technical University of Denmark, Lyngby, Denmark, 4th edition, 1984.

[9] T. F. Cootes and C. J. Taylor. *Statistical Models of Appearance for Computer Vision*. Wolfson Image Analysis Unit, Department of Biomedical Engineering, University of Manchester, July 2000.

[10] T. 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, 1995.

[11] R. H. Davies. *Learning Shape: Optimal Models for Analysing Natural Variability*. PhD thesis, University of Manchester, October 2002.

[12] R. H. Davies, T. F. Cootes, and C. J. Taylor. A Minimum Description Length Approach to Statistical Shape Modelling. In 17^{th} *Conference on Information Processing in Medical Imaging*, pages 50–63, 2001.

[13] R. H. Davies, C. J. Twining, P. D. Allen, T. F. Cootes, and C. J. Taylor. Building Optimal 2D Statistical Shape, in press.

[14] R. H. Davies, C. J. Twining, T. F. Cootes, J. C. Waterton, and C. J. Taylor. 3D Statistical Shape Models Using Direct Optimisation of Description Length. In 7^{th} *European Conference on Computer Vision*, pages 3–20, 2002.

[15] R. H. Davies, C. J. Twining, T. F. Cootes, J. C. Waterton, and C. J. Taylor. A Minimum Description Length Approach to Statistical Shape Modelling. In *IEEE Transactions on Medical Imaging*, volume 21, pages 525–537, May 2002.

[16] I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis*. Wiley Series in Probability and Statistics. John Wiley & Sons, Chichester, West Sussex, England, 1999.

[17] Nira Dyn, David Levin, and John A. Gregory. A Butterfly Subdivision Scheme for Surface Interpolation with Tension Control. *ACM Transactions on Graphics*, 9(2):160–169, April 1990.

[18] J. Eising. *Lineær Algebra*. Matematisk Institut, Technical University of Denmark, Lyngby, Denmark, 1st edition, 1993.

[19] S. Farlow. *Partial Differential Equations for Scientists and Engineers*. 1982.

[20] J. C. Gower. Generalized Procrustes analysis. *Psychometrika*, 4:33–51, 1975.

[21] S. Haker, S. Angenent, A. Tannenbaum, and R. Kikinis. Nondistorting Flattening for Virtual Colonoscopy. In *MICCAI*, pages 358–366, 2000.

[22] M. Halstead, M. Kass, and T. DeRose. Efficient, fair interpolation using Catmull-Clark surfaces. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 35–44. ACM Press, 1993.

[23] B. K. P. Horn. Closed Form Solution of Absolute Orientation using

Unit Quaternions. *Journal of the Optical Society A*, 4(4):629–642, April 1987.

[24] T. J. R. Hughes. *The Finite Element Method*. Dover Publications, Inc., 31 East 2nd Street, Mineola, New York, USA, 2000.

[25] A. Hyvärinen. Survey on Independent Component Analysis. *Neural Computing Surveys*, 2:94–128, 1999.

[26] K. Kanatani. Analysis of 3-D rotation fitting. *IEEE Transactions on Pattern Analysis and machine Intelligence*, 16(5):543–549, May 1994.

[27] M. Kaus, V. Pekar, C. Lorenz, R. Truyen, S. Lobregt, J. Richolt, and J. Weese. Automated 3D PDM Construction Using Deformable Models. In *Proceedings of the Eighth International Conference On Computer Vision (ICCV-01)*, pages 566–572, Los Alamitos, CA, July 2001. IEEE Computer Society.

[28] A. C. W. Kotcheff and C. J. Taylor. Automatic Construction of Eigenshape Models by Direct Optimization. *Medical Image Analysis*, 2:303–314, 1998.

[29] S. G. Krantz. *Handbook of Complex Analysis*. Birkhäuser, February 1999.

[30] E. Kreyszig and E. J. Normington. *Advanced Engineering Mathematics*. John Wiley, New York, USA, 1993.

[31] R. Larsen and H. Eiriksson. Robust and Resistant 2D Shape Alignment. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, September 2001.

[32] C. T. Loop. Smooth Subdivision Surfaces Based on Triangles. Master's thesis, Department of Mathematics, University of Utah, August 1987.

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

[34] Merriam-Webster. *Merriam-Webster's Collegiate Dictionary*. Merriam-Webster, Inc., Springfield, USA, 10th edition, January 1998.

[35] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[36] J. A. Nelder and R. Mead. A Simplex Method for Function Minimization. *The Computer Journal*, 7(4):308–313, January 1965.

[37] E. Polak. *Computational Methods in Optimization*, volume 72 of *Mathematics in Science and Engineering*. Academic Press, New York, 1971.

[38] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C : The Art of Scientific Computing*. Cambridge University Press, 2nd edition, 1993.

[39] M. Quicken, C. Brechbühler, J. Hug, H. Blattman, and G. Székely. Parameterization of Closed Surfaces for Parametric Surface Description. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2000*, volume 1, pages 354–360, Los Alamitos, June 2000. IEEE Computer Society.

[40] J. Rissanen. Modelling by shortest data description. *Automatica*, 14:465–471, 1978.

[41] W. J. Schroeder, K. M. Martin, L. S. Avila, and C. C. Law. Kitware, Inc., May 2001.

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

[43] B. Sharp. Subdivision Surface Theory. *Game Developer*, January 2000.

[44] M. B. Stegmann. Active Appearance Models: Theory, Extensions and Cases. Master's thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, August 2000.

[45] P. Switzer and A. A. Green. Min/max autocorrelation factors for multivariate spatial imagery. *Computer Science and Statistics*, 13–16, 1985.

[46] J.-P. Thirion. The Extremal Mesh and the Understanding of 3D Surfaces. In *IEEE Workshop on Biomedical Image Analysis*, pages 3–12, Seattle, USA, June 1994.

[47] J.-P. Thirion and A. Gourdon. The 3D Marching Lines Algorithm and its Application to Crest Lines Extraction. Technical Report RR-1672, Inria, Institut National de Recherche en Informatique et en Automatique, May 1992.

[48] D. Tosun and J. L. Prince. A Hemispherical Map for the Human Brain Cortex. In *SPIE Medical Imaging*, pages 113–116, February 2001.

[49] Y. Wang, B. Peterson, and L. Staib. Shape-based 3D Surface Correspondence using Geodesics and Local Geometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR-00)*, pages 644–651, Los Alamitos, June 2000. IEEE Computer Society.

[50] P. Yiu. The uses of homogeneous barycentric coordinates in plane euclidean geometry. *International Journal of Mathematical Education in Science and Technology*, 31(4):569–578, July 2000.

[51] Z. Zhang. Iterative point matching for registration of free-form curves. Technical Report RR-1658, Inria, Institut National de Recherche en Informatique et en Automatique, April 1992.

[52] D. Zorin, P. Schröder, and W. Sweldens. Interpolating Subdivision for

Meshes with Arbitrary Topology. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 189–192. ACM SIGGRAPH, Addison Wesley, August 1996.

Index

Appendix A

Additional Results

The following pages contain additional pictorial documentation related to the experiments conducted. For ease of explanation, the material has been divided into two sections, one for each related algorithm.

A.1 Spherical Mapping

In the following pages spherical mapping is documented based on one example shape from each data set by illustrations of:

- Original shape.
- Conformal map.
- Area-preserving map, $\omega = 1.0$.
- Area-distortion removal per iteration.

A.1.1 Human Brains

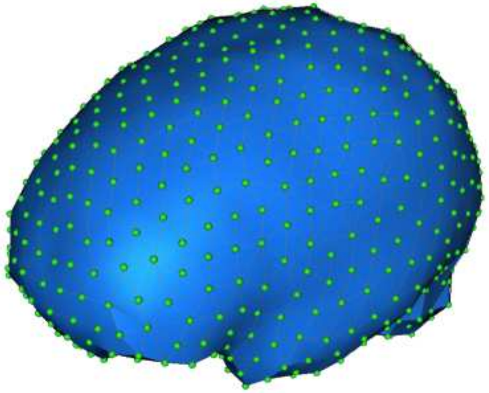


Figure A.1: Human Brain: Original shape.

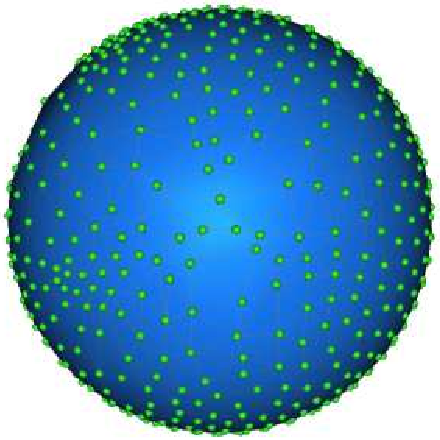


Figure A.2: Human Brain: Conformal map.

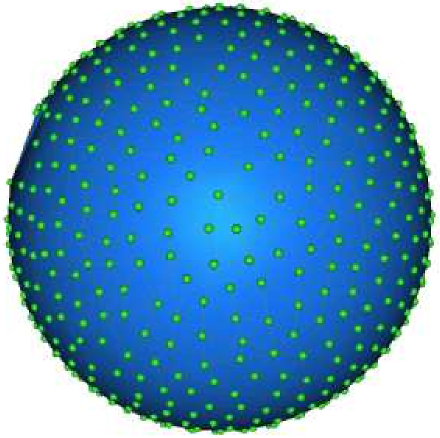


Figure A.3: Human Brain: Area-preserving map.

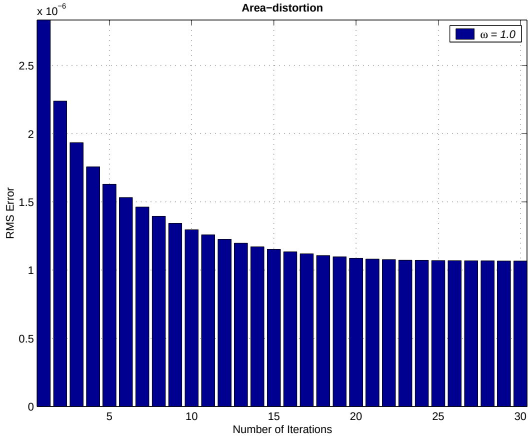


Figure A.4: Human Brain: Area-distortion removal per iteration.

A.1.2 Synthetic Boxes

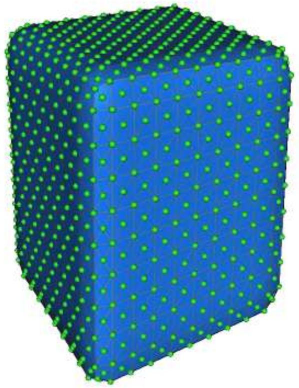


Figure A.5: Synthetic Box: Original shape.

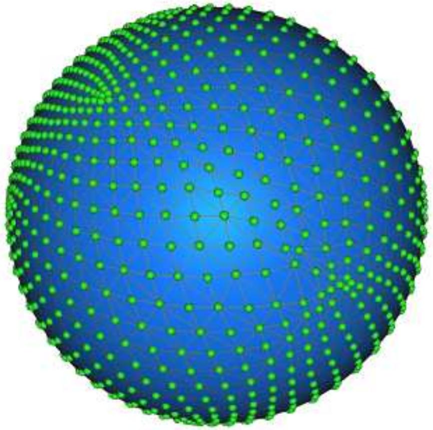


Figure A.6: Synthetic Box: Conformal map.

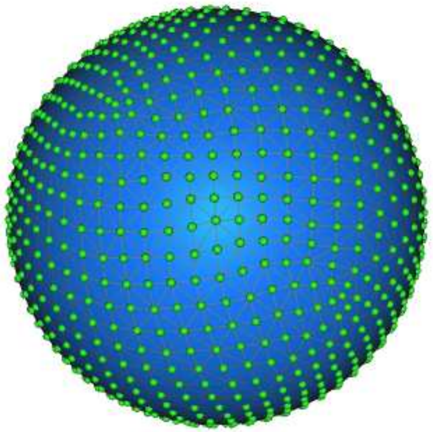


Figure A.7: Synthetic Box: Area-preserving map.

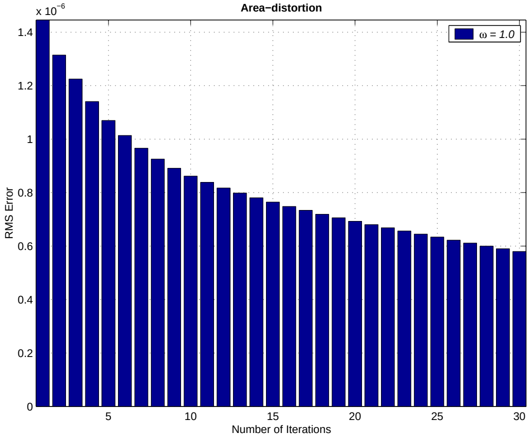


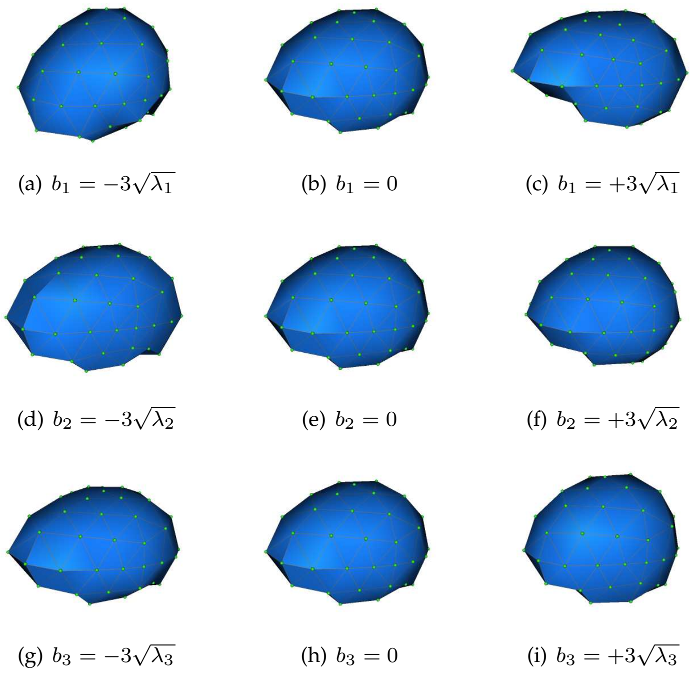
Figure A.8: Synthetic Box: Area-distortion removal per iteration.

A.2 Minimum Description Length Shape Model

In the following pages MDL shape modelling is documented for each data set by illustrations of:

- Mean shape deformation along 1st, 2nd and 3rd principal axis.
- Cumulative variance in uniformly sampled model.
- Cumulative variance in MDL model.
- Mean RMS errors in uniformly sampled model.
- Mean RMS errors in MDL model.

A.2.1 Human Brains



(a) $b_1 = -3\sqrt{\lambda_1}$ (b) $b_1 = 0$ (c) $b_1 = +3\sqrt{\lambda_1}$

(d) $b_2 = -3\sqrt{\lambda_2}$ (e) $b_2 = 0$ (f) $b_2 = +3\sqrt{\lambda_2}$

(g) $b_3 = -3\sqrt{\lambda_3}$ (h) $b_3 = 0$ (i) $b_3 = +3\sqrt{\lambda_3}$

Figure A.9: Human Brains: Deformations of the mean shape along the first three principal axes.

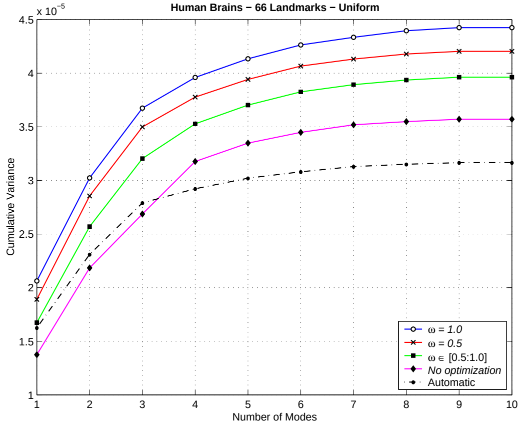


Figure A.10: Human Brains: Cumulative variance described by each principal component in the uniformly sampled model, using different ω-values.

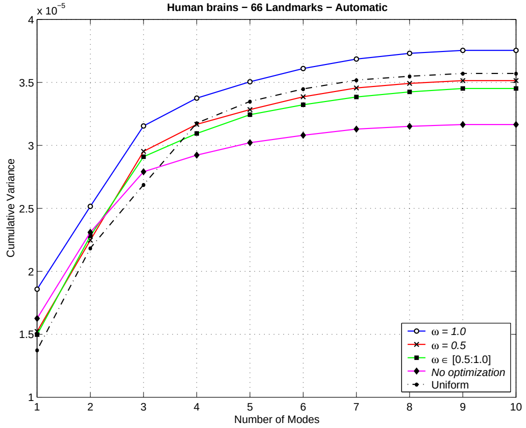


Figure A.11: Human Brains: Cumulative variance described by each principal component in the MDL model, using different ω-values.

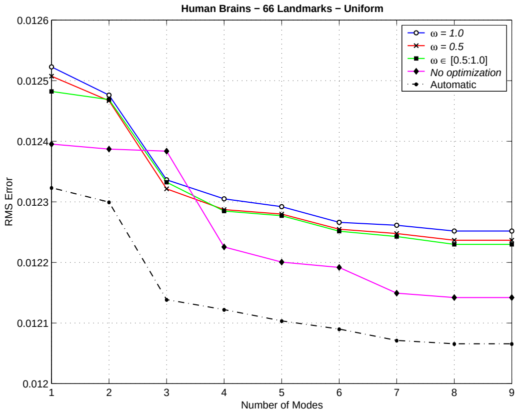


Figure A.12: Mean RMS errors between an example and synthetical regeneration in the uniformly sampled model.

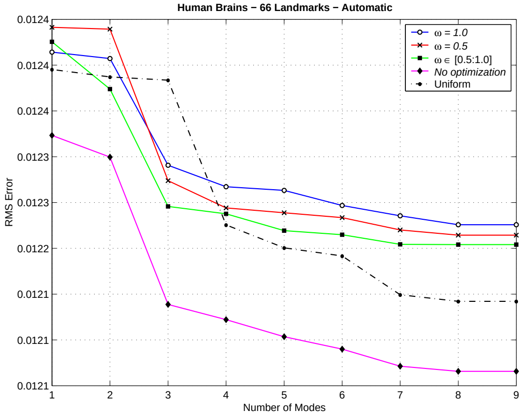
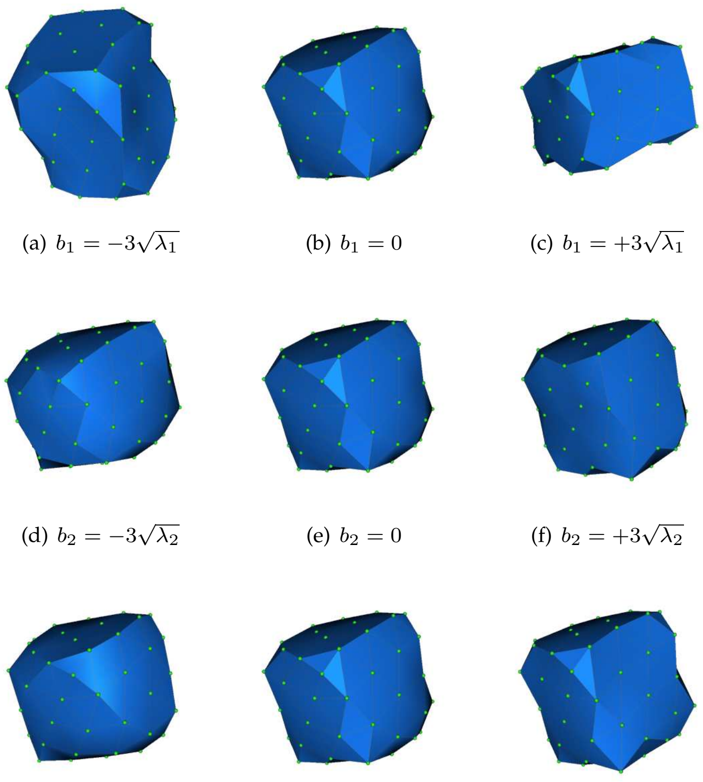


Figure A.13: Mean RMS errors between an example and synthetical regeneration in the MDL model.

A.2.2 Synthetic Boxes



(a) $b_1 = -3\sqrt{\lambda_1}$

(b) $b_1 = 0$

(c) $b_1 = +3\sqrt{\lambda_1}$

(d) $b_2 = -3\sqrt{\lambda_2}$

(e) $b_2 = 0$

(f) $b_2 = +3\sqrt{\lambda_2}$

(g) $b_3 = -3\sqrt{\lambda_3}$

(h) $b_3 = 0$

(i) $b_3 = +3\sqrt{\lambda_3}$

Figure A.14: Synthetic Boxes: Deformations of the mean shape along the first three principal axes.

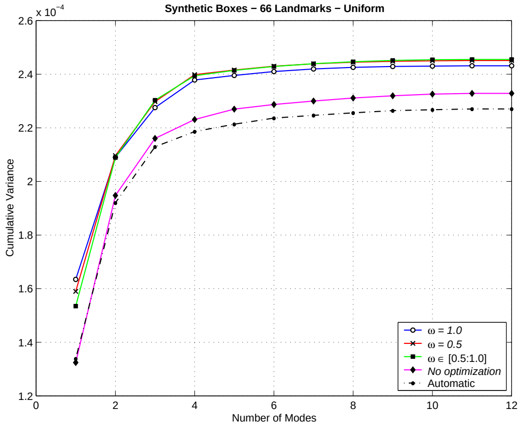


Figure A.15: Synthetic Boxes: Cumulative variance described by each principal component in the uniformly sampled model, using different ω-values.

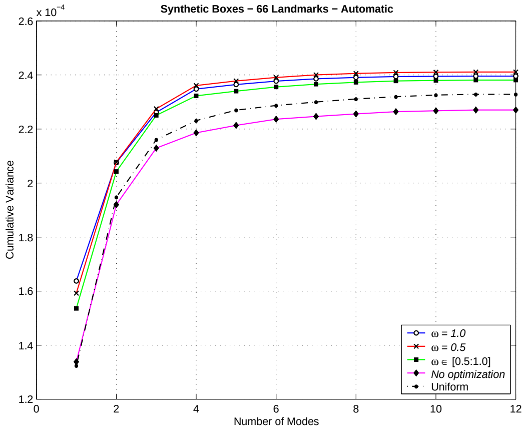


Figure A.16: Synthetic Boxes: Cumulative variance described by each principal component in the MDL model, using different ω-values.

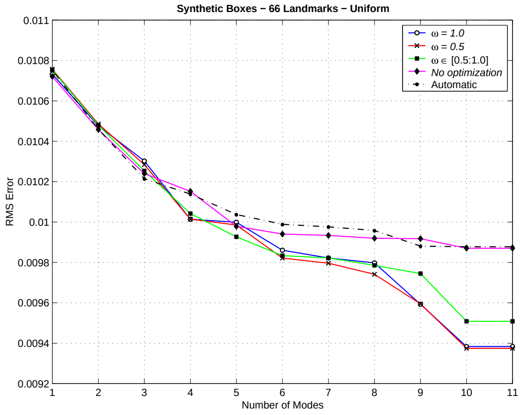


Figure A.17: Synthetic Boxes: Mean RMS errors between an example and synthetical regeneration in the uniformly sampled model.

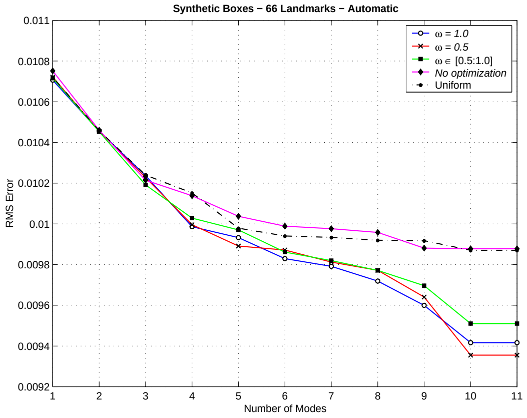


Figure A.18: Synthetic Boxes: Mean RMS errors between an example and synthetical regeneration in the MDL model.

Appendix B

SM^2-API Console Interface Usage

```
3DSMC - 3D Sherical Mapping/Shape Modelling Console Interface - version 1.0
Copyright (c) Allan Reinhold Kildeby 2002 - All rights reserved.

USAGE:
       3dsmc <configuration file>

DESCRIPTION:
3D shape model building using spherical mapping and minimum description
length. Input data is mapped to the unit sphere and a number of landmarks
is placed across each sphere. Based upon evaluation of an objective
function, landmarks are optimized until an optimal shape model is derived.

OUTPUT:
Output is written to mapping<n>.vtk, landmarksphere<n>.vtk and
landmarks<n>.vtk in the output directory specified in the XML
configuration file.

XML PARAMETERS:
   prg_extension        : Extension used by polygonal data files.
                          Ex: "VTK"
   prg_directory        : Data directory to read polygonal data from.
                          Ex: "D:\Data\"
   icp_landmarks        : Number of landmarks used by ICP algorithm.
                          Ex: "100"
   icp_iterations       : Maximum number of iterations used by ICP
                          algorithm.
                          Ex: "100"
   icp_tolerance        : ICP algorithm in aborted when the residual
                          between two iterations is below this tolerance.
                          Ex: "1e-6"
   mts_method           : Spherical mapping method.
```

```
                          "0" = Simple, "1" = Area-preserving.
   mts_cgiterations     : Maximum number of iterations used by conjugate
                          gradient method in spherical mapping.
                          Ex: "1000"
   mts_cgtolerance      : Conjugate gradient is aborted when the residual
                          between two iterations is below this tolerance.
                          Ex: "1e-15"
   mts_bcoord_xdim      : Number of cuts along x-axis when creating cut
                          shapes.
                          Ex: "5"
   mts_bcoord_ydim      : Number of cuts along y-axis when creating cut
                          shapes.
                          Ex: "5"
   mts_bcoord_zdim      : Number of cuts along z-axis when creating cut
                          shapes.
                          Ex: "5"
   mts_opt_tolerance    : Area-preserving optimization of spherical
                          mapping is aborted when the residual between
                          two iterations is below this threshold.
                          Ex: "1e-7"
   mts_opt_fileprefix   : File prefix used when writing area-preserving
                          minimization steps to image files.
                          Ex: "optimization_step"
   mts_opt_iterations   : Maximum number of iterations used by area-
                          preserving minimization of spherical mapping.
                          Ex: "100"
   mts_opt_enablenoise  : Enable weighted area-preserving minization of
                          spherical mapping.
                          "0" = disabled, "1" = enabled.
   mts_opt_noise        : Noise type used in weighted area-preserving
                          minization of spherical mapping.
                          "0" = uniform weight, "1" = random weight.
   mts_opt_weight       : Maximum weight used in weighted area-preserving
                          minization of spherical mapping.
                          Ex: "0.5"
   mts_opt_enableoutput : Enable stepwise images generation in area-
                          preserving minization of spherical mapping.
                          "0" = disabled, "1" = enabled.
   ann_pca_quantile     : Percentage of variations to be included in the
                          resulting shape model.
                          Ex: "0.95"
   ann_mdl_landmarks    : Number of landmarks used in the MDL shape model.
                          Ex: "50"
   ann_mdl_delta        : Delta value used in evaluation of MDL objective
                          function to avoid infinitely large
                          contributions.
                          Ex: "1e-8"
   ann_mdl_quantization : Delta value used in evaluation of MDL objective
                          function to avoid infinitely large
                          contributions.
                          Ex: "1e-8"
   ann_mdl_iterations   : Maximum number of iterations used by MDL
                          algorithm.
                          Ex: "100"
   ann_mdl_tolerance    : MDL optimization algorithm is aborted when the
                          residual between two iterations is below this
                          tolerance.
                          Ex: "0.001"
```

```
ann_mdl_enableoutput : Enable stepwise images generation in MDL
                       minimization algorithm.
                       "0" = disabled, "1" = enabled.
ann_mdl_fileprefix   : File prefix used when writing MDL minimization
                       steps to image files.
                       Ex: "mdl_optimization_step"
ui_output_dir        : Directory to write output data to.
                       Ex: "D:\Output\"
ui_window_dimension_x: X-dimension of output window.
                       Ex: "512"
ui_window_dimension_y: Y-dimension of output window.
                       Ex: "512"
```