Anatomical Standardization of the Human Brain in Euclidean 3-Space and on the Cortical 2-Manifold

Steven M. Robbins

School of Computer Science
McGill University, Montreal

October 2003

A thesis submitted to McGill University in partial fulfilment of the requirements of the degree of Doctor of Philosophy.

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-612-98359-5
Our file Notre référence
ISBN: 0-612-98359-5

Abstract

Anatomical standardization (also called spatial normalization) is a key process in cross-sectional studies of brain structure and function using MRI, fMRI, PET and other imaging techniques. This process has two components: (i) specification of a 3D template brain, which defines a common coordinate space for analysis of any subsequent datasets; and (ii) a method to align the template with an individual 3D brain image, thereby associating each point of the standard template to a point on the individual. The association should be able to consistently match a particular template location to an anatomically corresponding location on each individual of a population.

Standardization methods in widespread use employ a 3D affine spatial transformation to map from the individual to the template, which matches only overall size and gross shape of the input brain. A wide range of more flexible image deformation algorithms have been developed in order to better match fine detail. All such algorithms involve design choices that are subject to debate, and most have numerical parameters whose value must be specified by the user. In order to provide guidance for such choices, the first part of this thesis develops two measures of alignment consistency that are used to evaluate performance of a standardization method. The performance of different choices for algorithm design, numerical parameters, and template selection strategy for 3D normalization are compared.

Since the processing of brain function occurs on a thin, highly convoluted sheet of cortex along the surface of the brain, there has been much recent interest in studying the structure and function along the brain cortex only, modelled as a 2D manifold. The second part of this thesis proposes an algorithm for highly-flexible deformation in 2D of a template cortex to an individual. The alignment consistency measures developed for 3D are reformulated for the 2D manifold and used to evaluate the algorithm design and numerical parameters. Finally, the question of whether it is better to standardize the 3D images or the 2D cortical manifold is addressed, identifying the problem classes which are best suited to each type of normalization.

Précis

La standardisation anatomique (également appellée normalisation spatiale) est un processus-clé dans les études trans-sectionnelles de structure et de fonctions cérébrales en imagerie par résonance magnétique (IRM), IRM fonctionnelle, (IRMf), tomographie par émission de positron (TEP) ainsi que d'autres techniques d'imagerie. Ce processus est composé de deux parties: (i) la spécification d'un cadre de réference tri-dimensionnel (3D), qui définit un espace de coordonnées communes pour l'analyse subséquentes des données; et (ii) une méthode pour aligner ce cadre avec l'image 3D d'un cerveau individuel. Cette dernière permet d'associer à chaque point du cadre de réference commun un point dans un cerveau individuel. Cette association devrait être capable de mettre en correspondance de façon constante un endroit particulier du cadre à un endroit anatomique correspondant et ce, pour chaque cerveau dans une population donnée.

Les méthodes de standardisation les plus employées utilisent une transformation spatiale affine en 3D pour mettre en correspondance le cerveau d'un individu avec le cadre de réference commun. Cette méthode ne fait qu'ajuster la taille totale et la forme globale du cerveau. Un large éventail d'algorithmes de déformation d'images, plus flexibles, ont été développés pour faciliter l'ajustement de détails locaux. Tous ces algorithmes impliquent des choix méthodologiques qui sont sujet à débat, et la plupart ont des paramètres numériques dont les valeurs doivent être spécifiées par l'usager. Pour pouvoir guider ces choix, deux méthodes de mesure de la constance de l'alignement, utilisées pour évaluer la performance de la méthode de standardisation, ont été développées et sont présentées dans la première partie de cette thèse. La performance de différents choix méthodologiques pour les algorithmes, de différents paramètres numériques et de stratégie de sélection du cadre de référence commun pour la normalisation 3D est ensuite comparée.

Puisque le processus de l'activité cérébrale se déroule sur une mince couche de cortex très plissée le long de la surface du cerveau, un grand intérêt existe dans l'étude de la structure et de l'activité le long du cortex cérébral, modélisé comme une variété 2D. La deuxième partie de cette thèse propose un algorithme de déformation 2D ultra-flexible d'un cadre de référence cortical commun à un cortex individuel. Les mesures de constance de l'alignement développées en 3D sont reformulées pour la variété 2D et utilisées pour évaluer la conception de l'algorithme ainsi que ses paramètres numériques. Finalement, la question à savoir si il est mieux de standardiser des images 3D ou une variété corticale 2D est présentée, en identifiant les classes de problèmes pour lesquelles chaque type de normalisation serait le mieux adapté.

Acknowledgements

I'd like to thank, first of all, my co-supervisors Sue Whitesides and Alan Evans. This thesis would not exist without their guidance and support over the years.

This thesis builds on the wealth of software created at the MNI Brain Imaging Centre, specifically the MINC tools developed by Peter Neelin, the surface extraction program, ASP, of David MacDonald, and the 3D registration code, ANIMAL, of Louis Collins. I'm grateful to them all, for sharing both their code and their insights. Thanks also are due to Jean-François Malouin, Sylvain Milot, and Dale Einarson for keeping the computing lab going over the years, even under the strain of the thousands of experiments that this thesis represents.

I owe a great debt to Jurgen Germann, who generously shared his painstaking labelling of the frontal lobe sulci. Additional sulcal data kindly supplied by Arnaud Cachia and Jean-François Mangin was a great help.

Above all, it is the people, both in computer science and at the BIC who make these departments great places in which to work. I've had informative and enjoyable discussions with many in the lab, notably Jason Lerch, Alex Zijdenbos, Chris Cocosco, Serge Dumoulin, Andrew Janke, Vivek Singh, Richard Webster, Yasser Ad-Dab'bagh, and Oliver Lyttelton. I'm grateful to them all.

Finally, I'd like to thank my family for their understanding and for trying hard not to ask "is it done yet?" too often during the final stages. Most appreciated of all during my long sojourn has been the love, support, and patience of my wife, Sutanuka Bhattacharya.

Contents

List of Figures

List of Symbols

$f : A \longrightarrow B$ Notation for function f that maps elements of set A to set B.

$f : a \longmapsto b$ Function f maps a to b; i.e. $f(a) = b$.

$f|_X$ Function obtained by restricting f to a subset X of its domain.

I, J Real-valued, volumetric images, e.g. from magnetic resonance imaging. Mathematically, image I is idealized as a mapping from some domain $\mathcal{D}_I \subset \mathbb{R}^3$ to $\mathbb{R}$. The domain is generally a rectangle, e.g., $\mathcal{D}_I = [0, 256]^3$.

x, y Elements of $\mathbb{R}^3$. Subscripts are used for coordinates, i.e., $x = (x_1, x_2, x_3)$.

$||x||$ The L^2 norm of x, $||x|| = \sqrt{x_1^2 + x_2^2 + x_3^2}$.

T A spatial mapping. In 3 dimensions, $T : \mathbb{R}^3 \longrightarrow \mathbb{R}^3$. In 2 dimensions, T maps from one 2-manifold to another.

Δ 3D deformation field, defined by $\Delta(x) = T(x) - x$. Subscripts are used for the coordinate functions $\Delta(x) = (\Delta_1(x), \Delta_2(x), \Delta_3(x))$.

$\nabla f, \nabla T$ For scalar function, $f : \mathbb{R}^3 \longrightarrow \mathbb{R}$, ∇f is the gradient defined as $\nabla f(x) = (\frac{\partial f}{\partial x_1}(x), \frac{\partial f}{\partial x_2}(x), \frac{\partial f}{\partial x_3}(x))$. For spatial mapping, $T = (T_1, T_2, T_3)$ with each $T_i : \mathbb{R}^3 \longrightarrow \mathbb{R}$, ∇T is the Jacobian matrix,

$$\nabla T(x) = \begin{pmatrix} \frac{\partial T_1}{\partial x_1}(x) & \frac{\partial T_1}{\partial x_2}(x) & \frac{\partial T_1}{\partial x_3}(x) \\ \frac{\partial T_2}{\partial x_1}(x) & \frac{\partial T_2}{\partial x_2}(x) & \frac{\partial T_2}{\partial x_3}(x) \\ \frac{\partial T_3}{\partial x_1}(x) & \frac{\partial T_3}{\partial x_2}(x) & \frac{\partial T_3}{\partial x_3}(x) \end{pmatrix}.$$

Φ An objective function, generally a functional of two images and the spatial mapping written $\Phi(I, J, T)$.

$\arg\min_x \Phi(x, y)$ A value of parameter x that minimizes the expression $\Phi(x, y)$.

$\mathbb{S}^2$ The 2-sphere, generally considered as the point set $\{x \in \mathbb{R}^3 : ||x|| = 1\}$ embedded in $\mathbb{R}^3$.

$\circ$ Functional composition; $h = f \circ g \Leftrightarrow h(x) = f(g(x))$ for all x in the domain of g.

∂A Boundary of set A.

$\mathbf{Int}\, A$ Interior points of set A.

$A \setminus B$ Set difference, $A \setminus B = \{a \in A : a \notin B\}$.

$\mathsf{P}(a)$ Probability of event a.

$\mathsf{E}X$ Expectation value of random variable X.

$\lfloor x \rfloor$ For $x \in \mathbb{R}$, $\lfloor x \rfloor$ is the largest integer $\leq x$.

Chapter 1

Introduction

This thesis addresses a thorny and continuing problem arising from the field of brain mapping, specifically the problem of comparing functional or structural characteristics across a population of brains. Such comparisons are best carried out in a reference frame for which the normal anatomical variability in the population has been removed, to the extent possible. The aims of this thesis are therefore: (1) to develop and apply tools that quantify the anatomical variability, whether the reference frame is a three dimensional Euclidean space or a two dimensional manifold; (2) to develop an automatic method for anatomical standardization of the two dimensional brain surface; and (3) to compare the anatomical variability of the brain remaining after standardization using 3D methods or using 2D methods.

While this chapter describes the problem in the context of comparing functional data, the issues of anatomical standardization are largely the same when the data to be compared is structural in nature. The comparisons are carried out on populations of hundreds or thousands of brains, so this work emphasizes fully-automated solutions.

1.1 Brain Mapping

The main aim of brain mapping is to understand the functional anatomy in the human brain [TM96, Sav01]. Though brain tissue looks homogeneous to the naked eye, by the 1950's enough was known from studying patients with brain lesions and from direct stimulation of the brain during surgery to conclude that regions of the brain specialize in processing of particular functions and to draw reasonably detailed summary maps of brain function [Sav01]. The mental processing happens in neuron bodies that accumulate in a thin layer of *gray matter* on the surface of the brain, termed the *cerebral cortex*, while the bulk of the brain inside the cortex is *white matter*, which is composed of fibres that carry signals between regions of the cerebral cortex [Hei95]. While there are also some gray matter structures found deep inside the brain, the main interest of *functional brain mapping* is to describe the location and extent of functionally-specialized regions of the cortex.

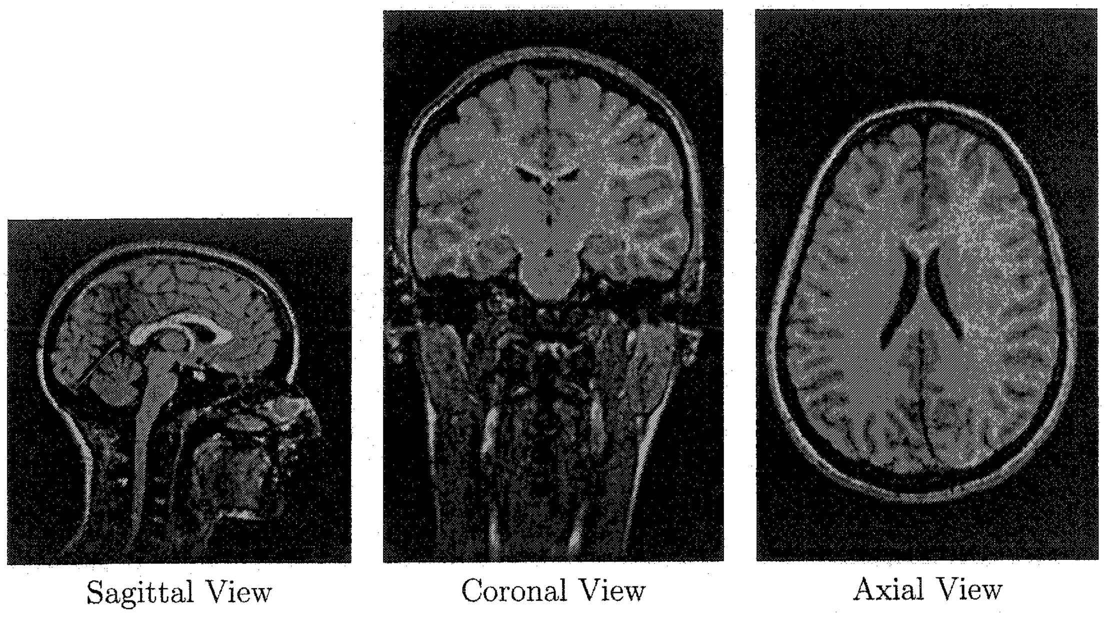
Today, the study of brain function is largely carried out by imaging living brains [Sav01].

1.1.1 Magnetic Resonance Images

The dominant method for studying the brain *in vivo* is using *Magnetic Resonance Imaging* (MR imaging) [DSFL$^+$99]. The MR scanner produces a 3D array of intensity values known as a *volumetric image*, or simply "an image". Throughout this thesis, the unqualified word "image" will mean a volumetric image. The array elements are known as "volume elements" or *voxels*, in analogy with pixels ("picture elements") of a 2D digital image.

The value produced at each voxel by an MR scanner is related to the density of hydrogen nuclei (mainly from water molecules) at that location. By manipulating parameters of the scan, the signal can be modulated by other characteristics of the tissue. In particular, the signal can be weighted by the so-called "T_1" and "T_2" relaxation time constants [Nis96]. These time constants differ amongst the tissue types of the brain, so images can be formed that have different intensity values for the different tissue types. Figure 1.1 shows slices from an MR image in the three standard orthogonal views: sagittal (from the side), coronal (from the back), and axial (from above).

In T_1-weighted images, a common type of MR image, three main tissue types of the brain can be distinguished: white matter, gray matter, and cerebral spinal fluid (CSF). The white matter appears brightest, the CSF darkest, and gray matter intensity lies between the two as shown in Figure 1.2. Anatomically, the folds of the cortex form very deep crevices known as *sulci* (singular sulcus). The fold of tissue between two sulci is known as a *gyrus* (plural gyri). While the pattern of folding is highly variable between individuals [OKA90], some sulci and gyri are consistent enough across the population that they can be used as features for identifying corresponding anatomy in different brains. In particular, some folds generally delimit boundaries between cortical regions of different *cytoarchitecture* (cell structure) [RCSG93], which may coincide with boundaries between different functional regions.



Sagittal View

Coronal View

Axial View

Figure 1.1: Example slices from a magnetic resonance image, illustrating the three standard views of the brain.

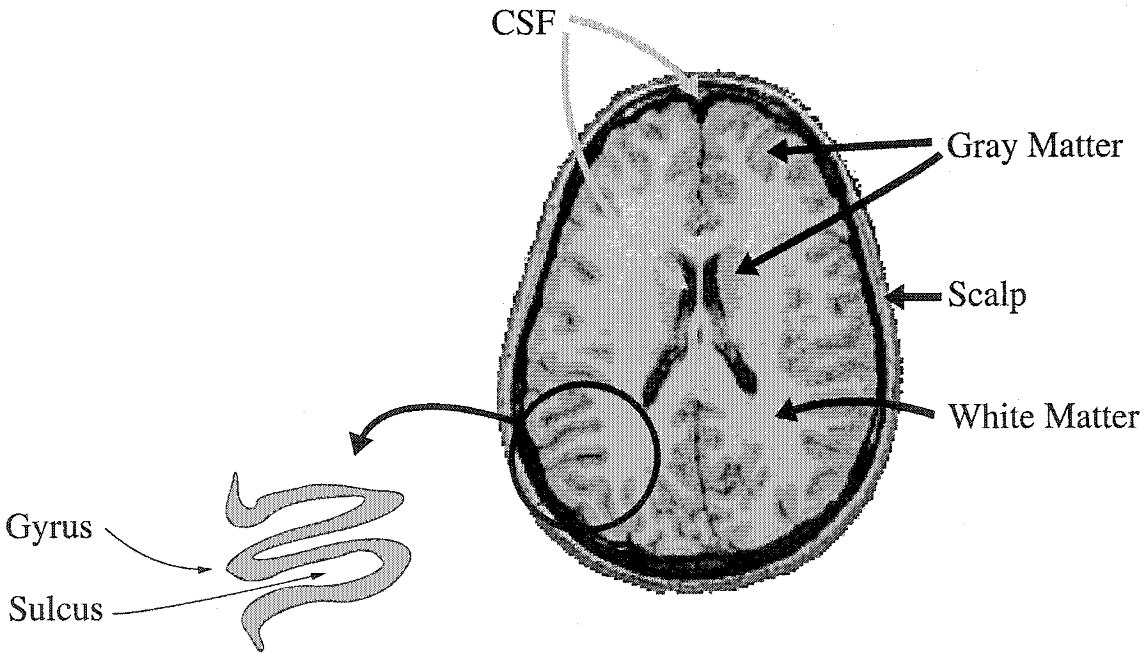


Figure 1.2: Brain tissue types visible in a T_1-weighted image. Crevices in the brain are called sulci. A gyrus is the fold of tissue that lies between two sulci.

1.1.2 Spatial Normalization

In the context of brain mapping, *spatial normalization* (also referred to as *anatomical standardization* [MKFK94, SIO$^+$98]) refers to "selective removal of inter-individual anatomical variance" [Fox95]. The goals and applications of spatial normalization are also discussed by, e.g. Ashburner and Friston [FAF$^+$95, AF99b], Toga and Thompson [TT01], Collins et al. [CZPE01]. In practice, the term has come to refer to imposing a standard coordinate system on the brain by mapping each brain image into a standard space [AF99b].

The spatial map between each individual brain and some standardized reference system is established so that variability in location or signal strength of some spatially-varying data can be quantified across subjects. After spatially normalizing the population, the data to be compared at each point in the standardized reference system is the data from the corresponding point in each individual. The data to be quantified can be either functional, in which the signal is related to brain activity, or structural, in which the data is an anatomical measure such as the thickness of the cortex. The important characteristic of this mapping strategy is that anatomically corresponding, i.e. *homologous*, locations in each subject brain be mapped to the same standardized coordinates so that the data being compared is associated with corresponding anatomy.

One of the original motivations of spatial normalization for examining brain function comes from studies that use Positron Emission Tomography (PET) to image the brain. PET images are formed by injecting a radioactive tracer into the subject's bloodstream and analyzing the resulting radiation to obtain the pattern of blood flow in the brain, which can be related to brain activity [CP96]. Due to dosage limitations, the images obtained from a single subject are not always adequate to detect activation so data from several individuals is combined to increase the overall signal-to-noise ratio. The data is combined by averaging the intensity values at each point in the standardized space [FMRR88, FAF$^+$95, AF97, AF99b]. It is thus important that each standard space location maps to the homologous location in each individual.

Currently, functional imaging is more commonly performed using MR, a technique known as *functional magnetic resonance imaging*, or *f*MRI [DSFL$^+$99]. Functional MRI is sensitive enough that activation can be detected in a single subject, so averaging across subjects is not needed to detect the signal [AF99b]. However, it may be that different individuals perform a given task differently, perhaps using a different anatomical location. After spatially normalizing the population, the spatial

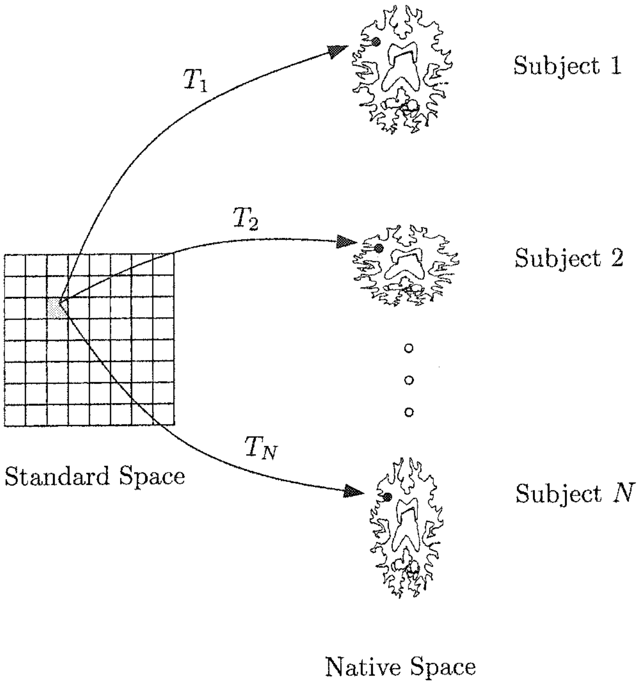


Figure 1.3: The spatial normalization problem is to find the corresponding location in each subject image, for each point in standard space. The result is a set of mappings $\{T_i : i = 1, \ldots, N\}$ from standard space to the native space of each subject, where standard coordinate x corresponds to point $T_i(x)$ in the brain of subject i.

distribution of the functional signal can be examined in standard space, i.e. with respect to a standardized anatomy.

Finally, spatial normalization is useful for communicating results between labs. The location of a functional region determined during a study is often reported using standardized coordinates [BJO02]. The neuroimaging community generally reports locations using so-called Talairach coordinates, and there is interest in the neuropsychology community to do the same for lesion studies [RB01].

Talairach Coordinates

Talairach defines a piecewise-affine mapping procedure to standardize brains for surgical purposes [TST+67]. Based on the plane separating the two cortical hemispheres and two structures known as the anterior commissure (AC) and posterior commissure (PC), the bounding box of the brain is subdivided into 12 boxes, each of which is independently affinely mapped to a standard location, orientation, and size. Talairach coordinates are 3D Cartesian coordinates associated with a particular template brain image [TT88], although different conventions are used for the location of the origin [Woo96] so the coordinates are not always comparable. Moreover, automated and semiautomated standardization methods may instead use a global affine transform (as it is more convenient to work with than Talairach's piecewise transformation), but still designate the results as "Talairach coordinates" [Woo96].

Transformation Function

A correspondence between an individual brain and a standardized space is conveniently described in terms of a spatial transformation function, generically denoted as T throughout this thesis. It is most convenient to define T as mapping **from** the standardized space **to** the space of the individual brain, which is referred to as the *native space* of the individual. When the mapping is defined in this manner (rather than from the native space to the standard space), data can be transferred to the standard space without any gaps on the standard space sampling grid [Jäh99] and without needing to invert the function T. This data mapping procedure is further elucidated in Sections 4.1 and 7.1. The spatial normalization problem is to automatically find the function T_i corresponding to subject i, for $i = 1, \ldots, N$ as illustrated in Figure 1.3.

It is presumed that neighbouring points in the standard space should have homologous points that are neighbours in each subject, implying that the mapping T should be a continuous function [CRM$^+$95]. If the biological homology between brains extended to arbitrarily small spatial scales, then each location in standard space would map to its unique homologous point in a subject brain, meaning that T should be injective, i.e. a one-to-one mapping. However, it is not expected that brains correspond down to each individual neuron, so at some point the homology breaks down. Indeed, the variation in brain topography is rather large, even at the scale of sulcal folds [OKA90]. Moreover, in the data analysis done after spatial normalization the data being compared is smoothed in order to enhance the signal-to-noise ratio [Wor97] and to compensate for imperfect spatial normalization [AF97, BJO02]. Thus a comparison at point x in the standard space is really not a point-to-point comparison with data precisely at $T_i(x)$ in each subject i, but a comparison of data in some region about $T_i(x)$. A certain amount of small-scale noninjectivity can therefore be tolerated in the transformations used for spatial normalization.

1.2 3D Image Registration

Automated spatial normalization of brain images is mainly accomplished through image registration, by aligning the image with a template image that is already in the standardized coordinate system [AF99b]. The sensitivity of a comparison of spatially-varying data (structural or functional) is enhanced by reducing, to the extent possible, the natural anatomical variability in a population. This is the aim of practical spatial normalization. The effectiveness of spatial normalization depends on the template image and the registration method chosen. One of the themes of this thesis is defining performance measurements used to select between competing template images and between competing registration algorithms.

Early spatial normalization approaches, e.g. [FPR85, EMN+92] take their cue from Talairach's normalization procedure, using an affine or piecewise-affine transformation to map each brain into standard space. Such a mapping procedure corrects only for location, orientation, and overall size of the input brains, leaving much anatomical variability [SFF89]. However, even these fairly rigid transformations align individual brains sufficiently to bring about an increase in sensitivity when used prior to combining functional measurements from different individuals [FMRR88].

Normalizing with more flexible warping transformations that remove even more anatomical variability ought to bring a further increase in sensitivity. There is some evidence showing this is the case [MKFK94, GAA97, SIO+98, HAC+02]. Many such algorithms are proposed, e.g. [AF99a, BK89, Boo89, CRM96, CE97, Dav96, Thi98, TT96, WGW+98]. These differ in the set of transformations searched, transformation parameterization, how the search is conducted, and the image feature used to drive the search. Such algorithms search for a spatial mapping T from the template image I_0 to subject image I_i by explicitly or implicitly minimizing some objective function of the form

$$\Phi(T) = \Phi_D(I_0, I_i \circ T) + \lambda\Phi_M(T), \tag{1.1}$$

where Φ_D represents the data (image similarity) term and Φ_M represents the model term that embodies the "prior knowledge" about the transformation expected.

The mathematical form for a data term has a theoretical basis in some instances [RMA00]. However, there is no biological theory to suggest a model term appropriate for transformation of one individual to another, so the models in use are generally either ad-hoc [CE97] or borrowed from physics (e.g. elastic solids [BK89], viscous fluids [CRM96], or diffusion [Thi98]). These models include parameters corresponding to physical quantities such as "stiffness" or viscosity whose value is not determined by theory. The coefficient λ in Equation 1.1, balancing the contribution of the data and model terms, is also undetermined by theory. An empirical performance measure is therefore required to evaluate design choices such as data and model terms, and to select parameter values. In the context of spatial normalization, residual anatomical variability is the natural choice for performance measure.

The first aim of this thesis is to develop measures that quantify the amount of variability in a population. These variability measures are applied in two ways. The first application is to evaluate design choices and select optimal numerical parameters of a particular algorithm. The second application is to compare spatial normalization methods in order to select the method that produces a normalized population with the smallest variability.

1.3 Cortical Surface Registration

While spatial normalization is typically carried out using 3D transformations to match the volumetric images, recently there has been a lot of interest in normalizing only part of the brain, namely the cerebral cortex [VEDJM98, FSD99, FSTD99, VD99, Dav97, TT96, TMM+97]. This is useful because the cortex is where the neuron bodies are, where the neural computation actually occurs, and thus is what neuroscientists are interested to map. As most of the signal in functional imaging comes from the cortex, it should be helpful to take the 2D nature of the cortex into account. For example, when the data analysis is performed in 3D functional signals from the opposite banks of a sulcus can be mixed together by the smoothing and averaging operations, which is undesirable as such locations may be quite distant as measured along the surface. Methods have been presented to analyze the fMRI signal on the cortex of an individual, e.g. [KGF00, AKM+01].

Anatomical measures like cortical thickness can also be mapped to the cortical surface [KLGME01]. Population comparisons of thickness are of interest to investigate thinning due to normal aging [LZG+03] or changes due to disease processes [ADPL+03, LBBE03].

The cerebral cortex is a large sheet of tissue, roughly 160 000 mm^2 in area but only 3mm thick [VEDJM98]. Typically, the functional or anatomical information of interest varies along the two large dimensions of the cortex, but not along the small ("thickness") dimension. Thus it is natural to model the cortex as an infinitesimally thin object, i.e. as a 2D manifold embedded in 3D.

The second aim of this thesis is to develop and validate an automated surface normalization algorithm that produces a surface-to-surface mapping, which is described as a "2D mapping" throughout this thesis. The validation addresses the extent to which anatomical variability in the cortex can be reduced using such a 2D mapping.

1.4 About the Thesis Organization

The first two parts of the thesis draw a deliberate parallel between 3D spatial normalization and 2D spatial normalization. The three chapters in each section address: (1) the appropriate mapping function and how to parameterize it, (2) obtaining the mapping function by optimization (i.e. image registration), and (3) measuring residual anatomical variability to gauge progress towards the goal of removing variability.

The third part of the thesis uses the variability measures to probe two more questions. First: what makes a good template image? Second: is the performance with respect to reducing variability of the cortex better if a 2D normalization is used rather than a 3D normalization?

A glossary containing both computer science and neuroimaging terms to aid a reader from either community is provided, as is an index of all terms defined in the text.

1.5 Thesis Contributions

The following are believed to be the significant new contributions made in this thesis.

As discussed in Section 1.2, one of the major unresolved issues with respect to nonaffine image registration algorithms is how to choose a suitable model term and how to choose the numerical parameters. Choosing a suitable template image for spatial normalization also remains a problem. These problems are addressed by items 1, 3, and 5.

To date, 2D normalization has been compared only to low-dimensional (affine) 3D normalization methods [DECS99, FSTD99]. In order to address the question in a general setting, representative high-dimensional 3D and 2D registration algorithms that demonstrably match the sulci well are required (items 2 and 4). The localization measure (item 3) is then used to compare the performance (item 6).

1. I adapt a measure of label consistency [WRH$^+$01] and develop a general method to obtain registration algorithm parameters, by evaluating overall alignment of segmented images. This method is equally applicable to 3D (Section 4.1) and

to 2D (Section 7.1) registration. A preliminary version of this work appears in a conference publication [RECW03].

2. I demonstrate the utility of the general evaluation technique by applying it to an existing 3D registration algorithm, ANIMAL [CHPE95] (Section 4.1.1). In doing so, I dramatically improve the ability of ANIMAL to match cortical structures such as sulci.

3. I propose a measure of dispersion, based on the mean of a random closed set, that is applicable to extended point sets such as sulci (Section 4.2). I use this dispersion to construct a novel measure of the degree to which a particular structure is localized (Section 4.3). The dispersion and localization measures also apply in 2D (Chapter 7).

4. I propose a general method to align cortical surfaces of spherical topology (Section 6.7), by matching the pattern of a scalar function defined on each surface. The operation of the algorithm is oblivious to the meaning of the scalar function. I obtain good results using the geodesic distance transform starting from a set of gyral seed points (Section 6.2.2).

5. I perform a quantitative comparison of the effectiveness of four different strategies for obtaining a template image (Chapter 8).

6. I demonstrate the first comparison of the effectiveness of performing spatial normalization in 2D versus normalization in 3D using high-dimensional warping (Chapter 9).

Part I

3D Spatial Normalization

Chapter 2

Spatial Transformation

As discussed in Chapter 1, the spatial normalization problem is to obtain a spatial transformation T that maps from standard space to an input image domain. Ideally, the mapping should be a *homeomorphism*, i.e., continuous, invertible, and with continuous inverse though, as discussed in Chapter 1, a small amount of noninjectivity is tolerated. The class of *all* homeomorphisms is too large to represent in a computer, let alone search through. Rather, the transformation T must be representable with a finite number of parameters. This chapter is concerned with the appropriate class of transformations in which to search for T.

The term *warp* is used as a synonym for a spatial mapping. It is also used as a verb to mean "to apply a spatial mapping". For example, given a spatial mapping $T : \mathbb{R}^3 \to \mathbb{R}^3$, a point set $A \subset \mathbb{R}^3$ is said to be warped to the point set $\{T(x) : x \in A\}$.

2.1 Image Domain

Mathematically, each input MR image I is idealized as a continuous function from some domain $\mathcal{D}_I \subset \mathbb{R}^3$ to $\mathbb{R}$. The domain is generally a rectangle, e.g., $\mathcal{D}_I = [0, 256]^3$.

2.2 Auxiliary Space

While the 2D registration problem (see Chapter 5) employs an auxiliary space in addition to the input surfaces, no auxiliary space is required for 3D registration. The spatial mapping, T, is defined on the same domain as image I, i.e. $\mathcal{D}_I$.

Generally speaking the mapping function is either defined on all of $\mathbb{R}^3$, or it can be easily extended to all of $\mathbb{R}^3$. So, unless specified otherwise, T is considered a mapping $\mathbb{R}^3 \to \mathbb{R}^3$.

2.3 Low-Dimensional Warping

Consider the class of spatial transformations with very few parameters, specifically the class of "affine" transformations.

Brains vary in size so *rigid-body transformations*, i.e., the set of translations and rotations, is certainly too limited a set to search. Such a transformation will not be able to match a large brain to a smaller one.

In order to adjust for size, the set of possible transformations could be enlarged by including a global scaling operation in addition to rigid-body transformations. Alternatively, three operators could be added: independent scaling along the x-, y-, or z-axis. Such a transformation thus consists of a rigid rotation, followed by 1 or 3 scaling operations, then a translation. To specify a transformation of this kind, three real parameters are needed for the translation, three parameters (e.g. a unit vector plus a rotation angle, or three Euler angles [GPS02]) for the rotation, plus one or three parameters for the scaling operation(s). These are referred to as the "seven-parameter" and "nine-parameter" set of transformations, respectively.

All transformations just described fall in the group of affine transformations. A general *affine transformation* has 12 parameters, which can be arranged into a 3x3 matrix A and a 3-vector b for the affine transformation that maps point x to point $Ax+b$. Matrix A represents the general linear transform part of the affine transformation, and must have nonzero determinant. The set of rigid-body transformations is the subgroup for which A is a rotation matrix (rather than, say, a scaling or a reflection). The *seven-parameter set* restricts A to the form sR for scalar $s > 0$ and rotation matrix R. The *nine-parameter set* restricts A to the form

$$A = \begin{pmatrix} s_1 & & \\ & s_2 & \\ & & s_3 \end{pmatrix} R,$$

where R is a rotation matrix and s_1, s_2, and s_3 are positive scalars representing the scale factors for each axis.

An affine transformation has the property that a straight line is mapped into a straight line. For this reason, these transformations are sometimes referred to as "linear".

The first applications of spatial normalization for brain mapping take their cue from the atlases of Talairach and co-workers [TST+67, TT88] and use affine or piecewise-affine transformations (with a small number of pieces) [FPR85, EMN+92].

See [Woo96] for a good review and history of Talairach and Talairach-inspired normalization procedures. Automated procedures for brain image registration using an affine mapping have been presented by several groups, including Woods et al. [WCM92, WMC93, WGH$^+$98, Woo99], Collins et al. [Col94, CNPE94], Downs et al. [LFD$^+$99, DILF99], Ashburner et al. [ANC$^+$97]. These algorithms differ in how the search is conducted, the topic of Chapter 3.

An affine transformation allows the transformed image to match the rough size and shape of the template. After allowing six parameters for location and pose and one for overall size, there are only five parameters left (two for differential scaling along the coordinate axes, and three for "skew") to specify the affine transformation. Five parameters (or two if using the 9-parameter subset) are certainly inadequate to completely capture the shape variations of the normal human brain [EKCM94].

Figure 2.1 shows a classification of two brains into tissue types (gray matter, white matter, and CSF voxels). The brains have been registered using a 9-parameter spatial normalization. The third column in the figure shows an image of the unmatched gray matter, i.e. voxels that are gray in one individual but not the other. If the 9-parameter normalization were sufficient there would be no mismatch, but this is manifestly not the case.

2.4 High-Dimensional Warping

In order to better match fine detail, a more flexible class of transformation functions must be used. It is standard to work with the *deformation field*, defined as

$$\Delta(x) = T(x) - x,$$

rather than working directly with T. One reason for this is that the boundary conditions for the deformation field are easily specified, making it more convenient to work with than T itself. For example, it is common to require that the displacement be zero outside some region containing the source image I: let $\mathcal{D}_S$ be a bounded, open set of $\mathbb{R}^3$ containing the domain of I and require

$$\Delta(x) = 0 \; \forall x \notin \mathcal{D}_S.$$

Define the coordinate functions of the deformation field as

$$\Delta(x) = (\Delta_1(x), \Delta_2(x), \Delta_3(x)).$$

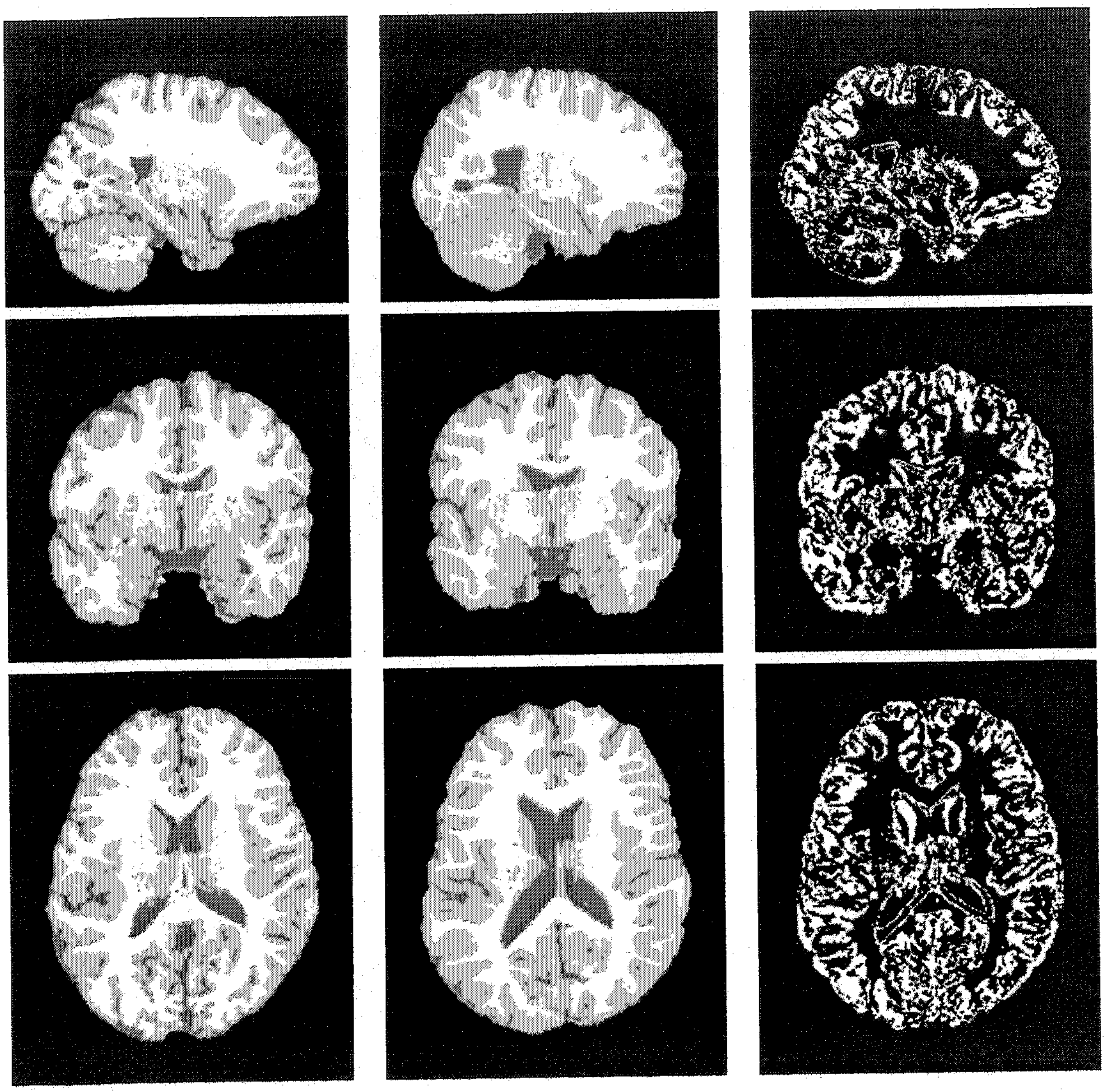


Figure 2.1: Map of gray matter mismatch after a 9-parameter spatial normalization. Left column shows a classification of one individual brain into white matter, gray matter, CSF (darker gray) and background (black). Another individual is shown in the middle column. The third column shows in white the voxels that are gray in one subject but not in the other.

There are a vast number of possible nonaffine transformations. A registration algorithm searches through a pre-specified set of transformations, each transformation being specified by a vector of parameters. A parameter is said to have a *global effect* if the parameter is used to compute the value of $\Delta(x)$ at all points $x \in \mathbb{R}^3$. Otherwise, the parameter has a *local effect*. While it is conceivable to have a transformation class for which some parameters have local effect and others a global effect, it is more common that all parameters have a global effect, or all have a local effect. Such transformations are denoted *global transformation* and *local transformation*, respectively [Bro92, MV98]. An affine transformation is an example of a global transformation. A transformation with mixed local effect and global effect parameters is most often written as the concatenation of two or more transformations, each of which is global or local.

2.4.1 Global Transformations

The deformation field is generally expressed as a sum of basis functions, each of which has global support (i.e. is non-zero on all of $\mathbb{R}^3$). The parameters are the coefficients that weight the basis functions of the summation. The deformation field Δ can be expanded using vector-valued basis functions. Alternatively, each component Δ_i can be written as an expansion of scalar-valued basis functions.

Vector Basis Functions

Let the basis functions take values in $\mathbb{R}^3$ and write $\Delta(x) = \sum_{k=i}^{p} c_k \phi^k(x)$, where $\phi^k : \mathbb{R}^3 \longrightarrow \mathbb{R}^3$ is the k^{th} basis function and $\{c_k\}$ is the set of parameters that defines Δ (and thus T). Basis functions of this form might arise as eigenfunctions of a differential operator [CRM94, CJM97], as discussed in Section 3.4.1.

Scalar Basis Functions

The three deformation field coordinate functions can be expanded independently using a common basis of scalar functions $\phi^k : \mathbb{R}^3 \rightarrow \mathbb{R}$, writing $\Delta_i(x) = \sum_k c_{ik} \phi^k(x)$. The coefficients $\{c_{ik}\}$ are the parameters required to specify the transformation. From this point of view, affine functions are those using x_1, x_2, x_3, and 1 as basis functions. Woods [WGW$^+$98, Woo99] extends the basis set to include polynomials of the coordinates up to fifth order. Friston et al. [FAF$^+$95, AF99a, AF99b] use sine and cosine basis functions.

Another set of approaches to registration borrows from scattered data interpolation [MR99]. This approach presupposes that the transformation is partially specified *a priori*, as discussed in Section 3.2.1. For example, T could be required to satisfy $T(p^k) = q^k$ with n pairs $\{(p^k, q^k) : k = 1 \dots n\}$ of pre-specified homologous "landmark" points. Points $\{p^k\}$ lie in the source image domain while the set $\{q^k\}$ lie on the target image domain. The deformation field Δ is obtained by solving 3 independent interpolation problems, one for each coordinate function Δ_i. The system of equations for coordinate i is $\{p_i^k + \Delta_i(p^k) = q_i^k : k = 1, \dots, n\}$.

A common parameterization of this sort employs a low order polynomial in x plus a function of $||x - p^k||$ for each landmark point,

$$\Delta_i(x) = f_i(x) + \sum_{k=1}^{N} c_{ik}\psi(||x - p^k||),$$

where $f_i : \mathbb{R}^3 \longrightarrow \mathbb{R}$ is a polynomial and $\psi : \mathbb{R} \longrightarrow \mathbb{R}$ is known as a *radial basis function*.

Properties of the transformation T depend on the radial basis function chosen. Of the many possible choices, perhaps the "thin-plate" splines promoted by Bookstein [Boo89, Boo91] are the most widely known in medical imaging. The thin-plate spline radial basis function (for d dimensions) is defined as

$$\psi_{\mathrm{TPS}}(r) = \begin{cases} r^2 \log r & d = 2 \\ r & d = 3 \end{cases} .$$

Other choices include the multiquadric [RM95]

$$\psi_{\mathrm{M}}(r) = (r^2 + a)^b, a > 0, b \neq 0,$$

or Gaussian [AR95]

$$\psi_{\mathrm{G}}(r) = e^{-r^2/2\sigma^2}.$$

A global transformation can be difficult to control, as a change in any parameter affects the value of the transformation globally. Iterative estimation of the parameters may thus be expensive, since all the data needs to be taken into account in order to estimate any one parameter. If the parameter only affects the transformation locally, on the other hand, it may be estimated using just local information.

2.4.2 Local Transformations

The global nature of radial basis functions can be avoided by using a function with *compact support*, i.e. $\psi(r)$ is nonzero only on an interval $[0, a]$, e.g. [FRS99]. This guarantees that a landmark point has local influence on T. However, the choice of support radius a is depends on the number and configuration of the landmark points, as well as on the displacement of the landmarks $||p^k - q^k||$. It is not clear how to use such a transformation in an automated registration system.

The alternative chosen here is to avoid parameters with global influence by defining T piecewise. Break domain of T into small simple subdomains, such as cubes or tetrahedra, with Δ defined on each subdomain using a low-degree polynomial. Each parameter affects Δ only in one or few subdomains. Examples of transformations defined in this manner include Goshtasby's piecewise mappings defined on triangulations (in the plane) [Gos86, Gos87], the analogous 3D mappings using tetrahedra [AAF00], and mappings defined using various splines, e.g. [SP86, DSTA95, LWCS96, SC97, Sub99].

Since digital images are usually defined on a regular rectangular grid it is convenient to use rectangular parallelepiped subdomains to parameterize Δ, with each face of a subdomain parallel to a coordinate plane.

Definition 2.4.1 *Mesh Warping. Let the domain* $[0, L_1] \times [0, L_2] \times [0, L_3]$ *be subdivided into* $N_1N_2N_3$ *subdomains, each of size* $s_1 \times s_2 \times s_3$*, where* $s_i = L_i/N_i$*. The set of corners of the parallelepipeds comprise the vertices of the* control mesh, *written as* $\{V^{ijk} = (is_1, js_2, ks_3) : i = 0, 1, \ldots, N_1; j = 0, 1, \ldots, N_2; k = 0, 1, \ldots, N_3\}$ *where* s_i *is the* vertex spacing *along the* i^{th} *axis.*

The mapping function $x \mapsto x + \Delta(x)$ *is determined by a continuous deformation field,* Δ*, that is interpolated from values* $\{\Delta^{ijk}\}$ *stored at the 8 control vertices belonging to a subdomain containing* x*.*

Mesh warping is also known as a *free-form deformation*, though this term is often restricted to the case that the interpolation method is a tensor-product cubic spline. Mesh warpings are used in computer graphics for image morphing (e.g. [LWCS96]) and also in automated brain mapping by many groups, e.g. Bajcsy and Kovačič [BK89, KB99], Collins et al. [CPDE92, CPE94, Col94, CHPE95, CE97, CE99], Davatzikos et al. [Dav96], Christensen et al. [CRM96, CJM97], and others [KPH+96, RSH+99, KLF99, RCRM01, SD02]. Indeed, mesh warping is probably the most common parameterization used, as it is a natural parameterization when Δ is obtained using finite difference or finite element methods (discussed further in Section 3.4.1). The interpolation method used is generally either linear or cubic.

The set of transformations representable (i.e. the flexibility of the transformation T) using a mesh warping is controlled in a natural fashion by the vertex spacing. Very often the vertex spacing is equal in the three dimensions ($s_1 = s_2 = s_3$) and a single value is quoted as the vertex spacing. A finer control mesh allows a more flexible transformation to be represented than a coarser mesh. A registration algorithm often estimates the final transformation in a series of steps, starting with a coarse control mesh and proceeding to finer and finer meshes as discussed in Section 3.5. The final vertex spacing is often on the order of the image voxel size. In other circumstances, the vertex spacing can be limited to a larger value. This is useful to match the gross shape of a brain to a template when the latter is missing fine details, but a better matching is desired than is possible using just an affine transformation. This strategy is used in Chapter 8 to obtain improved template brain images.

Mesh Warping with Linear Interpolation

Consider first the two-dimensional analog *bilinear interpolation* for which the domain is subdivided into rectangles. The interpolation works as follows. First, locate x in the control grid: let $i = \lfloor x_1/s_1 \rfloor$, $j = \lfloor x_2/s_2 \rfloor$. Point x is located in rectangular patch with control vertices $V^{i,j}$, $V^{i+1,j}$, $V^{i+1,j+1}$, and $V^{i,j+1}$, as shown in Figure 2.2. Let $\alpha = x_1/s_1 - i$ and define two points on the patch boundary in a vertical line with x,

$$M^j \equiv (1-\alpha)V^{i,j} + \alpha V^{i+1,j}, \tag{2.1}$$

$$M^{j+1} \equiv (1-\alpha)V^{i,j+1} + \alpha V^{i+1,j+1}. \tag{2.2}$$

Point x lies on line segment $M^j M^{j+1}$. Define $\beta = x_2/s_2 - j$ so that

$$\begin{aligned} x &= (1-\beta)M^j + \beta M^{j+1} \\ &= (1-\alpha)(1-\beta)V^{i,j} + \alpha(1-\beta)V^{i+1,j} + (1-\alpha)\beta V^{i,j+1} + \alpha\beta V^{i+1,j+1}. \end{aligned} \tag{2.3}$$

Defining $B_0(s) = 1 - s$ and $B_1(s) = s$, Equation 2.3 becomes

$$x = \sum_{l=0}^{1} \sum_{m=0}^{1} B_l(\alpha) B_m(\beta) V^{i+l,j+m}. \tag{2.4}$$

The coefficients for each control vertex are used for interpolation of Δ

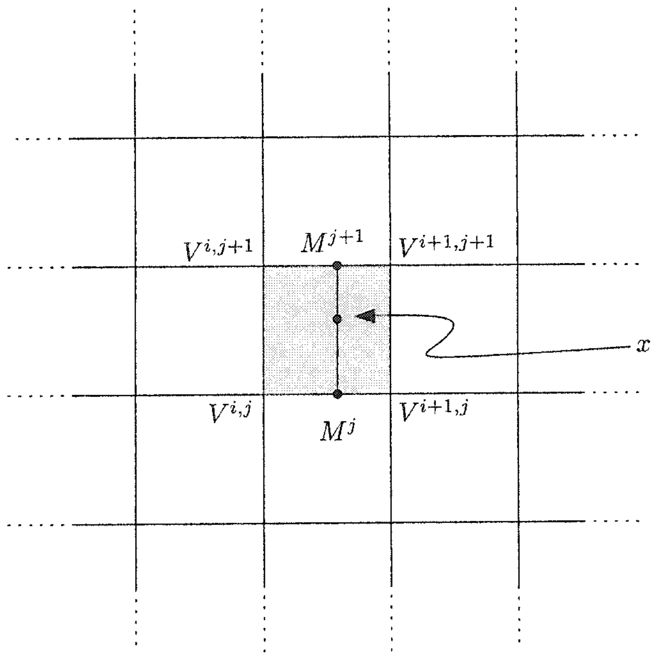


Figure 2.2: Bilinear interpolation for mesh warping. Points M^j and M^{j+1} are obtained by linear interpolation of adjacent control vertices. Point x is linearly interpolated on segment M^jM^{j+1}.

$$\Delta(x) = \sum_{l=0}^{1}\sum_{m=0}^{1} B_l(\alpha)B_m(\beta)\Delta^{i+l,j+m}. \tag{2.5}$$

This is easily extended to 3D,

$$x = \sum_{l=0}^{1}\sum_{m=0}^{1}\sum_{n=0}^{1} B_l(\alpha)B_m(\beta)B_n(\gamma)V^{i+l,j+m,k+n}, \tag{2.6}$$

with $k = \lfloor x_3/s_3 \rfloor$ and $\gamma = x_3/s_3 - k$, with interpolation done using

$$\Delta(x) = \sum_{l=0}^{1}\sum_{m=0}^{1}\sum_{n=0}^{1} B_l(\alpha)B_m(\beta)B_n(\gamma)\Delta^{i+l,j+m,k+n}. \tag{2.7}$$

For future reference note that *any* function can be similarly defined given values on a rectangular mesh, such as the image intensity function given values at voxel centres.

Mesh Warping with Cubic Interpolation

Cubic interpolation proceeds in much the same manner as for linear interpolation, except that the polynomials B_i are now third order. Also, there are four B_i, so the

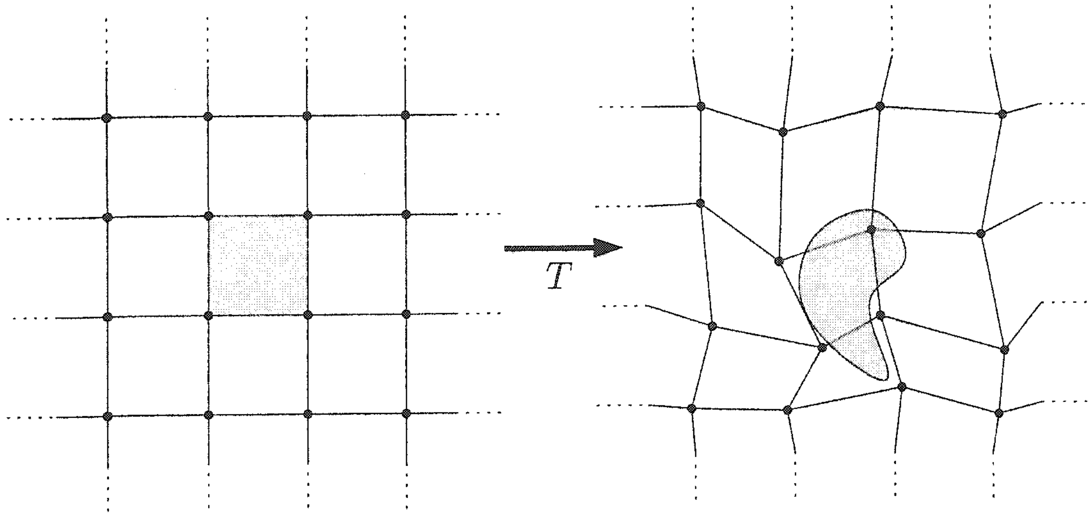


Figure 2.3: Mesh warping in $\mathbb{R}^2$ with tensor-product cubic splines. The transformation for each patch is determined by the location of 4^2 control-points.

interpolation requires (in d dimensions) a 4^d grid of control vertices instead of a 2^d grid. Equation 2.6 becomes

$$x = \sum_{l=0}^{3} \sum_{m=0}^{3} \sum_{n=0}^{3} B_l(\alpha) B_m(\beta) B_n(\gamma) V^{i+l-1,j+m-1,k+n-1}. \tag{2.8}$$

The patch to be transformed lies at the centre of the 4^d set of control vertices. Figure 2.3 illustrates the situation for warping in $\mathbb{R}^2$.

The form of the cubic function B_i is determined by the type of spline desired. Sederberg and Parry [SP86] use Bernstein polynomials in order to warp geometric objects. Others use B-spline patches, e.g. [LWCS96] for image warping, Catmull-Rom splines, or other spline functions [SC97].

Both linear interpolation and Catmull-Rom splines have the useful property of interpolating the function value at control mesh vertices, i.e. $\Delta(V^{ijk}) = \Delta^{ijk}$ for all V^{ijk}. The set of transformation values on the control mesh are thus often interpreted as a "sampling" of the deformation field Δ. This property is convenient in understanding the registration process as matching the location of point V^{ijk} in the source image to point $V^{ijk} + \Delta^{ijk}$ in the target image. Both linear interpolation and the interpolating Catmull-Rom splines are a reasonable choice for the mesh warping interpolation function. For the 3D experiments presented in this thesis, an existing algorithm is used [CHPE95] which uses the cubic Catmull-Rom interpolating spline.

2.5 Injectivity

An affine transformation is invertible as long as the linear part is non-singular, i.e. $|A| \neq 0$. In contrast, injectivity of a high-dimensional warping is difficult to ensure. For example, there is nothing in the definition of a mesh warping that guarantees it to be injective.

Researchers often prove that the spatial transformation function T is injective under the assumption that T is differentiable, generally relying on the classical inverse function theorem. The inverse function theorem states that an inverse exists in a neighbourhood of point $T(x)$ if the determinant of the Jacobian is nonzero at x, i.e. if $|\nabla T(x)| \neq 0$. Such proofs are derived in a continuous setting, implicitly assuming a computer model that can solve integral equations over real numbers. Though no proof is given that the mapping remains injective after the problem is discretized and solved numerically, it is generally believed that the transformations obtained by an implementation of such an algorithm will be sufficiently close to the theoretical result if sufficiently many basis functions or a sufficiently small control mesh vertex spacing is used.

Dupuis, Joshi, et al. [CJM97, DGM97, Jos98] prove that the solution to their image registration problem is a differentiable homeomorphism, i.e. it is a *diffeomorphism* and thus is injective. Another tactic, adopted by Lee et al. [LCHS96] and Ashburner et al. [AAF99, AAF00], among others, is to add a term to the objective function that penalizes a zero Jacobian determinant. This discourages the optimization from converging to a solution that has a zero Jacobian determinant value. It may also cause the optimization to become stiff and thus not as effective at locating a good solution for T.

Lee et al. [LWCS96] use a 2D cubic B-spline mapping function with uniform control mesh vertex spacing (i.e. $s_1 = s_2$) to morph digital images. They investigate conditions on the control mesh vertex displacements to ensure that the warp is one-to-one, obtaining a sufficient condition: that the maximum control point displacement should not exceed $0.48 s_1$, i.e. $\max_{ijk} |\Delta_i(V^{jk})| < 0.48 s_1$. In the same paper, they provide an example in which control point displacements of $0.49 s_1$ result in a noninjective map. Choi and Lee [CL00] refine the analysis somewhat and extend it to 3D. However, the resulting bounds on the maximum control point displacement are fairly small and are only sufficient conditions, so it is not clear how useful these are in practice. For example, setting $\Delta(x) = (1, 1)$ for all x is a valid injective translation, but it violates the conditions for the theorem of Lee et al.

2.6 Summary

A necessary condition for a successful spatial normalization of the cortex is that the warping of one brain to another matches gray matter to gray matter. This is not possible to achieve using an affine transformation [EKCM94], as illustrated by Figure 2.1, so a high-dimensional warping is sought.

For the work in this thesis, the transformation is parameterized as a mesh warping (Definition 2.4.1) using a cubic Catmull-Rom spline for interpolation. This parameterization has two advantages in addition to local control. First, by restricting the control mesh vertex spacing, there is a natural control over the amount of flexibility allowed by the transformation. Second, the transformation interpolates its values at control mesh vertices which is convenient for setting up the registration process (discussed in the next chapter) as the parameters stored at each control mesh vertex can be interpreted as the value of the deformation field.

As discussed in Section 2.5, there are no useful necessary conditions known for the deformation field values that ensure the mesh warping is injective. If required, injectivity is therefore enforced through the optimization algorithm, discussed in the next chapter.

Chapter 3

Searching for an Optimal Transformation

The goal of this chapter is to survey the components and principles of an algorithm for solving the following problem. Given a template image I and an image J, compute a transformation function $T : \mathcal{D}_I \subset \mathbb{R}^3 \rightarrow \mathbb{R}^3$ that associates each point x in image I with a point $T(x)$ in image J. This association should have some meaning: the points x and $T(x)$ should correspond biologically, i.e., be homologous. This problem, known as the *image registration* problem, has a long history in remote sensing and computer vision. Brown [Bro92] surveys image registration problem generally. Surveys by Maintz and Viergever [MV98] and by Lester and Arridge [LA99] cover registration in the medical imaging context, as do the books [Tog99, HHH01].

The registration problem is often cast as an optimization. The precise optimization problem will come into focus as the chapter progresses. Roughly speaking, T is sought that optimizes an overall measure of how well T matches I to J. Registration algorithms differ in the details of some key components. Viewing registration as an optimization problem provides a useful framework in which to compare different algorithms appearing in the literature.

The first component to specify is the "feature" data that is used to find the match. In this context, *feature* refers to any piece of data obtained from the image, whether identified by a human or extracted by the computer. The feature could be the image value itself, a geometric point, curve, or surface patch, or the result of sophisticated pre-processing such as classification of each voxel (as to its tissue type, for example).

Closely tied to the feature used in the match is the *objective function*, that is, a function of the transformation T that measures how well T matches the features of I with the features of J. The complete objective function, generically denoted Φ

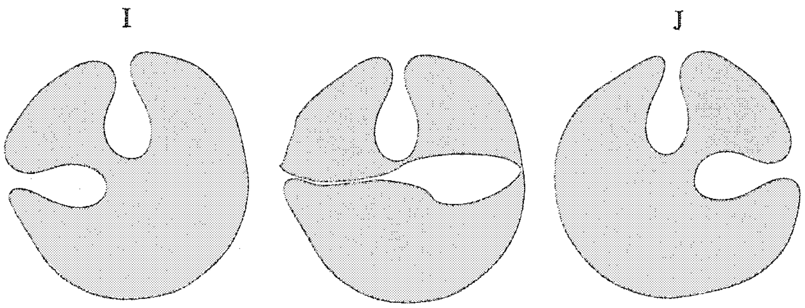


Figure 3.1: Example of implausible registration. Figure on left is matched to figure on right by distorting the cavity on its side. A rotation by 90° is a more plausible match.

throughout this thesis, often has two or more terms. The term that measures the quality of the data match is called the *data term* and is denoted $\Phi_D(I, J, T)$.

The third component of matching is the set of functions in which to search for T. Chapter 1 argues that the transformation should in principle be a homeomorphism, or nearly so (some noninjectivity is tolerated). However, depending on the parameterization of T (Chapter 2), it is not necessarily possible to limit the search to only homeomorphisms. Moreover, the space of homeomorphisms is too large: it contains biologically-implausible matches. Figure 3.1 shows a schematic example of an implausible transformation. The leftmost figure (image I) is to be matched with the rightmost (image J). In the middle is an image of I after applying a spatial transformation that causes the left cavity to deform, fitting to the right cavity on image J. A rotation by 90° would have fit as well or better, and would be much more plausible. Methods to deal with this problem are discussed in Section 3.3.1. The result of such considerations is that a second kind of term, known as a "regularization" or "model" term, is incorporated into the objective function. Once the data and regularization terms are chosen, the objective function Φ is generally expressed as a sum

$$\Phi(I, J, T) = \Phi_D(I, J, T) + \lambda \Phi_R(T), \tag{3.1}$$

and the optimization problem to solve becomes

$$T = \arg\min_{T'} \Phi(I, J, T').$$

There are several possible approaches to search for the minimizer of this expression. These algorithms are the subject of Section 3.4.

Spatial normalization methods generally estimate first an affine transformation (or, say, a nine-parameter transformation described in Section 2.3), T_A, followed by a nonaffine warping, T_W. The complete mapping is a concatenation of the two. This chapter is only concerned with the nonaffine warping transformation (hereafter referred to as simply T).

Except when specified otherwise, the images and the transformation are understood to be continuous functions defined on a continuous domain. The input image and the template image are both given as samples on a 3D rectangular grid. Assuming it is band limited and adequately sampled, the underlying intensity function could be recovered using sinc interpolation although in practice a simpler (hence faster) interpolation method such as trilinear interpolation (Equation 2.7) is more often used. See [LGS99] for a survey of interpolation methods. After choosing an interpolation method, the images I and J can each be treated as a real-valued function: $I : \mathcal{D}_I \subset \mathbb{R}^3 \to \mathbb{R}$ and $J : \mathcal{D}_J \subset \mathbb{R}^3 \to \mathbb{R}$. Though the transformation T is only required for points on domain $\mathcal{D}_I$, it is convenient to describe the theory assuming that transformation T is a continuous map $\mathbb{R}^3 \to \mathbb{R}^3$.

3.1 Coordinate Invariance

Before enumerating specific data and regularization terms for the objective function, it is worth considering a broad principle that constrains the choice of objective function terms. The principle is simply that the spatial mapping obtained by registering image I to image J should not depend on the specific Cartesian coordinate frame chosen in which to work. This principle is generally satisfied by using only terms that are both rotation-invariant and translation-invariant in the objective function.

Let P be a point in $\mathbb{R}^3$ with coordinates x_P in a particular right-handed Cartesian coordinate frame, denoted $\mathbb{F}$. Suppose there is a second right-handed Cartesian coordinate system, $\overline{\mathbb{F}}$, in which the coordinates are given as

$$\overline{x} = Rx + s \tag{3.2}$$

for a point at coordinates x in frame $\mathbb{F}$, where R is a rotation matrix and s is a vector. Point P has coordinates $\overline{x_P} = Rx_P + s$ in frame $\overline{\mathbb{F}}$.

A quantity that does not depend on the coordinate frame is called a *Cartesian scalar*. The image intensity I at a particular point in space, say P, is independent of the coordinate system, thus $I(x_P)$ is a Cartesian scalar. The *mathematical form* of the intensity function I does, of course, depend on the coordinate system but the

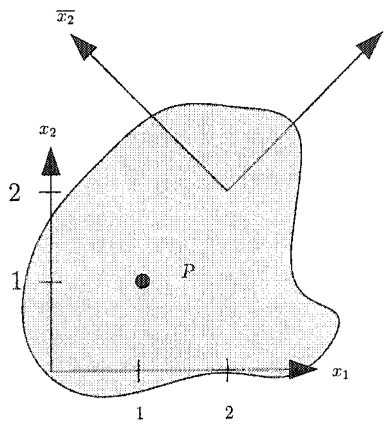


Figure 3.2: Coordinate change. Point P has coordinates $x_P = (1,1)$ in the x_1x_2 system, and coordinates $\overline{x_P} = (-\sqrt{2}, 0)$ in the $\overline{x_1x_2}$ system. Image intensity at point P does not depend on the coordinate system, so the intensity function in each system must satisfy $I(x) = \overline{I}(\overline{x})$ where $\overline{x} = Rx + s$, $R = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}$, and $s = (-2\sqrt{2}, 0)$.

mathematical form in the two systems must be related by $\overline{I}(\overline{x}) = I(x)$ for all points, where the coordinate values are related by Equation 3.2. These definitions work for any dimension; Figure 3.2 illustrates the concepts in $\mathbb{R}^2$. In contrast to a scalar, the component values of a vector *do* depend on the coordinate frame. A vector with components $v = (v_1, v_2, v_3)$ in frame $\mathbb{F}$ has components

$$\overline{v}_j = \sum_{h=1}^{3} R_{jh} v_h$$

in frame $\overline{\mathbb{F}}$. This sort of transformation property can be generalized to quantities with more than one subscript as follows.

Definition 3.1.1 *3D Cartesian Tensor [LR75]. Let r be any positive integer. A set of 3^r quantities $T_{h_1h_2\ldots h_r}$, $h_i = 1,2,3$ for $i = 1,\ldots,r$, is said to constitute the components of a Cartesian tensor of rank r if, under the coordinate transformation 3.2, these quantities transform according to the transformation law*

$$\overline{T}_{j_1j_2\ldots j_r} = \sum_{h_1=1}^{3} \cdots \sum_{h_r=1}^{3} R_{j_1h_1} \cdots R_{j_rh_r} T_{h_1\ldots h_r}. \tag{3.3}$$

Note that vectors are Cartesian tensors of rank 1 and scalars are regarded as tensors of rank 0. Tensors can be defined in space with arbitrary dimension and with respect to a general nonlinear coordinate transformation (see Lovelock and Rund [LR75]) but such generality is not needed here. In this chapter the term "tensor" is used without qualification to refer to 3D Cartesian tensors. The modifier "Cartesian" (some authors use the term *affine tensor* instead of Cartesian tensor) is applied because the coordinate transformation (Equation 3.2) is restricted to be an affine relationship between Cartesian frames in a Euclidean space.

An *invariant tensor* is a tensor that has the same numerical value in *every* Cartesian coordinate system. One example is the zero tensor. Another is the Kronecker tensor

$$\delta_{jk} \equiv \begin{cases} 1 & \text{if } j = k, \\ 0 & \text{otherwise} \end{cases} . \tag{3.4}$$

The Kronecker tensor is shown to be invariant by computing the component values in the frame $\overline{\mathbb{F}}$ using Equation 3.3,

$$\overline{\delta}_{jk} = \sum_{l}\sum_{m} R_{jl}R_{km}\delta_{lm} = \sum_{l} R_{jl}R_{kl} = (RR^T)_{jk} = \delta_{jk},$$

where the last equality holds because R is an orthogonal matrix.

New tensors may be formed from existing tensors in a number of ways. Addition of tensors of the same rank forms a tensor; for example if A_{ijk} and B_{ijk} are the components of two rank 3 tensors, then so is $C_{ijk} \equiv A_{ijk} + B_{ijk}$. A rank r and a rank s tensor may be multiplied to form a rank $r+s$ tensor; e.g. $C_{ijklmn} \equiv A_{ijk}B_{lmn}$ form the components of a rank 6 tensor. A rank $r \geq 2$ tensor may be summed over a pair of indices to produce a rank $r-2$ tensor, a process known as *contraction*. For example, $C_j \equiv \sum_{i=1}^{3} A_{iij}$ comprise the components of a rank 1 tensor. Finally, differentiation of a rank r tensor function forms a rank $r+1$ tensor function. For example if $A_{ij}(x)$ are the components of a rank 2 tensor function, then $B_{ijk} \equiv \partial A_{ij}/\partial x_k$ forms a rank 3 tensor function.

The requirement that the transformation obtained by registration not depend on the coordinate system is satisfied by requiring that the objective function $\Phi(I, J, T)$ is a scalar. All values of Φ are then invariant under coordinate transformations. In particular, the local minima are unchanged by a coordinate transformation so the minimizing transformation T does not depend on the coordinate frame chosen.

Often the objective is an integral over some domain $\Omega \subset \mathbb{R}^3$, written $\Phi(I, J, T) = \int_{\Omega} \phi(I, J, T)dx$. In order that this integral be a scalar, it is necessary and sufficient [LR75] that the integrand be a scalar, i.e. $\overline{\phi}(\overline{x}) = \phi(x)$ where the coordinate change is given by Equation 3.2.

3.2 Data Terms

There are two broad classes of feature used in image registration. One class of features is geometric objects such as points, curves, surfaces, or volumes, and the other class is the set of image intensities, or a transformation thereof (such as smoothing or differentiation) [HHH01].

One registration strategy, used with geometric features, requires an *a priori* matching of corresponding feature pairs, $\{(f_I^k, f_J^k) : k = 1, \ldots, n\}$, where feature f_I^k is the k^{th} feature of image I and is matched to the k^{th} feature of image J, denoted f_J^k. Such a matching is used to seek a transformation T that maps the point(s) of feature f_I^k on to, or near to, the point(s) of feature f_J^k. Since generating the set of corresponding feature pairs is equivalent to assigning the label k to feature f_J^k, this kind of feature data is known as *labelled feature* data. The main dichotomy in data terms for medical image registration is between data terms that use labelled features and data terms that use unlabelled features (whether geometric features or intensity-based features) [AF99b].

3.2.1 Labelled Features

As discussed above, a labelled feature is a geometric object, i.e. a point set in $\mathbb{R}^3$. The most commonly used type of geometric feature is a discrete point, which is termed a *landmark point*. An extended feature comprised of a continuum of points that make up a one-dimensional curve [Sub99, DSTA95], a two-dimensional surface patch [HB00, TT96], or a three-dimensional volume set [CLGE98] can also be used.

There are two approaches to dealing with labelled data: the transformed feature, $T(f_I^k)$, could be required to map exactly onto the corresponding feature f_J^k, or it could be allowed to map only approximately to the corresponding feature. The resulting transformations are described as interpolating, or approximating, respectively.

Interpolating Transformation

Suppose it is desired that the transformation T maps each feature f_I^k of I exactly to its match f_J^k in image J. The registration then becomes a constrained optimization problem. For example, suppose there is a set of n point landmarks $\{p^k \in \mathcal{D}_I : k = 1, \ldots, n\}$ and a corresponding set of landmarks $\{q^k\}$ in the domain of J. An *interpolating transformation* has the property that $T(p^k) = q^k$ for all k. There are many interpolation methods available [MR99] including distance-weighted and radial basis function interpolation (discussed in Section 2.4.1). In the latter category are found the thin-plate spline methods of Bookstein [Boo89], and the landmark matching of Joshi et al. [Jos98, JM00].

An interpolating transformation can also be used for extended features, i.e. curves, surface patches, and volumes. For example, suppose that, in addition to the pairing of a curve in $c \in \mathcal{D}_I$ with a curve in $c' \in \mathcal{D}_J$, a mapping from c to c' is given, a transformation T can be found that extends this mapping (or, generally, a set of mappings between matched curve pairs) to the entire image domain [MR99]. In practice, however, curves and surfaces are often discretized into equally-spaced sample points and landmark matching is employed [VEDJM98] or an approximating transformation, described next, is used [CLGE98].

Approximating Transformation

In typical medical images, the landmark points are not easy to locate precisely. Grachev et al. [GBR+99] had an expert select 128 landmarks per hemisphere two times in the same brain, separated by a one month interval. They report an RMS error in locating the landmarks of 1.6 mm for each hemisphere. The error is generally larger if two different experts are asked to select the landmarks, as they might use different criteria for locating them.

Given that there is some uncertainty in locating the landmark, the registration problem can be relaxed so that the transformation is not required to interpolate the landmarks. Rather, the location $T(p^k)$ should be near q^k, though not necessarily exactly equal to it. The idea is to trade off data fitting for a smoother transformation, much like the problem of fitting a curve to noisy data. The data term for the registration problem can take the form of a weighted square distance penalty, e.g.

$$\Phi_{LM} = \sum_{k=1}^{n} \alpha_k ||q^k - T(p^k)||^2, \tag{3.5}$$

where p^k is a landmark point in image I that should match to the landmark point q^k of template image J. The α_k are weighting factors that could vary according to uncertainty in the coordinate values measured for p^k and q^k [JM00, CMP+01]. If the uncertainties are different for the different coordinate axes or if there are correlations between uncertainty values of the axes, a Mahalanobis distance can be used instead

$$\Phi_{LM} = \sum_{k=1}^{n} (q^k - T(p^k))^T \Sigma_k^{-1} (q^k - T(p^k)), \tag{3.6}$$

where Σ_k is the covariance matrix of uncertainties associated with the landmark pair (p^k, q^k) [JM00].

3.2.2 Unlabelled Features

Geometric features can also be used without labelling. Alternatively, the image intensity itself, or a simple transformation of it such as convolution with a blurring or differentiation operator, can be used in the data term.

Geometric Feature

The data term for geometric objects is generally based on distance from an object to its homologue in the other image, e.g. Φ_{LM} (Equation 3.6). While the features are initially unlabelled, a matching can be carried out during the process of registration. The iterative closest point (ICP) method and its successors are often used to estimate simultaneously T and a landmark point matching. The original ICP [BM92] is used for rigid-body transformations. Let $P_I = \{p^k \in \mathcal{D}_I\}$ be a set of landmark points extracted from image I, and let Q_J be another set of landmarks extracted from image J. The iterative closest points algorithm minimizes the distance from $T(p^k)$ to the *nearest* point in Q_J for each p^k, i.e. to minimize

$$\Phi_{ICP}(P_I, Q_J, T) = \sum_k \min_{q \in Q_J} ||T(p^k) - q||^2. \tag{3.7}$$

This does not require a 1-1 matching of features in I and J: several points in P_I may match to the same point in Q_J and some points of Q_J may not be matched to a point in P_I. The idea can be extended to nonrigid T [FA96], in which case extra constraints (i.e. regularization terms) are needed to prevent T from mapping all of P_I to a single $q \in Q_J$.

Image Intensity

For convenience define $\widetilde{J} = J \circ T$, obtaining a function that is defined on the same domain as I. In this section I and $\widetilde{J}$ are considered as discrete images, each given by a set of intensity values located on a common grid in $\mathcal{D}_I$.

Consider a data term that uses the image intensity alone. The criterion for deciding whether location $x \in \mathcal{D}_I$ is a good match to location $T(x) \in \mathcal{D}_J$ is made by comparing intensities $I(x)$ and $\widetilde{J}(x)$. This thesis is concerned with matching MR images of the same modality, generally T_1-weighted. In this situation it might be supposed that the intensity values should be approximately the same, leading to the data term that simply sums the squared intensity difference, $\sum(I(x) - \widetilde{J}(x))^2$, a function widely used for brain image registration, e.g. [CRM94, CPA99, AF99a, Woo99].

Consider the domain over which to perform this sum. Assume that the images cover the complete brain, avoiding issues that arise when one image covers a subset of the other, e.g. [SHH99]. An image that covers the entire brain often includes other parts of the head, e.g. scalp and neck. Non-head parts such as the neck can be removed by simple cropping of the images. The scalp can be left in the images or it, too, can be removed, e.g. [DFS99, Smi02, RMPO+02]. The summation should include all the voxels of I and of $\widetilde{J}$, a region that could be defined as $\Omega = \mathcal{D}_I \cup T^{-1}(\mathcal{D}_J)$. In this form, Ω inconveniently depends on T, the transformation sought. In practice, the domain of integration is taken to be $\mathcal{D}_I$ or a slight enlargement (by a few voxels, say) of $\mathcal{D}_I$, and the support for $\widetilde{J}$ is assumed to fall inside Ω. This last assumption can be checked by the registration algorithm as the match progresses.

The sum-of-squared-difference objective function is defined as

$$\Phi_{SSD}(I, J, T) = \sum_{x \in \Omega} (I(x) - \widetilde{J}(x))^2. \tag{3.8}$$

To account for the presence of noise in the images, suppose that

$$\widetilde{J}(x) = I(x) + \epsilon(x), \tag{3.9}$$

where $\epsilon(x)$ is a zero-mean random variable. In other words, I is the "true" image and $\widetilde{J}$ is a version corrupted by noise.

If it is further assumed that $\epsilon(x)$ is Gaussian with constant variance σ^2, then the probability for observing intensity $\widetilde{J}(x)$ when the true value is $I(x)$

$$\mathsf{P}(\tilde{J}(x)|I(x)) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(I(x)-\widetilde{J}(x))^2}{2\sigma^2}}.$$

To simplify matters, it is common to assume that measurements of each voxel of $\widetilde{J}$ are conditionally independent given I [RMA00]. The probability of observing the entire image is then just the product of the probability of each voxel observation. The probability of observing image $\widetilde{J}$ given that I is the "true" image is

$$\begin{aligned} \mathsf{P}(\tilde{J}|I) &= \prod\nolimits_{x\in\Omega} \mathsf{P}(\tilde{J}(x)|I) \\ &= (2\pi\sigma^2)^{-n/2} \exp(-\tfrac{1}{2\sigma^2} \sum\nolimits_{x\in\Omega} (I(x) - \widetilde{J}(x))^2), \end{aligned} \tag{3.10}$$

where $n = |\Omega|$ is the number of voxels in Ω. A common criterion for choosing T is to choose the transformation for which this probability is a maximum, the "method of maximum likelihood" [DHS01]. From Equation 3.10, it is clear that maximizing $\mathsf{P}(\tilde{J}|I)$ is equivalent to minimizing the expression in the exponent, which is equivalent to minimizing Φ_{SSD}.

However, it is well-known that Equation 3.9 is not a reasonable approximation in general. Even when the scanner settings are the same the signal elicited from, say, the white matter, can be very different between individuals. Figure 3.3 shows the intensity histogram of two individuals from the International Consortium for Brain Mapping (ICBM) dataset [MTE+95]. White matter voxels are among the brightest in a T_1-weighted scan and so are part of the bump furthest to the right. The top histogram in the figure indicates that white matter in subject 106 has a raw intensity value of about 500 000, while in subject 107 the white matter intensities are about 280 000. Even after the scans are preprocessed to remove the "shading" artifact (using the method of [SZE98]) caused by nonuniform sensitivity of the scanner receive coils, the intensity values (bottom histogram) for a given tissue differ between subjects.

Given that the "nearly equal" intensity assumption is not satisfied, Φ_{SSD} cannot be used alone. The images need to be further processed so that the intensity of a given tissue type is the same in each image. Consider an affine intensity relationship; that is, if the intensity for a given tissue in image I is i, then the same tissue should be measured to have intensity $\alpha + \beta i$ in image J. The two parameters α and β allow two intensity values in I and J to be matched, e.g. the peak values of gray and of white tissue. The values of α and β could be estimated by finding the intensity values for the two peaks in the histogram and solving a 2×2 system of linear equations. Then the intensity at each voxel x of image I can be remapped to $\alpha + \beta I(x)$ before registration.

An alternative to precomputing the intensity relationship is to estimate it as part of the optimization. Woods [Woo99] assumes a global linear relation between intensity values,

$$\Phi(I, J, T) = \sum_{x \in \Omega} (\alpha I(x) - \widetilde{J}(x))^2, \tag{3.11}$$

and includes α in the set of parameters to estimate during the optimization. Friston et al. [FAF+95] estimate an even more general intensity relationship.

It is also possible to build a certain intensity relationship into the data term *without* needing to estimate parameters of the intensity relationship. Incorporating

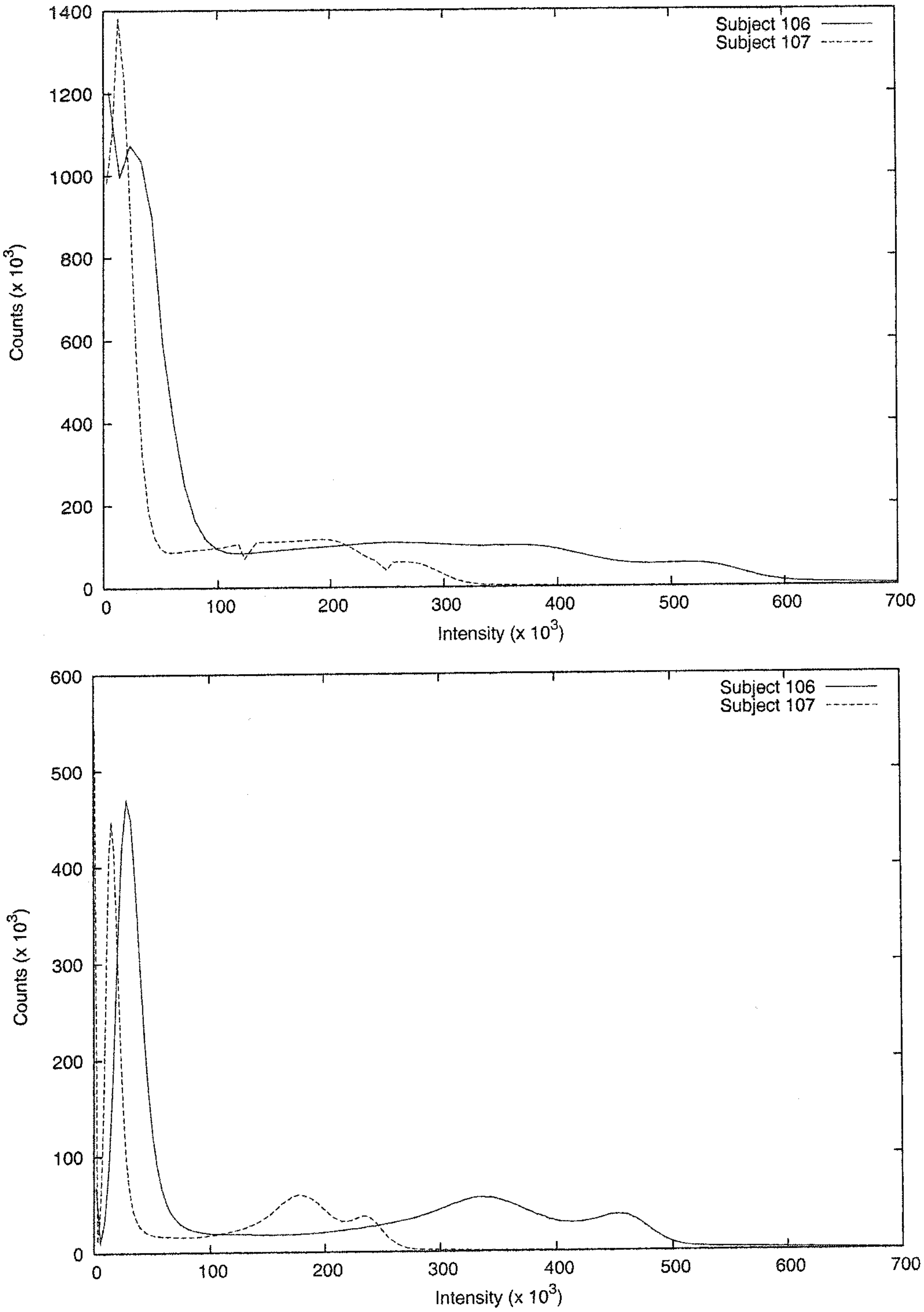


Figure 3.3: Intensity histograms of T_1-weighted scans of two subjects. The top plot shows the raw intensities. The bottom plot shows the intensities after preprocessing to correct for the intensity nonuniformity artifact. Notice that the intensity of gray and of white matter, located at the rightmost two peaks of each histogram, differs between the two subjects even after nonuniformity correction has been applied.

noise, the assumption of a global affine relation between intensities can be written

$$\tilde{J}(x) = \alpha + \beta I(x) + \epsilon(x). \tag{3.12}$$

Using a more sophisticated likelihood argument [RMA00], the affine intensity relation assumption leads to maximizing the *correlation coefficient* similarity measure

$$\Phi_{CC}(I, J, T) = \frac{Cov(I, \tilde{J})}{\sqrt{Var(I)Var(\tilde{J})}}, \tag{3.13}$$

in which the variance is defined as

$$Var(I) = \frac{1}{N}\sum_{x\in\Omega}(I(x) - \mu_I)^2,$$

and

$$\mu_I = \frac{1}{N}\sum_{x\in\Omega} I(x)$$

is the mean. The variance and mean for $\tilde{J}$ are defined analogously. The covariance is

$$Cov(I, \tilde{J}) = \frac{1}{N}\sum_{x\in\Omega}(I(x) - \mu_I)(\tilde{J}(x) - \mu_{\tilde{J}}).$$

The correlation coefficient produces a value in the range $[-1, 1]$, with higher scores indicating a better match.

Sometimes the assumption is made that intensities are linearly related, i.e.

$$\tilde{J}(x) = \beta I(x) + \epsilon(x),$$

rather than Equation 3.12. In the maximum likelihood framework [RMA00], this is equivalent to maximizing the *normalized cross correlation*

$$\Phi_{XC}(I, J, T) = \frac{\sum I(x)\tilde{J}(x)}{\sqrt{\sum I^2(x) \sum \tilde{J}^2(x)}}, \tag{3.14}$$

with all sums taken over the domain Ω.

Regional Similarity Measure

The intensity-based similarity measures defined so far, Φ_{SSD}, Φ_{CC}, and Φ_{XC}, are each evaluated globally, i.e. summed over all Ω. This is appropriate if the corresponding intensity relationship (equality, affine, and linear, respectively) holds over the entire image. However, the MR signal intensity from homogeneous tissue, is seldom uniform throughout an image. Instead, the intensity varies smoothly across the image, an artifact termed *intensity nonuniformity*. Methods exist (e.g. [SZE98]) to estimate and correct for the nonuniformity.

The nonuniformity can also be accommodated directly in the similarity measure, by allowing the coefficients α and β in Equation 3.12 to be spatially varying,

$$\widetilde{J}(x) = \alpha(x) + \beta(x)I(x) + \epsilon(x). \tag{3.15}$$

This complicates evaluation of a maximum likelihood data term. However, since the intensity variation is known to be smooth with slow variation in space [SZE98], the functions $\alpha(x)$ and $\beta(x)$ may be approximated as piecewise-constant functions. Break the domain Ω into n smaller regions,

$$\Omega = \cup_{k=1}^{n} \Omega_k,$$

where α and β are assumed constant on each Ω_k. The maximum likelihood solution on region Ω_k is obtained by maximizing Φ_{CC} over points in Ω_k. Maximizing independently over each Ω_k leads to a data term expressed as a sum of regional data terms,

$$\Phi_D(T) = \sum_{k=1}^{n} \phi_D^k(T), \tag{3.16}$$

where ϕ_D^k is a similarity term (e.g. Φ_{CC} or Φ_{XC}) evaluated on Ω_k.

The regional form of similarity measure has an additional benefit when used with a mesh warp transformation. Let T be parameterized as a mesh warping, and choose Ω_k to be a sphere centered on the k^{th} control mesh vertex. Equation 3.16 can then be indexed in terms of the set of control mesh vertices, V, rather than using the integer k. Let ϕ_D^v denote the data term evaluated on Ω_v, a region centered at the position of v. The evaluation of ϕ_D^v depends on T through the function $\widetilde{J} = J \circ T$. The value of $T(x) = x + \Delta(x)$ for $x \in \Omega_v$ requires an interpolation of $\Delta(x)$. However, with the aim of quickly evaluating ϕ_D^v (evaluation time is critical inside the innermost loop of an optimization), $\Delta(x)$ is approximated as $\Delta(v)$. After doing so, the only parameter of T on which ϕ_D^v depends is $\Delta(v)$, the deformation vector associated with control vertex v. The data term can thus be written

$$\Phi_D(T) = \sum_{v \in V} \phi_D^v(\Delta(v)).$$

Each term of this sum is a function only of $\Delta(v)$ for *one* control mesh vertex v. It is shown in Section 3.4.2 that the regularization term $\Phi_R(T)$ can also be expressed as a sum of terms, each of which only depends on $\Delta(v)$ for a single control mesh vertex v. Thus the whole objective function can be put into a sum of such terms. The optimization problem is then said to be *separable* [NW99]: it can be solved using a series of independent optimization problems, each of which involves only a few variables, i.e. $\Delta(v)$ for one control mesh vertex v. This allows for parallel implementation, but more importantly each optimization problem involves only the three variables $\Delta_i(v), i = 1, 2, 3$, rather than $10^5 - 10^7$ variables in the full set $\{\Delta_i(v) : v \in V\}$ where V is the set of control mesh vertices. Performing an optimization on a small set of variables allows more sophisticated optimization techniques, which typically require quadratic storage [NW99]. Clearly quadratic storage is not an option with the full variable set.

Regional normalized cross correlation is the measure used by Collins' program ANIMAL [CNPE94, CHPE95], which is used for 3D experiments in this thesis.

3.2.3 Multiple Features

One advantage to using intensity features rather than geometric features is that the former are defined at every voxel of the image and so generally provide more information than does a sparse set of geometric features. However, using intensity-based measures alone has proved unsatisfactory for matching structures of the cortex (i.e. sulci or gyri) [CLGE98].

In order to aid the matching for difficult areas like the cortex, an intensity-based similarity measure can be augmented by a geometric feature. The geometric feature matching could be imposed as a hard constraint, using an interpolating transformation as described previously. Alternatively, the data term for the geometric feature (such as the expression 3.6) can simply be added to that of the intensity feature, leading to a data term like $\Phi_{CC} + \lambda\Phi_{LM}$, where λ is a parameter balancing the contributions of the two terms.

3.3 Model Terms

As alluded to at the beginning of this chapter, the space of transformations searched is too large. There are many implausible transformations (e.g. Figure 3.1) that result in a "good" match, when the match quality is measured by a data term alone. One way to alleviate this problem is to restrict the search to a suitable subset of the homeomorphisms. If the two images are known to be of the same individual the choice of subset is clear, namely the rigid-body transformations. However, this thesis addresses the problem of matching brain images of different individuals. The appropriate subset of homeomorphisms for this task is not presently known.

3.3.1 Regularization Rationale

Since the appropriate set of transformations is not known, the approach widely used is to search an overly-large class of transformations but to add a term (Φ_R) to the objective function that penalizes "undesirable" transformations [Tog99]. A sufficiently-large penalty serves to prevent the optimization from converging to such a transformation. There are two heuristic arguments used to motivate this penalty term. Each motivation sheds some light on the meaning of this term.

Bayesian

One approach is to view registration as a statistical decision process: which T should be chosen, given the data I and J [CMGA92, Gee99]? The decision is based on the posterior probability of T being the correct transformation, i.e. $\mathsf{P}(T|I,J)$. If all errors are viewed as equally serious (the "zero-one loss function"), the optimal choice of transformation is the *maximum a posteriori* solution, i.e. that which maximizes the posterior probability $\mathsf{P}(T|I,J)$ [DHS01].

The decision problem is cast as a minimization problem by using the negative logarithm, $-\log \mathsf{P}(T|I,J)$, as the objective function. Using Bayes theorem, $\mathsf{P}(T|I,J) = k\mathsf{P}(J|I,T)\mathsf{P}(T)$, where k is a constant independent of T, the problem becomes to minimize

$$\Phi(I,J,T) = -\log \mathsf{P}(J|I,T) - \log \mathsf{P}(T). \tag{3.17}$$

In Section 3.2.2 (see Equation 3.10, for example) it is argued that the data term (for intensity features) can often be written in the form $-\log \mathsf{P}(J|I,T)$, so the first term of Equation 3.17 can be identified with the data term, Φ_D, and the second can be identified with Φ_R.

Thus, Φ_R is related to $\mathsf{P}(T)$, the prior probability distribution on the space of transformations. This agrees with the earlier notion of introducing Φ_R in order to penalize "unlikely" transformations. However, this does not aid in actually computing $\mathsf{P}(T)$ or Φ_R. Construction of a good statistical model for the transformation distribution is an on-going research topic. For the case of a low-dimensional registration such as an affine transformation, it is feasible to estimate the probability distribution function for the transformation parameters by sampling registrations from the template to a population [ANC+97]. Or, if landmark points are used as the feature, the spatial distribution of each landmark can be learned and the "principal modes" of the distribution are then used to produce a prior distribution for T, e.g. [CET98, CET99].

In order to obtain a prior distribution $\mathsf{P}(T)$ this approach requires more samples than parameters, which is infeasible when dealing with transformations such as mesh warping that have many thousands of parameters. However, by projecting the observed deformation vectors Δ onto a low-dimensional subspace using the Gram-Schmidt orthonormalization process [Lan87], Gee and Le Briquer [GLB97, Gee99] are able to learn the principal modes of variation in the observed subspace. While this does not provide a prior for the mesh warping, it does provide $\mathsf{P}(T)$ for the observed subspace and can be used as the regularizer for an intermediate matching stage that is applied after affine registration and before mesh warping [GB99].

Tikhonov

Another point of view, coming from research in optical flow and other early-vision problems [PTK85], borrows ideas from Tikhonov's theory of regularizing "ill-posed" problems (see, e.g. [EHN96, Kir96]). A *well-posed problem* is one for which a solution exists, is unique, and depends continuously on initial data. Many problems fail to meet one or more of these three criteria, and are called *ill-posed*.

An ill-posed problem often turns out to be the inverse of a well-posed problem. Inverse linear problems have been well-studied since papers by Ivanov, Phillips and Tikhonov in the 1960s [BB98]. This classical theory concerns linear maps $K : F \to G$, where F and G are Hilbert spaces (F could be the space of square-integrable functions $\mathbb{R}^3 \to \mathbb{R}^3$, for example). It frequently happens that the forward problem of computing $g = Kf$ is well-posed, while the inverse problem (solve $Kf = g$ for f) is not. The inverse problem may not have a unique solution, or the solutions may not depend continuously on the input. The classical approach to an ill-posed inverse problem is to modify what is meant by "solution". The equation $Kf = g$ is replaced by a variational formulation in which f is sought to minimize a function

$$||Kf - g||_G^2 + \lambda ||Lf||_F^2, \tag{3.18}$$

where L is some operator, commonly the identity or a differential operator. The norms are associated with the space given by the subscript (i.e., $||\cdot||_G$ is a norm on space G), and are typically integrals $||g||_G^2 = \int_G g^2$. Suppose the inverse problem is ill-posed because the solution is not unique. The first term in Expression 3.18 is zero for many f, but with an appropriate operator L, the second term has a unique solution.

The first term in Expression 3.18 is analogous to the data term of image registration (with T playing the role of f and function composition with J playing the role of operator K); recall $\Phi_{SSD} = \int (J \circ T - I)^2$ for example. The second term in Expression 3.18 is called the regularization term, which is where the nomenclature for Φ_R comes from.

This form of regularizer can be viewed as imposing prior knowledge about the smoothness of the solution. In contrast to the previous section, the prior knowledge is not a probability distribution $\mathsf{P}(T)$, but rather the more vague notion that T ought to be "smooth". The common choice of a differential operator for L in Equation 3.18 effectively penalizes "high frequency" functions. To see this, suppose $f : \mathbb{R} \to \mathbb{R}$ is a function with Fourier transform $\widehat{f}(k) \equiv \int f(x) e^{2\pi i k x} dx$; then the Fourier transform of $f' \equiv df/dx$ is $-2\pi i k \widehat{f}(k)$. Using Parseval's equality $\int |f(x)|^2 dx = \int |\widehat{f}(k)|^2 dk$, so $\int |f'(x)|^2 dx = \int (4\pi k)^2 |\widehat{f}(k)|^2 dk$. Choosing $Lf \equiv f'$ as regularizer in Equation 3.18 effectively penalizes a transformation in proportion to the energy in each frequency component multiplied by the square of that frequency. Minimization with such a term tends to favour transformations with little contribution from higher frequencies, which are generally "smoother" than functions with high-frequency components. Differential regularizers can be generalized to higher-order derivatives, each order of differentiation adding a power of k^2 to the penalty.

3.3.2 Low-Order Differential Regularizers

At this time, choosing a prior probability distribution on T for mesh warping does not seem feasible, so a quadratic differential regularization term $\Phi_R = \int \phi_R$ is often adopted to impose smoothness [BK89, Ami94, SD02, MCAG93].

Following Cachier and Ayache [CA01b], the possible forms this penalty term might take are investigated. The integrand ϕ_R must be a scalar function, as argued in Section 3.1. Since the registration problem is formulated as a minimization, such a term must be bounded from below. It is no loss of generality to require the term be positive semidefinite.

The constraints imposed so far (positive semidefiniteness and invariance under coordinate transformation) leave a large number of possible penalty terms that can be investigated systematically according to the total degree in derivatives of Δ. The quadratic polynomials in $\partial\Delta_j/\partial x_k$ that are rotationally invariant and positive semidefinite can be systematically enumerated as follows. Recall that Δ_i form the components of a vector (rank 1 tensor) and its derivatives form the components of a rank 2 tensor. Introducing the notation

$$\Delta_{j,k} \equiv \frac{\partial \Delta_j}{\partial x_k},$$

quadratic terms $\Delta_{j,k}\Delta_{l,m}$ can be formed. These are rank 4 tensors from which scalars are obtained by performing two contractions. There are 6 ways to group the four indices (i, j, k, l) into pairs of indices to contract, but the order of contraction doesn't matter (i.e. $(j, k)(l, m)$ is the same as $(l, m)(j, k)$) so only half the groupings are distinct. There are thus three distinct pairings of the four indices: $(j, k)(l, m)$, $(j, l)(k, m)$, and $(j, m)(k, l)$, giving three possible quadratic terms

$$\phi_{ER}(\Delta) = \sum_{jk} \left(w_1 \Delta_{j,k}\Delta_{j,k} + w_2 \Delta_{j,k}\Delta_{k,j} + w_3 \Delta_{j,j}\Delta_{k,k} \right). \tag{3.19}$$

The other way to obtain a scalar from the rank 4 tensor $\Delta_{j,k}\Delta_{l,m}$ is to contract it with a rank 4 tensor that has the same value in all Cartesian coordinate frames. It turns out that all such rank 4 tensors are products of the Kronecker tensor (Equation 3.4) and leads to the same three terms as Equation 3.19 [CA01b].

Imposing the condition that the regularizer must be positive semi-definite implies the weights are restricted as follows [CA01b]

$$\begin{aligned} w_1 + w_2 &\geq 0 \\ w_1 - w_2 &\geq 0 \\ w_1 + w_2 + 3w_3 &\geq 0 \end{aligned} \tag{3.20}$$

Cachier and Ayache also characterize the rotationally-invariant quadratic differential forms of second order (there are five).

Algorithm 1 Basic Iterative Registration

```
1. Set $\{c_k\}$ to an initial estimate.
2. Estimate corrections $\{\delta c_k\}$.
3. Set $c_k \leftarrow c_k + \delta c_k$.
4. Repeat Steps 2-3 until done.
```

3.4 Iterative Registration

As discussed in Chapter 2, the transformation function $T : \mathbb{R}^3 \to \mathbb{R}^3$ is specified using a finite set of parameters, generically denoted $\{c_k\}$. This section deals with the problem of estimating those parameters.

In special cases, an analytic solution is available. An example of this is using four non-coplanar point pairs to specify an affine transformation (the 3D analogue of Equation 5.2). More often, however, the parameter estimation is an iterative procedure.

3.4.1 Basic Algorithm

Let $\{c_k\}$ denote the coefficients describing the transformation T that need to be estimated. The generic iterative algorithm is shown in Algorithm 1. The two details that need to be filled in are: (1) how the corrections are estimated, and (2) what is meant by "done". There are three routes taken in the literature. One method ("landmark matching") is only applicable to labelled geometric data terms. For other data terms, there are two possibilities. The most direct route is to discretize the objective function Φ and apply techniques of numerical optimization. This leads naturally to an iterative algorithm. The other route is to derive and then discretize a set of "force balance" (Euler-Lagrange) equations. Historically, the latter approach is taken first, in the work of Broit.

Landmark Matching

This method is applied when using labelled geometric data terms, whether interpolating or approximating the data. In the former case there is no data term in the objective function and $\Phi = \Phi_R$, while in the latter case the data term is typically Φ_{LM}. The minimizer turns out to be an expansion in terms of basis functions [Boo89, JM00]. The set of basis functions used is dictated by the form of Φ_R. The coefficients in the expansion are obtained through solving an $n \times n$ linear system of equations, where n is the number of landmark pairs in the data.

Discretized Euler-Lagrange Equations

Some intensity-based data terms, e.g. Φ_{SSD}, are summations that can be regarded as a discretization of an integral data term $\Phi_D = \int \phi_D(I, J, \Delta)$. Suppose that the regularizer is also an integral with the integrand built using first derivatives of Δ (e.g. Expression 3.19), $\Phi_R = \int \phi_R(I, J, \Delta)$. Then the complete objective function to minimize can be written

$$\Phi = \int \phi(\Delta_k, \Delta_{k,j}, x_j) \tag{3.21}$$

with

$$\phi = \phi_D + \phi_{ER} = \phi_D(\Delta) + \sum_{jk}(w_1\Delta_{j,k}\Delta_{j,k} + w_2\Delta_{j,k}\Delta_{k,j} + w_3\Delta_{j,j}\Delta_{k,k}). \tag{3.22}$$

Using standard techniques from the calculus of variations [CH53, GPS02], a necessary condition for Φ to be at a minimum can be derived. Setting the "first variation" of Φ to zero, results in the *Euler-Lagrange equations*, a set of differential equations that Δ must satisfy,

$$\sum_k \frac{\partial}{\partial x_k}\left(\frac{\partial \phi}{\partial \Delta_{j,k}}\right) - \frac{\partial \phi}{\partial \Delta_j} = 0. \tag{3.23}$$

These equations are analogous to the condition $f'(x_0) = 0$ for x_0 to be a minimum of a real-valued f. As is well-known, this condition only implies that f is stationary at x_0; it could be a maximum, minimum, or inflection point. The same is true of the Euler-Lagrange equations [CH53].

The Euler-Lagrange equations for ϕ as in Equation 3.22 are

$$\sum_k (2w_1\Delta_{j,kk} + 2(w_2 + w_3)\Delta_{k,kj}) - \frac{\partial \phi_D}{\partial \Delta_j} = 0.$$

With the replacements $2w_1 = \mu$, and $2(w_2 + w_3) = \mu + \lambda$, this equation can be re-written in more common notation as

$$\mu\nabla^2\Delta + (\mu + \lambda)\nabla(\nabla \cdot \Delta) = F, \tag{3.24}$$

where F is a vector with components

$$F_j = \partial\phi_D/\partial\Delta_j. \tag{3.25}$$

The Euler-Lagrange equations are a set of three coupled partial differential equations. This is the linearized elastic force equation introduced into image registration by Broit and Bajcsy (see, e.g. [BK89, GB99]). Equation 3.24 can be discretized using finite difference approximations for the derivatives, arriving at a sparse linear system that can be solved using standard methods.

The derivation of Cachier and Ayache [CA01b] just presented provides a rationale for choosing "elastic" regularization: Equation 3.24 is the most general expression involving rotationally-invariant quadratic forms of first derivatives in Δ. Of course, the equations of linear elasticity are derived with the same invariance properties in mind, so it is no surprise that they coincide.

Discretized Objective Function

There are a number of methods that work directly from the objective function Φ rather than deriving the Euler-Lagrange equations [CRM93, GP96, LBS99]. In each case, the strategy is to substitute the parameterized formula for Δ into Φ and use numerical optimization methods to find optimal values for the parameter set $\{c_k\}$.

Suppose $\Phi_R = \int ||L\Delta||^2$, where L is a linear differential operator that has a complete set of orthonormal eigenfunctions, $\{\phi^k\}$. Let λ_k be the eigenvalue associated with ϕ^k, i.e. $L\phi^k = \lambda_k\phi^k$. The orthonormality condition implies $\int \phi^k \cdot \phi^l = \delta_{kl}$. Then the transformation function can be conveniently parameterized using the eigenfunctions as a basis, as described in Section 2.4.1. An n-term approximation is written as $\Delta = \sum_{k=1}^{n} c_k\phi^k$ [MCAG93], leading to a very simple expression for the regularization term,

$$\begin{aligned}
\Phi_R &= \int ||L\sum_{k=1}^{n} c_k\phi^k||^2 \\
&= \int ||\sum_{k=1}^{n} c_k\lambda_k\phi^k||^2 \\
&= \sum_{k=1}^{n}\sum_{l=1}^{n} c_k c_l \lambda_k \lambda_l \int \phi^k \cdot \phi^l \\
&= \sum_{k=1}^{n} c_k^2 \lambda_k^2.
\end{aligned}$$

Another method of discretizing the objective function is to simply approximate the integral $\Phi = \int(\phi_D(I, J, T) + \phi_R(T))$ as a sum over the points of the control mesh vertices, approximating any derivatives by finite differences [MCAG93]. This approach fits well with algorithms that parameterize the transformation using mesh warping.

A third approach is to apply the finite element method [Bic94], in which the domain of integration is subdivided into small elements (e.g. the parallelepipeds used in mesh warping [GH99]) and the integrand is expanded over basis functions with support in a single domain element. The basis functions are simple enough (generally low-order polynomials) that most of the terms in the integral can be evaluated analytically. However, the data terms involve $J(T(x))$ and are often approximated using a quadratic function of Δ [GB99].

3.4.2 Two-Step Registration

When the transformation is a high dimensional warping (Section 2.4), it is not uncommon to have on the order of a million parameters: for example, a mesh warping with a $100 \times 100 \times 100$ control mesh and 3 parameters per control mesh vertex constitutes 3×10^6 parameters. With so many parameters to estimate, it is important to choose an effective optimization technique. Each of the basic methods described in the preceding section results in a single objective function containing all the parameters to estimate. Faced with such a large system, some researchers use simple steepest descent methods [MCAG93] which can be slow to converge [NW99].

One approach to handling this complexity is to take advantage of the sparse structure of Φ. An expression for differential regularization, for example, whether evaluated using finite differences or using finite elements, results in terms involving the deformation vector Δ at nearby control mesh points only. This sparsity can be used to advantage [GH99] in a Newton or quasi-Newton method [NW99, DS96].

A second strategy for handling the large number of parameters to estimate is to separate the problem into a number of smaller optimizations [NW99]. One of the advantages of using a regional similarity measure (Section 3.2.2) with mesh warping is that the data term is separable. The goal of this section is to show how the regularization term can also be made separable, by transforming the problem into two interleaved minimizations. The resulting registration has two steps at each iteration, which can be interpreted as a "matching" step followed by a "smoothing" step.

The modified problem formally uses *two* transformations, one denoted $U : \mathbb{R}^3 \rightarrow$

$\mathbb{R}^3$ for the data term and one denoted T for the regularization term. A third term is introduced into the objective function to link the transformations, resulting in

$$\Phi(T, U) = \Phi_D(U) + \frac{1}{2}\int ||T - U||^2 + \Phi_R(T). \tag{3.26}$$

After minimizing over both T and U, the transformation T is returned as the final solution. In other words, the registration problem solved is

$$\arg\min_T \Phi'(T),$$

where

$$\Phi'(T) = \arg\min_U \Phi(T, U).$$

The new objective function, Equation 3.26, has twice as many variables as the original objective function, $\Phi_D + \Phi_R$. An iterative minimization method such as those described in Section 3.4.1 is used, with half the variables held fixed at each iteration. One iteration fixes the variables that define T while the next iteration fixes the variables that define U. In this manner, the optimization of T and of U can be interleaved, resulting in two "half iteration" steps. First, fix T and find

$$U = \arg\min_{U'} \Phi(T, U') = \arg\min_{U'} \Phi_D(U') + \frac{1}{2}\int ||T - U'||^2. \tag{3.27}$$

Then, fix U and find

$$T = \arg\min_{T'} \Phi(T', U) = \arg\min_{T'} \frac{1}{2}\int ||T' - U||^2 + \Phi_R(T'). \tag{3.28}$$

The first step (Expression 3.27) can be seen as finding a transformation, U, that matches the data while remaining close to the previous solution, T. The term $\int ||T - U'||^2$ can be discretized as a sum over the control mesh vertices, each term of which involves only the displacement change $T - U$ at that control mesh vertex. Thus the first step of finding an optimal U is separable if the data term is separable. The second step (Expression 3.28) can be seen as regularizing the transformation U to produce a smoother transformation T. An efficient implementation of this step as a filtering operation is described below.

Cohen [Coh96] uses this kind of transformation in a situation where the regularization term is convex, but the data term is not convex. Instead of using the given data term, Φ_D, in the first step, Cohen introduces a transformed data term Φ'_D and iterates over the two steps

$$U = \arg\min_{U'} \Phi'_D(U') + \frac{1}{2}\int ||T - U'||^2, \tag{3.29}$$

and Equation 3.28. By choosing Φ'_D carefully, the expression

$$\Phi''_D(T) \equiv \min_{U'} \Phi'_D(U') + \frac{1}{2}\int ||T - U'||^2$$

has a minimum over U' for every T, and is equal to $\Phi_D(T)$ [Coh96]. Thus, the algorithm ends up solving

$$\begin{aligned} T &= \arg\min_{T'}\{\min_{U'} \Phi'_D(U') + \tfrac{1}{2}\textstyle\int ||T' - U'||^2 + \Phi_R(T')\} \\ &= \arg\min_{T'} \Phi_D(T') + \Phi_R(T') \end{aligned}$$

where Φ_D is the original data term for the problem. Moreover each half-iteration is a convex problem.

Cachier and Ayache [CA01a] point out that splitting the optimization into two steps has advantages, even if Equation 3.27 is used instead of Equation 3.29, i.e. even without transforming the data term. In their work, Φ_{SSD} is used as the data term, and the modified objective function minimized by the two-step algorithm performed better than when using the original data term.

The second advantage of a two-step method is that the regularization step (Equation 3.28), if Φ_R is formed of quadratic terms of derivatives, can be implemented by filtering [NFD97]. For example, suppose

$$\Phi_R(T) = \frac{\lambda}{2}\sum_{lm} T_{l,m}T_{l,m}.$$

Working in the Fourier domain (which means assuming periodic boundary conditions on the domain of interest), $\widehat{T_l}(k) = \int T_l(x)e^{2\pi ikx}dx$ and the Fourier transform of $T_{l,m}$ is $\int T_{l,m}(x)e^{2\pi ikx}dx = -\int 2\pi ik_m T_l(x)e^{2\pi ikx}dx = -2\pi ik_m\widehat{T_l}$. Using Parseval's equality, Equation 3.28 can be written as

$$T = \arg\min_{T'} \int \frac{1}{2}\sum_l \left((\widehat{T'}_l - \widehat{U}_l)^2 + \lambda\sum_m (2\pi k_m)^2 \widehat{T'}_l\widehat{T'}_l\right) dk.$$

The minimizer can be obtained by formally taking the derivative with respect to $\widehat{T'}_l$

Algorithm 2 Two-Step Registration.

```
1. Set $T$ to initial estimate.
2. Minimize $\Phi_D(U) + \Psi(T - U)$, where the term $\Psi$ penalizes deviation of $U$ from $T$.
3. Set $T$ to be the smoothed version of $U$.
4. Repeat steps 2-3 until done.
```

and setting it to zero, resulting in

$$\widehat{T}_l - \widehat{U}_l + \lambda(2\pi)^2 k^2 \widehat{T}_l = 0,$$

or

$$\widehat{T}_l = \frac{1}{1 + \lambda(2\pi)^2 k^2} \widehat{U}_l.$$

This is the Fourier domain equation for a linear filter. Thus T may be obtained by convolution with a filter whose kernel is the inverse Fourier transform of

$$\widehat{h}(k) = \frac{1}{1 + \lambda(2\pi)^2 k^2}.$$

So, given a particular choice of quadratic regularizer, Φ_R, a filter can be derived. This point of view can be reversed: given a filter, the form of Φ_R may be derived. This is only possible for certain filters. However, the goal of registration is to match images not to impose a certain form of regularizer, so it isn't necessary that the filter have a corresponding expression in terms of a quadratic form of derivatives. In this spirit, the notion of a two-step algorithm is broadened. The first step searches for a transformation U that minimizes the data term with a control on how much U deviates from T. Equation 3.27 is of that form, but controls other than the quadratic expression $||T - U||^2$ are permitted. The second step sets T to be a smoothed version of U using some kind of filtering operation. This is summarized in Algorithm 2. One cost of broadening the criterion for the two steps is that the resulting algorithm may no longer be interpretable as a minimization problem.

3.4.3 Four-Step Registration

There is ample experimental evidence, e.g. [CRM$^+$95], that "elastic" registration using $\Phi = \Phi_{SSD} + \int \phi_{ER}$ can result in a non-invertible transformation if the regularization parameters w_1, w_2, w_3 of Equation 3.19 are too small. Increasing the regularization strength alleviates this at a cost to the matching criterion Φ_{SSD}. Christensen et al. [CRM94, CRM96] ascribe this problem to the tendency of transformations based on ϕ_{ER} to "develop restoring forces proportional to the deformed distance" [CRM94], preventing the transformation from matching fine detail.

Christensen et al. [CRM93, CRM96] propose instead to base the registration penalty on a different physical model, that of a viscous fluid. Given this model, they proceed to derive a matching procedure termed *fluid registration* that repeatedly adds a smoothed correction field to the current transformation. The implementation of this algorithm is now described; see [CRM96] for a derivation of the procedure from the laws of fluid mechanics.

Break up the transformation Δ into a sum $\Delta = \delta^1 + \delta^2 + \cdots + \delta^n$. Each δ^j is obtained in sequence, by fixing δ^i $(i < j)$ and searching for

$$\delta^j = \arg\min_{\delta} \int \phi_D(\delta^1 + \cdots + \delta^{j-1} + \delta) + \int \phi_{ER}(\delta), \tag{3.30}$$

with ϕ_{ER} given by Equation 3.19. Writing $\phi = \phi_D(\delta^1 + \cdots + \delta^{j-1} + \delta) + \int \phi_{ER}(\delta)$, the optimization in Equation 3.30 can be seen to be the same as the optimization in Equations 3.21 and 3.22, but with the optimization over δ rather than Δ. The Euler-Lagrange equations are the analogue of Equation 3.24,

$$\mu\nabla^2\delta + (\mu + \lambda)\nabla(\nabla \cdot \delta) = F, \tag{3.31}$$

with $F_k = \partial\phi_D/\partial\delta_k$. Under the assumption that correction δ is small, Equation 3.31 is solved using the approximation that the force F is constant, i.e., given by $F_k = (\partial\phi_D/\partial\delta_k)_{\delta=0}$[CRM96].

Bro-Nielsen points out that with constant force term F Equation 3.31 can be implemented as a filtering of F, which is "at least an order of magnitude faster" than the original implementation of Christensen et al. [BNG96]. In the Bro-Nielsen formulation, the j^{th} iteration of the algorithm estimates δ^j by a filtering (i.e. smoothing) operation, reminiscent of the smoothing step of Algorithm 2. The difference is that in fluid registration the corrections δ^j are smoothed rather than the total displacement Δ (or, equivalently the transformation T). Both algorithms can be put into a common framework [CPA99] displayed in Algorithm 3. If only smoothing of δ done the algorithm is described as a fluid registration algorithm, even if the smoothing is not that associated with ϕ_{ER}. Similarly, if only smoothing of Δ is performed the

Algorithm 3 Four-Step Registration.

```
1. Set $\Delta$ to initial estimate.
2. Let $\delta = \arg\min_{\delta'} \Phi(\Delta + \delta')$.
3. Smooth $\delta$.
4. Set $\Delta = \Delta + \delta$.
5. Smooth $\Delta$.
6. Repeat steps 2-5 until done.
```

method is termed an elastic registration, even if the smoothing is not that associated with ϕ_{ER}.

3.5 Coarse-to-Fine Hierarchy

Registration is almost universally performed with multiple resolutions of the transformation [BK89, CHPE95, KPH$^+$96, CJM97, RSH$^+$99, KLF99, RCRM01, SD02]. When the transformation is parameterized using mesh warping, a very coarse grid is used first, say with vertex spacing 8 times the image voxel spacing. One of the algorithms discussed so far is run to completion. The mesh warping grid is then refined by a factor of two (to 4 times image voxel spacing), setting the value $\Delta(v)$ at new nodes v by interpolating values from the old (coarse) control mesh. The algorithm is run again, the mesh refined again, and so on.

An alternative strategy is possible when the transformation is expanded in terms of Fourier or wavelet basis functions: simply start with the low-frequency basis functions and refine the warp by adding higher frequency terms to the set of basis functions [CJM97, Ami94].

The idea behind these strategies is to speed up the optimization by aligning larger structures at the coarse levels, hopefully bringing the smaller structures into a better starting position for registration on the finer meshes. The goal is to have the registration converge more rapidly and to a better minimum [KB99].

3.6 Brain Image Registration Literature

3.6.1 Labelled Features

Landmark matching approaches use labelled geometric features, generally points, curves, or surface patches. Bookstein uses a landmark interpolating transformation obtained with a regularizer whose minimizer is a thin-plate spline transformation [Boo89, Boo91, Boo99]. Other splines are used for image matching [GT99], including "fluid model" splines of Joshi [Jos98, JM00, MJC99] that assure an injective mapping.

The number of easily-identifiable landmarks in the brain is limited, and so sometimes extra landmarks are generated heuristically, e.g. Van Essen et al. take triples of existing landmarks and add the centroid of the triple as an extra landmark [VELD+01].

Approaches for automatically extracting the features from image data are described for geometrically-salient points [IAC99], sulci represented using curves [TG93, DSTA95, Loh98, Sub99], medial sulcal surfaces [SBK+92, VDB96, LGBB97, ZTT99], sulcal regions on cortex [RHP00] and 3D sulcal basins [LvC98]. However, very often the feature is obtained manually [FPR85] or semi-manually, e.g. [TT96, Dav96, ZSS+99].

Automated Labelling

Amit and Kong [AK96, Ami97] locate potential landmark points using image processing techniques and then label the landmarks by fitting a "decomposable" graph shape model to the set of landmarks. In this context, decomposable means a triangulated, embedded graph that can be reduced to a single triangle by repeatedly deleting a vertex of degree 2. Fitting the model (with a label at each vertex) to the image serves to label the features.

Automated labelling for sulci is described by Royackkers et al. [RFD+95, RDFR99], Mangin et al. [MRB+95, MFB+95, RMPO+00, RMPO+02], and Le Goualher et al. [LGBB97, LGPC+99]. The labelling is typically based on location in space and simple shape descriptors, combined with neighbourhood information in the case of Mangin et al.

3.6.2 Unlabelled Features

Elastic Model

Determining the "optical flow" between two consecutive frames in a movie sequence has a long history in computer vision [HS81]. It is, in essence, a nonaffine registration problem using Φ_{SSD} as the data term. Various regularizations of the problem are proposed such as the square of the Laplacian, $||\nabla^2\Delta||^2$ [HS80].

In his 1981 thesis, Broit introduces elastic regularization for image matching. Bajcsy and Kovačič extend the idea to include a coarse-to-fine hierarchy [BK89]. Broit's original approach (described in, e.g., [GH99]) uses normalized cross-correlation as the data term, and a finite difference discretization of the equations on a regular grid. The equations are nonlinear in Δ due to the data term, i.e. to the term F in Equation 3.24. Broit obtains an expression for F that is linear in Δ from a Taylor expansion of ϕ_D about the current estimate for Δ, truncated at second order. The derivative of the quadratic approximation gives an expression linear in Δ for the force (Equation 3.25), resulting in a linear system of Euler-Lagrange equations, which is solved using Jacobi relaxation. The approximation to F is then updated with the new estimate of Δ for the next iteration of relaxation. The iterations terminate when the vector F is approximately zero.

Christensen et al.'s work on elastic matching [CMGA92, CRM94, CJM97, Chr99, MCAG93] uses Φ_{SSD} as the data term and an expansion $\Delta(x) = \sum_{k=1}^{n} c_k\phi^k(x)$ where ϕ^k are eigenfunctions of the operator $L = -a\nabla^2 - b\nabla\nabla\cdot + c$ and the chosen boundary conditions. With this form of Δ, the integral in the regularization term $\Phi_R = \int ||L\Delta||^2$ is easily evaluated in terms of eigenvalues and the coefficients c_k. Minimization of $\Phi(c_k)$ is achieved using steepest descent (on $\partial\Phi/\partial c_k$) with fixed step size. A coarse-to-fine matching is implemented by increasing the number of basis functions used, n, in sequence from low to high frequency.

Gee et al. [GH99] use a Ritz finite element approach [Bic94] to transform Φ into a function of the n parameters $\{c_k\}$. A series of quadratic problems is solved, approximating the data term as Broit did (above).

Thirion's "diffusing models" [Thi98, Thi99] are typically implemented using Φ_{SSD} as the data term and elastic smoothing with a Gaussian kernel [CPA99].

Lavallée et al. [LBS99] use an "octree spline", in effect a selectively refined control mesh for mesh warping. The mesh can be left coarse in regions for which the transformation is homogeneous, while being refined in regions where the transformation changes rapidly. They register images using a feature that consists of a landmark point location together with other characteristics such as normal, gradient vector, curvature values, image intensity, etc. A differential regularizer is used.

Rueckert et al. use a normalized mutual information data term with thin-plate spline bending energy regularizer [RSH$^+$99].

The HAMMER algorithm of Shen and Davatzikos [SD02] uses a data term that combines image intensity with information from tissue classification. The underlying parameterization of the transformation is a mesh warping, but the deformation is initially estimated only on a subset of "driving" control mesh vertices. The set of driving vertices increases to the entire set of control mesh vertices as the iterations proceed. A Laplacian regularizer is used to smooth the warp.

Fluid Model

Christensen et al. introduce the fluid model [CRM93, CRM94, CRM96, CJM97, Chr99] for image registration. Their implementation of the smoothing (step 3 of Algorithm 3) solves the Euler-Lagrange equations on a spatial grid with successive overrelaxation and consequently is quite slow. Bro-Nielsen and Gramkow [BNG96] point out that the smoothing can be done much faster, when implemented as a filtering of the force term.

The stated advantage of this model is that the resulting transformation (under certain conditions [Ami94, DGM98, JM00]) is invertible, even for large deformations.

Other

Woods et al. expand the transformation using low-degree (up to fifth order) polynomial basis functions with global support. The original "variance of intensity ratio" data term [WCM92], proposed for registering PET images, is also used for PET-MRI registration [WMC93]. In later work [WGH$^+$98, WGW$^+$98, Woo99] the Φ_{SSD} term is used, optionally with a single global scaling (Equation 3.11).

Friston et al. [FAF$^+$95, AF99a] use the Φ_{SSD} data term and regularize the problem by parameterizing T using a moderate number (up to 512 [AF99b]) of cosine functions for the basis. The parameter set is small enough that they can use the Gauss-Newton method [NW99] for nonlinear least squares to estimate them.

Ashburner et al. [AAF00] subdivide the domain of T into tetrahedra and use a piecewise-affine transformation. The model term is designed to penalize both small and large Jacobian determinant values in order to encourage both T and T^{-1} to have moderate, positive Jacobian determinant values. While the resulting warp will bend the template locally to match the input image, Δ is not very smooth as there is nothing to enforce long range coherence in the mapping.

Some methods that use geometric data terms attempt to generate a matching at the same time as estimating the best transformation, e.g. the iterated closest points [BM92] and the softassign methods [PGR96, RCB97, RCD99]. Feldmar and Ayache generalize this to non-rigid mappings for points lying on a surface [FA96]. Subsol et al. [DSTA95, Sub99] extend ICP to generate a consistent matching of points lying on polygonal lines (e.g. crest lines of sulci and gyri) and use cubic splines for the transformation.

Kochunov et al. [KLF99, KLT$^+$00] use an octree rather than a full grid to parameterize the mesh warping.

Davatzikos and Vaillant [Dav96, TT96, CLGE98, BGKM98, WS98, VD99] and Thompson and Toga [TT96] both use landmark matching in a hierarchical fashion. From matched sulcal curves, a set of points is digitized, and used as the data term in a surface warping (see Chapter 6) and the resulting surface map is used as a boundary condition in generating the volumetric warp.

Intermodal Registration

The assumption that the intensity values of I and $\widetilde{J}$ are affinely related is a reasonable one for same-modality matching. For cross-modality registration, the assumptions have to be further relaxed. If all that can be assumed is that there exists a functional relationship (without knowledge of the form of this function), then the maximum likelihood method is equivalent to using the so-called "correlation ratio" [RMPA98, RMA00]. This turns out to be closely related to Woods' variance of intensity ratios measure [WMC93]. If even a functional relationship between the images' intensities cannot be assumed, but only a statistical relationship, then maximum likelihood is equivalent to using the mutual information [CMD$^+$95, WIVA$^+$96]. This measure is very useful in cross-modality matching, but for matching MR to MR, it may have a smaller basin of attraction and be less robust [RMA99].

Most of the multimodal registration approaches are intended to register the same anatomy, and so search for a rigid-body transformation. Guimond et al. [GRAM01], however, jointly estimate a mesh warping and polynomial intensity remapping.

3.6.3 Multiple Features

Adding a sulcus-based term is investigated by Collins et al. [CLGE98], Hellier et al. [HB00], and Cachier et al. [CMP$^+$01]. In each case, a matching (or labelling) of the sulci is performed and the registration data term consists of a distance-based penalty term in addition to the original image intensity term. In Collins' work, each sulcus in the input image I is represented by a set of points, and the distance to the *nearest* point of the corresponding sulcus is used, reminiscent of the iterated closest points approach. Hellier represents the sulci using B-splines. Using the assumption that the splines' control points should match, the sum of squared distance between corresponding control points is added to the objective function. Cachier uses points along the bottom and along the top of the extracted sulci. The distance penalty is the squared distance to the closest matching sulcus point (matching top to top and bottom to bottom). The sulcus top curves are known to be much more variable than the sulcal fundi, and some of the more variable sulci are not robustly labelled by the automated procedure used. Therefore, the weighting for sulcus top points and for the "un-trustworthy" sulcus bottom points is multiplied by an exponentially-decaying function of distance, to avoid undue penalty.

3.7 Algorithm for 3D Experiments: ANIMAL

The 3D spatial normalization experiments for this thesis are carried out using the ANIMAL algorithm [CE97] as a prototypical nonaffine registration method. This section elaborates on the algorithm with attention to the numerical parameters the user must choose.

The deformation function Δ estimated by ANIMAL is parameterized as a mesh warping (Definition 2.4.1) with cubic interpolation using a Catmull-Rom spline [FvDFH90]. ANIMAL operates with a coarse-to-fine outer loop, and a two-step inner loop, i.e. an elastic registration. The transformation function is specified in a "world" coordinate system, which is defined independently of the source and target image voxel grids. Each image is endowed with the affine transformation function between world coordinates and its own voxel grid. Lengths in the world coordinate system, such as the control mesh vertex spacing, are given in units of millimeters.

Algorithm 4 Inner loop of ANIMAL.

```
1. Optimize $\Phi(\{\delta^v\}) = \sum_v (a_1\phi_D^v(\Delta^v + \delta^v) + (1 - a_1)\psi(||\delta^v||))$.
2. Let $\Delta^v = \Delta^v + a_2\delta^v$.
3. Let $\overline{\Delta^v}$ be mean displacement of 26-neighbours of $v$.
   Set $\Delta^v = a_3\overline{\Delta^v} + (1 - a_3)\Delta^v$.
4. Loop over Steps 1-3 a fixed number of times.
```

3.7.1 Outer Loop

The first iteration of the outer loop employs a control mesh with a vertex spacing of 8 mm and is referred to as the 8 mm grid. The feature used in the match is the two input images, each blurred using an isotropic Gaussian kernel with FWHM=8 mm. The next two iterations use a control mesh with a vertex spacing of 4 mm (4 mm grid) and 2 mm (2 mm grid). The input images are smoothed with an isotropic Gaussian kernel of FWHM=8 mm and FWHM=4 mm, respectively. Finally, a fourth iteration with a vertex spacing of 2 mm is done using smoothed (FWHM=4 mm) gradient magnitude images.

The initial iterate for the inner loop is interpolated from the result of the previous iteration of the outer loop, except the first iteration which starts with zero displacements.

3.7.2 Inner Loop

Using v to index the control mesh vertices, Δ^v is the current estimated displacement vector for vertex v, and δ^v is the correction to Δ^v estimated at each iteration of the inner loop. The inner loop of ANIMAL, displayed in Algorithm 4, is comparable to Lines 2 and 3 of Algorithm 2, with T replaced by Δ and U replaced by $\Delta + \delta$.

The objective function of Line 1 is comprised of two terms for each control mesh vertex. The first term, ϕ_D^v, is a regional image similarity measure, with the region for vertex v being a sphere of radius 1.5 times the control mesh vertex spacing. The similarity measure is either normalized cross correlation Φ_{XC}, or correlation coefficient Φ_{CC}. The second term, ψ, is an increasing function that approaches ∞ at a finite value of $||\delta^v||$, thus limiting the size of the correction vector. The parameter $a_1 \in [0, 1]$ balances the similarity and cost of correction step δ^v, and is known as the *similarity-cost ratio*.

The objective Φ is separable so the optimization is performed independently for each v. The result is a large number of small optimization problems, as δ^v has just three variables to optimize: the displacement in the x-, y-, and z-directions. Having only a small-dimensional optimization means that the quadratic-storage Nelder-Mead downhill simplex algorithm [PFTV88] can be used. This is an advantage because the derivative of the objective function, which uses Φ_{CC} or Φ_{XC}, is then not required. These data terms (Equations 3.13 and 3.14, respectively), are complicated functions of the displacement vectors through the function $\widetilde{J} = J \circ T$.

The update step of Line 2 employs a *weight* parameter a_2. The displacements are under-corrected if $a_2 < 1$ or over-corrected if $a_2 > 1$.

The displacement vector Δ^v is smoothed in Line 3 by taking a weighted sum of the current displacement estimate with the mean displacement of the 26 neighbours in the $3 \times 3 \times 3$ control mesh neighbourhood centered on v. The *stiffness* parameter $a_3 \in [0, 1]$ balances the two terms.

The three parameters a_1, a_2, and a_3 need to be specified in order to complete the description of ANIMAL. Collins and Evans choose values of 0.5, 0.6, and 0.5, respectively [CE97]. These values are obtained empirically using visual inspection of the displacements and resampled images to judge registration quality. The next chapter proposes a more objective method to select these parameters.

Node Thinning

The optimization at control mesh vertex v is *not* performed if the source image value at that location falls below 10% of the maximum source image value. Such locations are likely to be background and are skipped since there is nothing to be gained by fitting background regions that are dominated by noise. This heuristic is termed *node thinning.*

For the iterations using intensity data, this heuristic retains nearly all the nodes lying in brain tissue, while skipping control mesh vertices located outside of the head. The image data and corresponding node-thinning mask for the third outer iteration (using FWHM=4mm blurred data) is shown in Figure 3.4.

In the next chapter, this heuristic is shown to be detrimental to the gradient fit.

3.8 Summary

The purpose of this chapter is to survey the components and principles of a 3D image registration algorithm and, in particular, of MR image registration. The algorithms are put into a common framework of optimization in order that they may

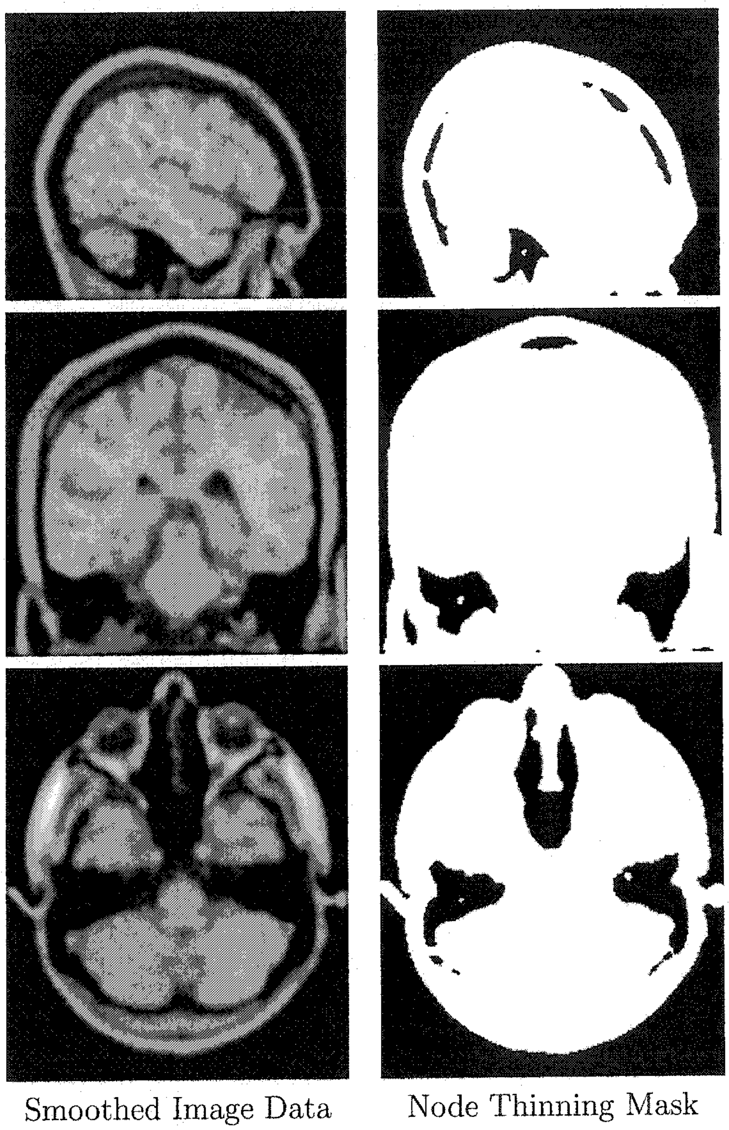


Figure 3.4: Left column shows the image data, smoothed with a FWHM=4 mm Gaussian kernel, the input data for the third iteration of the ANIMAL outer loop. Right column shows the corresponding mask used for node thinning heuristic. The displacement is only estimated for a control mesh vertex that lies in the white region. This strategy reduces the processing time, as vertices lying in the background region outside the head are skipped.

be compared, e.g. on the basis of the data and model terms employed in the objective function, $\Phi(I, J, T)$.

The principle of coordinate invariance is used to guide the choice of terms in the objective function, even if approximations used in a practical implementation (e.g. discretizing an integral) breaks strict invariance. The data terms involve only scalar values such as distance, image intensity, or image gradient magnitude, and so are easily seen to be invariant. The model terms, however, are constrained by the invariance requirement, as discussed in Section 3.3.2.

3.8.1 Data Terms

Data terms for registration are generally constructed either using image intensity or using a geometric feature extracted from the image. In the latter case a labelling, i.e., an *a priori* pairing between features in the source and target images may be given. There are a number of data terms in use for labelled features, as discussed in Section 3.2.1.

Chapter 2 argues that a high-dimensional warping transformation should be used for matching brains. This implies a reasonably large set of features that is well-distributed throughout the brain, so that the estimation of each parameter has some data with which to work. One of the best systems currently available for automated feature labelling can label 60 sulci per hemisphere with 76% of the labelled voxels in accordance with a manual labelling, which is considered good performance [RMPO$^+$02]. In fact a number of the disagreements with the manual labelling are observed to be ambiguous situations where the automatic labelling is preferred [RMPO$^+$02], so the success rate of voxel labelling is somewhat higher. Nevertheless, the number of voxels that are labelled by the system is small compared to the million or so parameters involved in specifying a grid warping, e.g. a 3-vector on a grid of size $100 \times 100 \times 100$. Thus label data can not be used alone to drive the registration, though it can be used in concert with intensity data [CLGE98, HB00, CMP$^+$01]. This thesis concentrates on first obtaining the best spatial normalization possible using intensity features alone. If the normalization is not sufficient, then future work can incorporate label data.

Section 3.2.2 presents conventional data terms based on image intensity in a framework of maximum likelihood [DHS01], i.e. in terms of maximizing $\mathsf{P}(\widetilde{J}|I)$, the probability of observing image $\widetilde{J} \equiv J \circ T$ given that I is the "true" image. In this framework, the data terms discussed correspond to an assumption on the functional relationship (linear, affine, etc) between the intensity generated by a given tissue type in image I and in image J [RMA00]. As Figure 3.3 demonstrates, the intensity of gray or of white matter is not equal in each MR image, though a linear or affine transformation is likely to be sufficient to match intensities. A linear relation maps intensity i to intensity αi, which means that only one intensity from each image is required to solve for the coefficient α. If the intensity of, say, the gray matter is chosen then gray matter intensity will match exactly but the white matter intensity may be mismatched. An affine relationship, on the other hand, maps intensity $i \mapsto \alpha i + \beta$. Using an affine mapping, two tissue types can be intensity-matched. Assuming an affine intensity relationship means that the correlation coefficient data term, Φ_{CC} (Equation 3.13) should be used, while a linear relationship implies the cross correlation, Φ_{XC} (Equation 3.14).

Any of the intensity data terms in Section 3.2.2 can be evaluated across the whole of images I and J. However, breaking the data term into a sum of such terms, each of which is evaluated on a small region of the domain, offers two advantages. First, since a regional similarity measure can be seen as a piecewise-constant approximation to letting the intensity relationship vary spatially, some extra protection against the intensity nonuniformity artifact [SZE98] is obtained, as explained on page 36. Second, when combined with a mesh warp transformation, the evaluation region can be chosen to centre around control mesh vertex v, leading to a data term in the form of a sum with one term per control mesh vertex. This brings the data term into a form where each parameter is estimated locally. If the entire objective function (model terms as well as data terms) is of this form, the problem becomes separable, which has the great advantage [NW99] of being able to perform the optimization independently for each control mesh vertex.

3.8.2 Model Terms

The goal of the model term, as discussed in Section 3.3, is to insert into the algorithm some prior knowledge of the solution. This can be very specific knowledge, e.g. of the distribution of transformation parameters obtained from statistical sampling [CET98, CT99, CET99]. Such sampling is not feasible for the large parameter set of a mesh warping, though a subspace can be sampled [GLB97, Gee99] and used as one step in the algorithm [GB99].

Thus, mesh warping based registration generally uses the less-specific knowledge that the transformation should be "smooth", which means in this context that the derivatives of T are small. This smoothness requirement is implemented either by adding a regularization term Φ_R that penalizes derivatives, e.g. $(\sum_k \partial T_k / \partial x_k)^2$, or by explicitly smoothing the deformation field in the algorithm, such as in Step 3 of the two-step algorithm (Algorithm 2).

3.8.3 Registration Algorithm

The registration algorithm is fundamentally a minimization of the objective function, Φ. The choice of data term and of transformation parameterization influence the approach used for the optimization, as discussed in Section 3.4.

For a mesh warping with a separable data term (i.e. using a regional similarity measure), it is a computational advantage to keep the model term separable [NW99]. With a regularization term expressed as a quadratic in derivatives, the two-step method of Cohen [Coh96] transforms the optimization into an iteration of a data matching step followed by a transformation smoothing step, as discussed in Section 3.4.2. More generally, an ad-hoc smoothing step can be used, as is done by the ANIMAL algorithm [CHPE95].

The smoothing step can be moved from smoothing the deformation field, Δ, to smoothing the deformation field corrections, δ. The result is the four-step algorithm (Algorithm 3), initially motivated by a viscous fluid model [CRM93]. The claimed advantage of the four-step method over the two-step method is that the former allows injective matching with larger deformations, under certain conditions [Ami94, DGM98, JM00]. In order to ascertain whether the fluid model is required for spatial normalization, this thesis concentrates on producing the best performance possible from a two-step method. If that matching is determined to be inadequate, future work can use a four-step method. Specifically, the 3D experimental work in the next chapter uses ANIMAL, a two-step algorithm which employs a regional cross-correlation data term and ad-hoc smoothing.

A coarse-to-fine hierarchy is standard practice [KB99] with any of the basic registration algorithms. The specific hierarchy used in ANIMAL increases the data complexity in step with the coarse-to-fine warping [LA99] by performing less smoothing of the data at each level. The idea is to reduce the number of small minima in the data term and, in so doing, increase the chance of arriving at a good minimum of the objective [Col94].

Chapter 4

Quantifying Spatial Variability

The goal of spatial normalization in brain imaging is to remove, to the extent possible, the natural anatomical variability in a population by warping each individual's anatomy into a standardized space. This residual variability must be quantified, in order to compare the efficacy of two spatial normalization methods. This chapter presents two different strategies for measuring anatomical variability. The first is automated and gives a global measure of variability. It is used to tune the numerical parameters of a registration algorithm. The second method, which relies on manual delineation of a structure such as a sulcus, gives a measure that is interpretable as the "variance" of that structure about a mean structure. The second quantity is used to judge the quality of a registration method for the purpose of spatial normalization.

Anatomical variability is often visualized qualitatively in the "sharpness" of the mean intensity image after spatial normalization, as in Figure 4.11. The intensity values of a structural magnetic resonance image, while obviously carrying anatomical information, are affected by factors such as scanner settings, the partial volume effect, and the shading artifact. It is unclear how much the averaged MR intensity value tells us about biological homology. Instead, the raw intensity values are processed to identify each voxel as gray matter or white matter, and used to measure how consistently the normalization matches tissue types. This can be done fully automatically and is the first method of quantifying variability.

Early quantitative methods of evaluating variability include overlaying contours traced along fundi of a given sulcus in several individuals and estimating a "zone of maximum variability", as done by Talairach and co-workers [TST$^+$67]. The idea behind this type of approach is to measure the amount of "dispersion" of a single structure. At the present time, such a structure is generally manually delineated from the images. Once that is done, the computation then comprises two steps: first, a

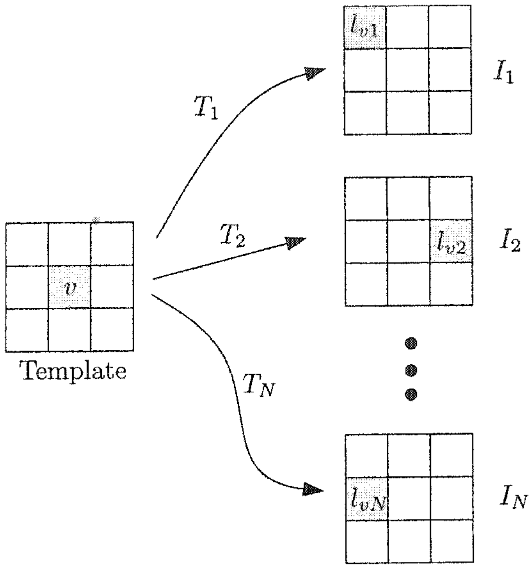


Figure 4.1: Induced labelling on the standard grid from spatially normalizing a population. Transformation T_i maps standard space voxel v to the shaded voxel (with label l_{vi}) on each image I_i. The resulting set of labels, $\{l_{v1}, \ldots, l_{vN}\}$, constitutes a sampling of the random label L_v of standard space voxel v. The voxel entropy, $H(L_v)$, is estimated from this sampling.

suitable "mean" structure is obtained from the input samples, and second, a measure of distance between the mean structure and each sample structure is computed. The dispersion is, like the standard deviation of ordinary statistics, related to the average of these distances.

4.1 Segmentation Variability

A *segmentation* of an image is an assignment (manual or automatic) of a class label to each voxel. Labels assigned to an input image that has been spatially normalized can be carried along with a spatial transformation to induce a segmentation on a voxel grid in the standard space. The labels can represent any relevant information. In this thesis, labels of tissue type (gray matter, white matter, or CSF) are used and also labels of sulci. However, the labels representing a functionally-defined region could equally well be used. Each voxel of the standard space grid is assigned a label by each input image. The goal is to measure the consistency of the set of labels assigned to a given standard space voxel.

Consider voxel v in the standard space grid. This voxel maps to a certain point in subject i, which has label l_{vi} (see Figure 4.1). A spatial normalization method that achieves its goal of matching homologous points of each input will result in identical labels across the subjects ($l_{v1} = l_{v2} = \ldots$) for each voxel v. Warfield et al. [WRH+01] suggest to consider the label of voxel v as a random variable L_v, of which the set $\{l_{vi}\}$ is a sampling. The entropy [CT91] of this distribution is

$$H(L_v) = -\sum_l p_l \log_2 p_l, \tag{4.1}$$

where p_l is the probability that L_v is assigned label l. The sum is taken over the labels l for which $p_l \neq 0$. The probabilities p_l are estimated as the frequencies observed after spatially normalizing a sample population as follows. Let N_l be the number of subjects for which the voxel label is l, i.e. the size of the set $\{i : l_{vi} = l\}$. Then the probability p_l is estimated as $p_l = N_l/N$. The entropy $H(L_v)$ measures the amount of uncertainty in the label L_v (in units of bits, when using base-2 logarithms), which is regarded as the anatomical variability at voxel v. The entropy satisfies $0 \leq H(L_v) \leq \log_2 n$, where n is the number of possible labels [CT91]. The lower bound $H(L_v) = 0$ is obtained when $p_l = 1$ for precisely one label l and is zero for the rest, while the upper bound $H(L_v) = \log_2 n$ is obtained when $p_l = 1/n$ for all l.

For each standard space voxel v, L_v is a distinct random variable. The joint entropy of random variables A and B satisfies $H(A, B) \leq H(A) + H(B)$, where the equality holds if, and only if, A and B are independent [CT91]. The joint entropy of $\{L_v : v \in V\}$, where V is the set of standard space voxels is thus bounded by the sum

$$H = \sum_v H(L_v). \tag{4.2}$$

The quantity H, termed the *total entropy*, is used as a measure of variability remaining after spatial normalization is applied.

4.1.1 Application: Evaluating Algorithm Design

Recall that the inner loop of ANIMAL (Algorithm 4) has three parameters that need to be specified: the similarity cost ratio (hereafter referred to as simply "similarity"), the weight, and the stiffness, summarized in Table 4.1. Collins and Evans use values of 0.5, 0.6, and 0.5, respectively [CE97], chosen empirically by visual inspection of the displacements and resampled images to judge registration quality. This section demonstrates that optimal parameter values can be selected using total entropy as a performance measure. In addition to selecting numerical values for user parameters, total entropy enables quantitative investigation of *all* design aspects of a registration

Table 4.1: Major Parameters of ANIMAL inner loop, Algorithm 4.

Symbol	Description
a_1	Similarity Cost Ratio
a_2	Weight
a_3	Stiffness

algorithm in the context of spatial normalization.

One may wonder whether it would be more straightforward to consider the entropy measure as a function of all N transformations and use total entropy as the objective function for registration. Minimizing total entropy would thus co-register all N images simultaneously. Such an approach, which is feasible for affine registration of small 2D images [MMV00], would be more direct than the method described here in which the registration optimiziation is done for each image (to the template) using fixed parameters selected using the entropy measure. However, given that the computational cost for (pairwise) 3D image registration using free-form deformations is already high, it is expected that attempting to simultaneously co-register a number of 3D images would not be feasible.

To investigate design choices of ANIMAL, 40 T_1-weighted images are selected arbitrarily from the ICBM data base [MTE+95]. All are segmented into white matter, gray matter, cerebral spinal fluid, and background classes [Kol96] with non-brain voxels removed [Smi02]. One of the images is selected to be the template and 10 other images are spatially normalized to that template. Some of the later experiments normalize the full set of 39 subjects (this will be explicitly indicated).

The total entropy is computed after affine registration, and after registration using various choices for weight, stiffness, and similarity. Table 4.2 shows the results for the classic choice of parameters (similarity=0.5, weight=0.6, stiffness=0.5) and the results with the same similarity and stiffness but using weight=1: changing just the one parameter improves the result. Alarmingly, however, it appears that the gradient magnitude fit degrades the result, so that aspect is investigated first.

Outer Loop

Several choices for the numerical parameters are used to normalize the test data set, and the entropy is computed after each of the four iterations of the outer loop. It is clear that the increase in entropy in the fourth level of the hierarchy is a general phenomenon. Figure 4.2 shows representative results for weight=1, stiffness=0.9 and

Table 4.2: Total entropy values (units of 10^6 bits) for 9-parameter affine registration and for registration with ANIMAL, after each iteration of the outer loop. Second column shows results for the "classic" set of parameters (similarity=0.5, weight=0.6, stiffness=0.5). Third column shows results using similarity=0.5, weight=1, stiffness=0.5. Note that changing just one parameter (weight) improves the performance.

Outer Level	Classic Animal	Weight=1
9-Parameter Affine	1.96	
8 mm Grid	1.74	1.71
4 mm Grid	1.72	1.71
2 mm Grid	1.67	1.66
2 mm (Gradient)	1.70	1.68

using a range of similarity values between 0.1 and 1.

Close examination of the algorithm reveals that the node thinning strategy (Section 3.7.2) is the culprit. For the three iterations of the outer loop that use intensity data, this heuristic retains nearly all the control mesh vertices lying in brain tissue, while skipping control mesh vertices located outside of the head. Figure 3.4 shows the blurred and thresholded data from a typical input when smoothed using FWHM=4 mm kernel.

In the gradient data iteration, however, only values on the scalp, ventricle, and superficial cortex edges are above the threshold as shown by Figure 4.3. Displacements are therefore estimated on very few control mesh vertices (about 1/3 of the number of vertices in the previous outer iteration, which uses the same control mesh), while all vertices participate in the smoothing of the displacement vectors, Step 3 of Algorithm 4. The effect is to smooth out the warp, degrading the data fit.

Omitting the node thinning heuristic for the 2 mm grid gradient data fit brings the total entropy down below the value obtained using the 2 mm grid intensity fit. Omitting the heuristic for the intensity fits does not change the results appreciably, so no node thinning is done for any of the following.

Data Term

It is noted in Section 3.2.2 that the normalized cross-correlation is an appropriate data term in the situation where the intensity relation between registered voxels is linear, whereas the correlation coefficient assumes an affine relation.

An affine function ($\mathbb{R} \rightarrow \mathbb{R}$) has two parameters, which can be solved for using

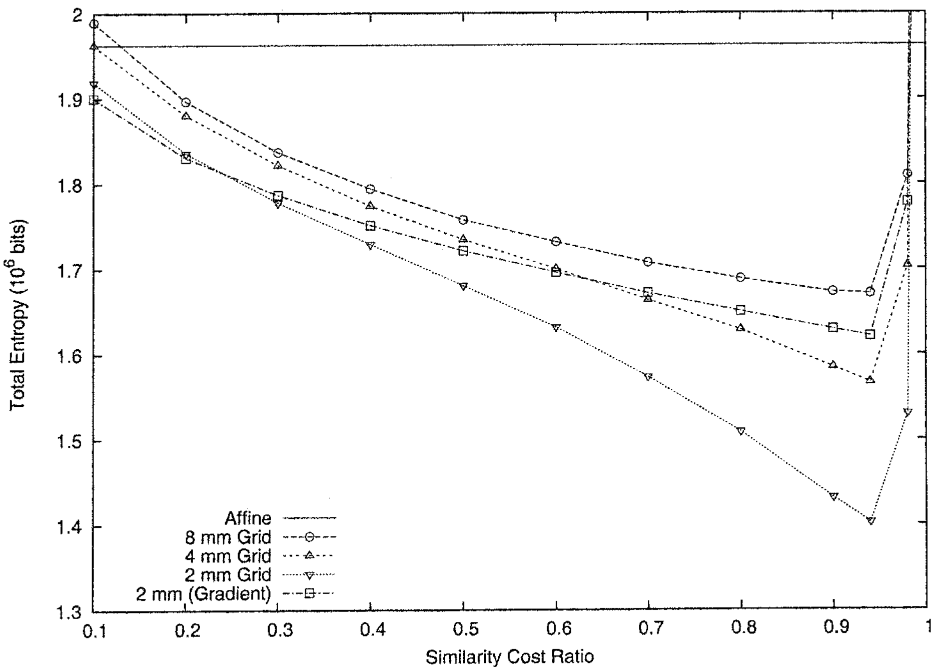
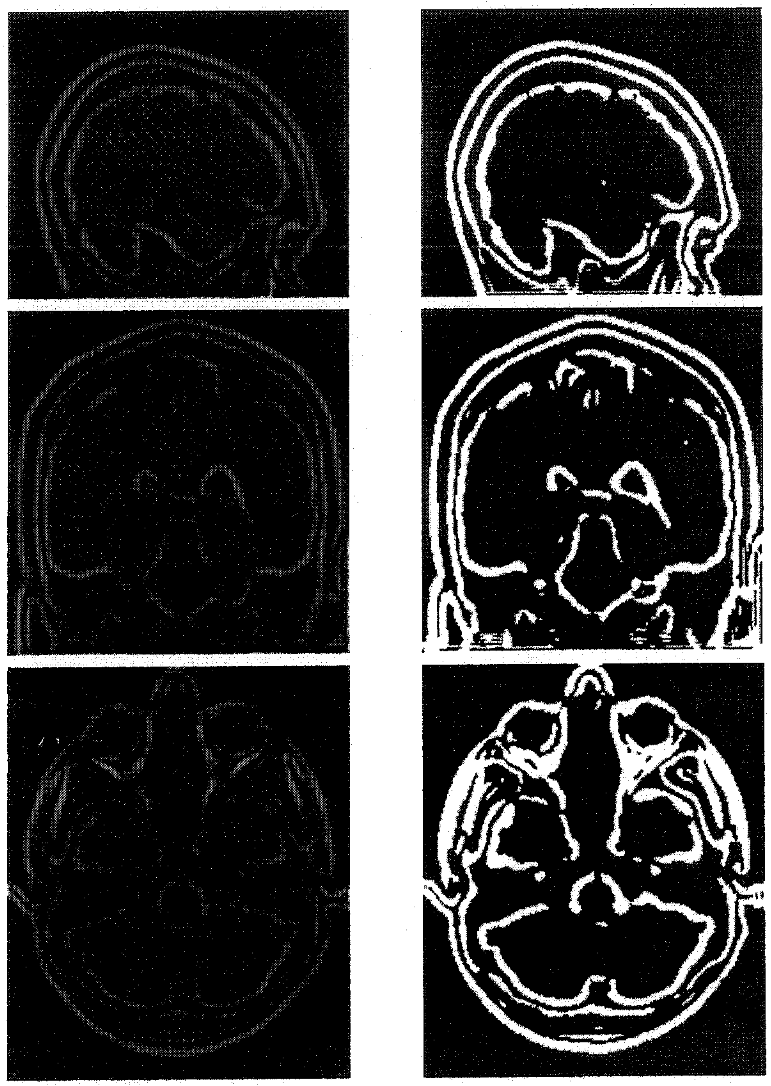


Figure 4.2: Residual anatomical variability as measured by total entropy, H, on a sample of 10 individuals after registration with ANIMAL. Plot shows results after each of the four iterations of the outer loop (weight=1.0, stiffness=0.9) along with the value for 9-parameter affine normalization, for reference. Note that the variability is reduced for each of the first three iterations of the outer loop, but *increases* on the fourth iteration.



Smoothed Gradient Magnitude Node Thinning Mask

Figure 4.3: Left column shows the gradient magnitude of an MR image, smoothed with a FWHM=4 mm Gaussian kernel, the input data for the fourth iteration of the ANIMAL outer loop. Right column shows the corresponding mask used for node thinning heuristic. The displacement is only estimated for a control mesh vertex that lies in the white region. The masking is clearly an inappropriate strategy for gradient data, as the set of control mesh vertices for which the displacement is estimated does not cover the brain. Compare with the mask for the third iteration of the outer loop, pictured in Figure 3.4.

two intensity values from each image. The parameters could be chosen, for example, so that the intensity of gray matter of the source image maps to the gray matter intensity of the target image and the intensity for white matter is similarly matched. In contrast, a linear function has just one parameter. Only one intensity point can be matched, so it may not be possible to accurately match both the gray and white intensities.

The ability to match two intensity values suggests that the correlation coefficient could be a better matching criterion than cross correlation, especially at the interface between two tissues. This hypothesis is tested by replacing the similarity function in ANIMAL by the correlation coefficient similarity function Φ_{CC} (Equation 3.13). Since ANIMAL uses a regional similarity measure (discussed in Section 3.2.2), the two parameters of the affine intensity mapping should be adequate: at the finest control mesh resolution (with control mesh vertices separated by 2 mm), the similarity is evaluated over a sphere of radius 3 mm (i.e. 1.5 times the control mesh spacing; see Section 3.7) which generally contains at most two tissue types.

The result of a number of tests using cross correlation with different parameter values is that the lowest entropy score is obtained using similarity=0.98, weight=0.8, stiffness=0.98. Similar testing using correlation coefficient yields similarity=0.3, weight=1.0, stiffness=1.0 as optimum. The entropy score in each case is approximately equal to 1.37×10^6 bits. However, the sensitivity to parameter variation about the optimal set is markedly different. Figures 4.4 and 4.5 compare the entropy scores as a function of similarity and of stiffness, respectively. Both plots show a much shallower curve with good performance (i.e. low entropy) over a broad range of parameter values, when using correlation coefficient. These plots present data from a single sample of 10 subjects. A different set of subjects might require slightly different parameter values to produce optimal results (this question is addressed below; see Figure 4.9). The behaviour of ANIMAL using correlation coefficient is much better in this regard than when cross correlation is used; the former produces good results even if the parameters are not precisely optimal. This behaviour is desirable, so correlation coefficient is used for subsequent experiments.

Regularization

Consider now the effect of the similarity cost ratio parameter (a_1) that controls the relative contributions of the data term, ϕ_D^v, and the displacement update penalty term, ψ, to the objective function. The objective function terms that involve the

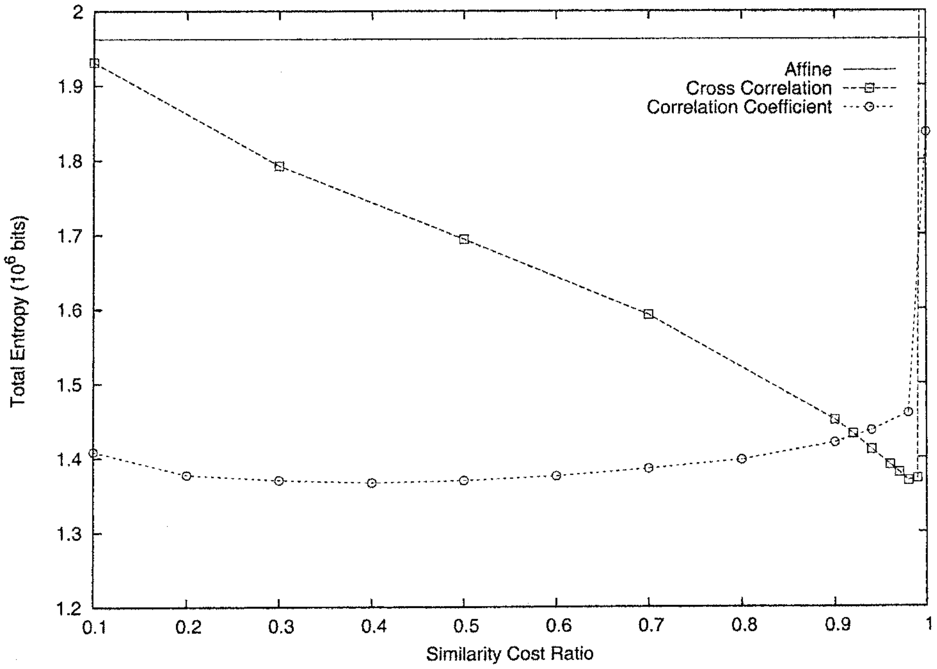


Figure 4.4: Comparison of residual variability as a function of similarity cost ratio for the cross correlation data term (using weight=0.8, stiffness=0.98) and the correlation coefficient data term (using weight=1, stiffness=1), along with the value for 9-parameter affine normalization for reference. Comparison made at 2 mm (intensity) level of ANIMAL outer loop. Though the minimum variability attained is about the same for each data term, the correlation coefficient achieves low variability over a wide range of values for similarity cost ratio so it is easier to choose a similarity value that produces good performance with that data term.

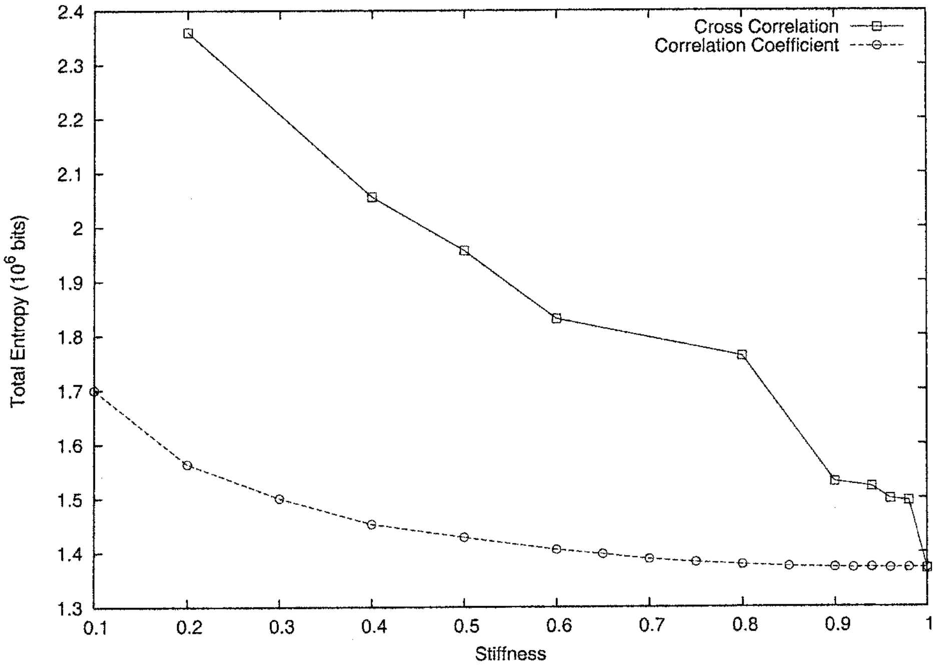


Figure 4.5: Comparison of residual variability as a function of stiffness for the cross correlation data term (using similarity=0.98, weight=1) and the correlation coefficient data term (using similarity=0.3, weight=1). Comparison made at 2 mm (intensity) level of ANIMAL outer loop. Note that the variability is much less sensitive to the value of stiffness when using the correlation coefficient data term.

control mesh vertex v are (from Line 1 of Algorithm 4)

$$a_1\phi_D^v(\Delta^v + \delta^v) + (1 - a_1)\psi(||\delta^v||). \tag{4.3}$$

Figure 4.4 shows that, when using correlation coefficient, a value in the range 0.2-0.6 provides good performance.

Figure 4.4 also shows that the entropy becomes large for similarity values near 1, even larger than obtained using the initial affine transformation. This phenomenon is observed using either of the two data terms. When the similarity parameter is set to 1 in Equation 4.3, the registration is driven only by the data term ϕ_D^v with no control on the size of the correction vector, $||\delta^v||$. The poor performance near similarity=1 indicates that the correction length penalty ψ plays an important role in controlling the optimization.

Transformations obtained with this ratio set to 1 (i.e. no contribution at all from ψ) contain much larger displacements, are much less smooth, and have more instances of folding (non-invertibility) than those obtained with similarity cost ratio < 1. An example of this is shown in Figure 4.6, where the only difference between the two registrations is the similarity setting. This confirms the importance of incorporating the regularization into the algorithm.

Smoothing

The smoothing step of ANIMAL (Line 3 of Algorithm 4) sets the deformation vector for control mesh vertex v to be

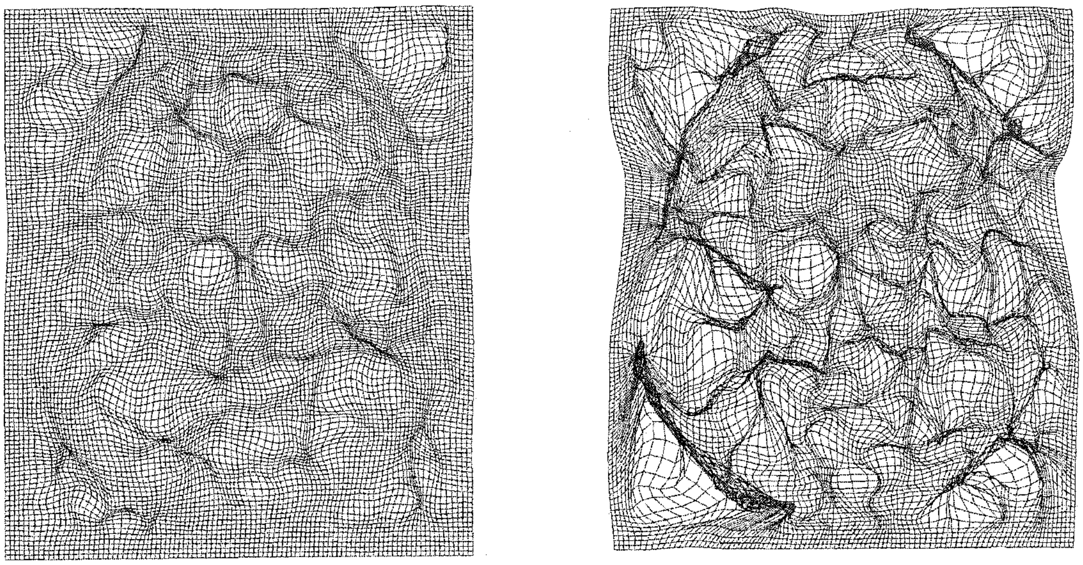
$$\Delta^v = a_3\overline{\Delta^v} + (1 - a_3)\Delta^v,$$

where $\overline{\Delta^v}$ is the mean displacement of 26-neighbours of v and a_3 is the stiffness parameter. Figure 4.5 indicates that very high values of stiffness parameter are best, so stiffness=1 is used.

Miscellaneous Parameters

The third major numerical parameter of ANIMAL is the weight value (a_2), used in Line 2 of Algorithm 4. Values in the range 0.8 to 1.4 (using similarity=0.3, stiffness=1) show little change in entropy values. The weight is set to 1 for all experiments.

The ANIMAL algorithm uses several other parameters not yet discussed. The



similarity=0.3

similarity=1.0

Figure 4.6: Control mesh vertices shown in their displaced positions, i.e. the grid at points $\{v+\Delta(v)\}$, for an axial slice through the source image. The mesh shown on the left is the result of registration using similarity=0.3 while the mesh on the right used similarity=1. All other parameters and the source and target images are identical for the two registrations. With similarity=1, the objective function has no contribution from the regularization term, resulting in a much rougher transformation with many more regions of noninjectivity.

only such parameter changed from the default setting is the number of iterations for which the inner loop is run. The termination condition for the inner loop of ANIMAL (Algorithm 4) is simply to iterate a fixed number of times. The iteration limits must be chosen large enough that the optimization reaches convergence.

An optimization is generally deemed to be converged when, to within a numerical tolerance, the gradient of the objective function is zero, the iterate updates become small, or both [DS96]. The objective function is not explicitly used in a two-step algorithm like ANIMAL (see Section 3.4.2), so only the change in iterate can be checked. Each iteration of the inner loop updates the deformation vector Δ^v for each control mesh vertex v. Let Δ_i^v be the deformation vector at iteration i of the inner loop, and consider the magnitude of deformation vector changes during iteration i, $\{||\Delta_i^v - \Delta_{i-1}^v|| : v \in V\}$, where V is the set of control mesh vertices. These quantities are the *total* change at each vertex, after both the matching and smoothing steps of ANIMAL. The inner loop can be deemed to have converged when the mean total change (averaged over all control mesh vertices) drops below a prespecified threshold.

The ANIMAL code, as distributed, uses 12 iterations for the 8 mm control grid, 8 iterations for the 4 mm control grid, and 5 iterations for the 2 mm control grid. Since it is unclear *a priori* how changing numerical parameters and other elements of ANIMAL (e.g. the data term) will affect the convergence of the inner loop, the iteration limits are raised to much larger values: 60, 60, and 20 for the 8 mm, 4 mm, and 2 mm control grids, respectively. This is done for all experiments in this thesis, including those already discussed. Figure 4.7 plots the mean total change as a function of iterations for three variants of ANIMAL. The upward spikes at iterations 60, 120, and 140 are due to starting the second, third, and fourth iterations of the outer loop, at which time both the control grid and the image data change (the latter is less smoothed). The graph indicates that the standard iteration limits would be inadequate when using the optimized parameter settings, whereas the limits in use suffice. For example, the mean deformation change is still large after five iterations for the 2 mm control grid, used in the third and fourth iterations of outer loop (starting at inner iterations 120 and 140 in Figure 4.7), but has settled down after 20 iterations.

Injectivity

As discussed in Section 2.5, mesh warpings are not guaranteed to be injective nor does the ANIMAL algorithm take any measures to enforce injectivity, though the regularization imposed by the penalty term ψ (see Figure 4.6) and the smoothing

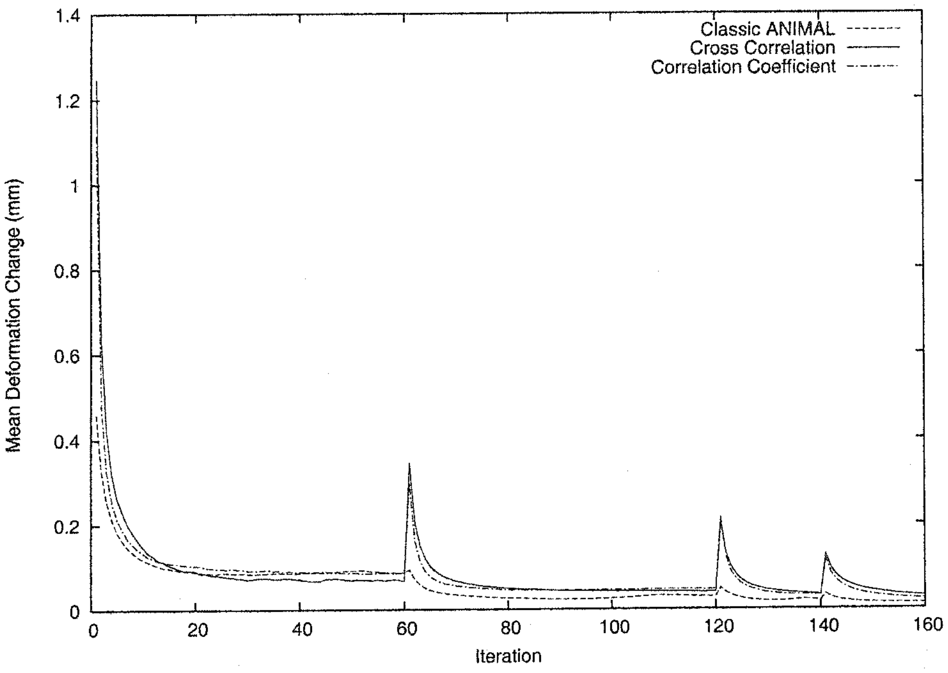


Figure 4.7: Mean magnitude of change in deformation vector, plotted for each iteration of the ANIMAL inner loop. Data for three variants of ANIMAL is shown. The curve labelled "classic ANIMAL" uses a cross correlation data term with similarity=0.5, weight=0.6, and stiffness=0.5. The other two curves use optimized parameter sets: cross correlation uses similarity=0.98, weight=0.5, and stiffness=0.98, while correlation coefficient uses similarity=0.3, weight=1, stiffness=1. The increases at iterations 60, 120, and 140 result from starting the second, third, and fourth iterations of the outer loop. Convergence of inner loop is demonstrated by the mean deformation change falling off before the next iteration of the outer loop. Note that the original limit of 5 iterations for the last two iterations of the outer loop (starting at inner iterations 120 and 140), while adequate for classic ANIMAL would be inadequate for the optimized versions of ANIMAL.

step both help.

Recall from Chapter 1 that the data to be analyzed in standard space is typically smoothed, meaning that a comparison at point x in the standard space is really not a point-to-point comparison with data precisely at $T_i(x)$ in each subject i, but a comparison of data in some region about $T_i(x)$. A certain amount of small-scale noninjectivity can therefore be tolerated in the transformations used for spatial normalization. To gauge whether noninjectivity is a serious problem in practice, the Jacobian determinant is estimated using finite differences at each control mesh vertex after registration. The full sample of 39 ICBM subjects is spatially normalized to the same template. The Jacobian determinant of the resulting transformation functions is negative (indicating noninjectivity) on a mean of 616/265401=0.2% of the control mesh vertices lying inside the scalp (standard deviation is 211 or 0.08%). These vertices tend to be concentrated around sulci, as shown in Figure 4.8. However, noninjectivity is relatively rare and thus will not be considered an impediment to using ANIMAL for spatial normalization.

Transferability

The experiments presented so far all use the same template and the same set of 10 test subjects. The question arises as to whether the parameters obtained are specific to the set of subjects used, or are generally applicable. To answer this, a second set of 10 subjects is registered using various similarity values with weight=1 and stiffness=1. The results (Figure 4.9) show the same shallow curves, indicating that the same similarity=0.3 value can be used for the new set. Chapter 8 investigates the variability obtained using other template images.

Since the parameters appear to be generally applicable, all registrations use similarity=0.3, weight=1, and stiffness=1, unless otherwise specified.

Summary of Experiments

Figure 4.10 provides a visual illustration of the reduced anatomical variability in the full set of 39 individuals, obtained using the tuned version of ANIMAL. The variability in the depth of many sulci is reduced, indicating that the sulci are better aligned.

The improved alignment is most readily apparent in the large regions of homogeneous tissue such as the white matter and ventricles. The coronal view (second row in Figure 4.10) clearly shows that the tuning is instrumental in aligning the white

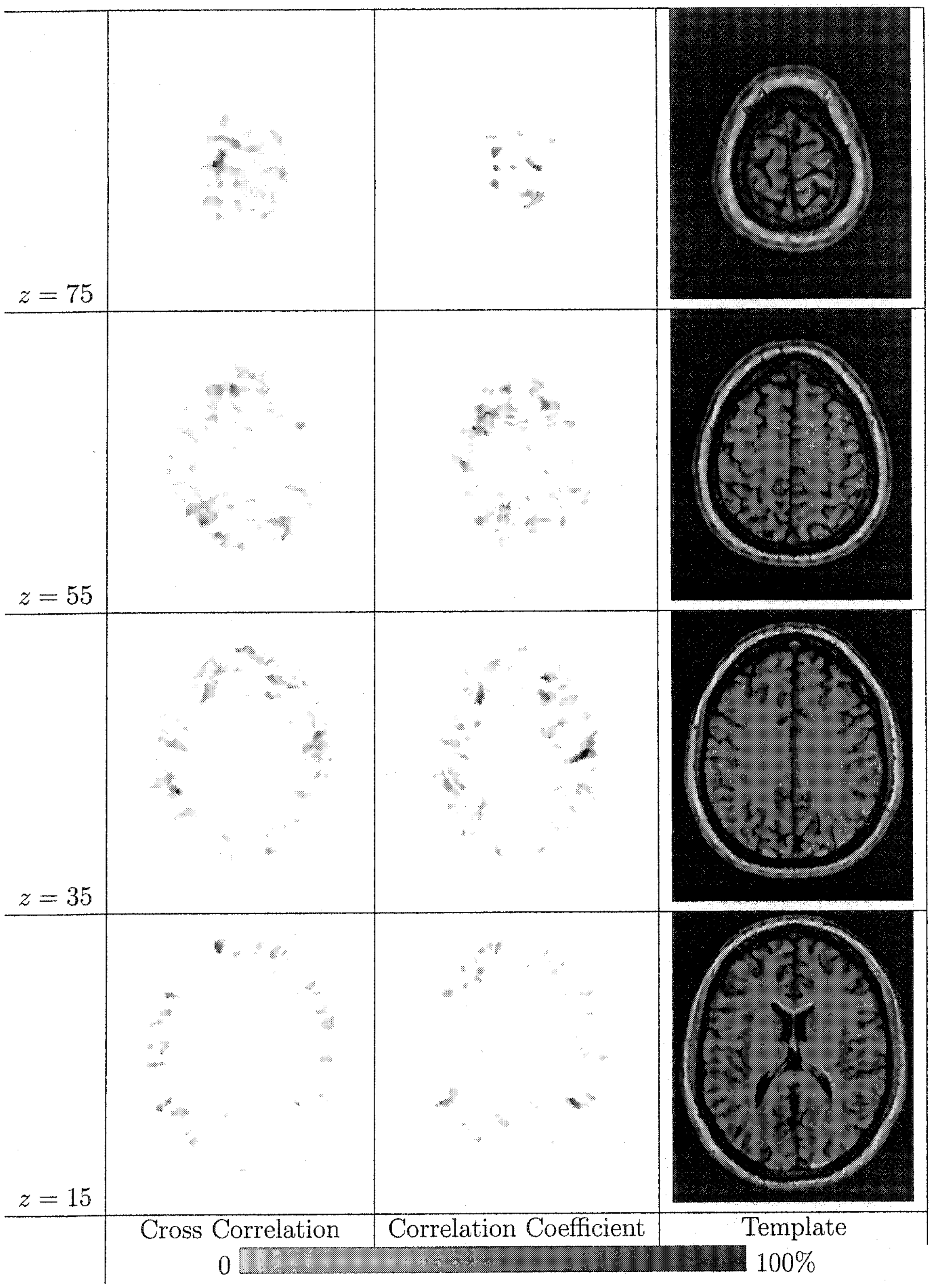


Figure 4.8: Maps showing frequency of negative Jacobian determinant at control mesh vertices, measured using transformations of 39 individuals. The largest frequency is approximately 25%. The left and middle columns show the results after using cross correlation and correlation coefficient, respectively. The right column shows the template, for reference. Overall, only 0.2% of the control mesh vertices in the brain region have negative Jacobian determinant, so the problem is negligible.

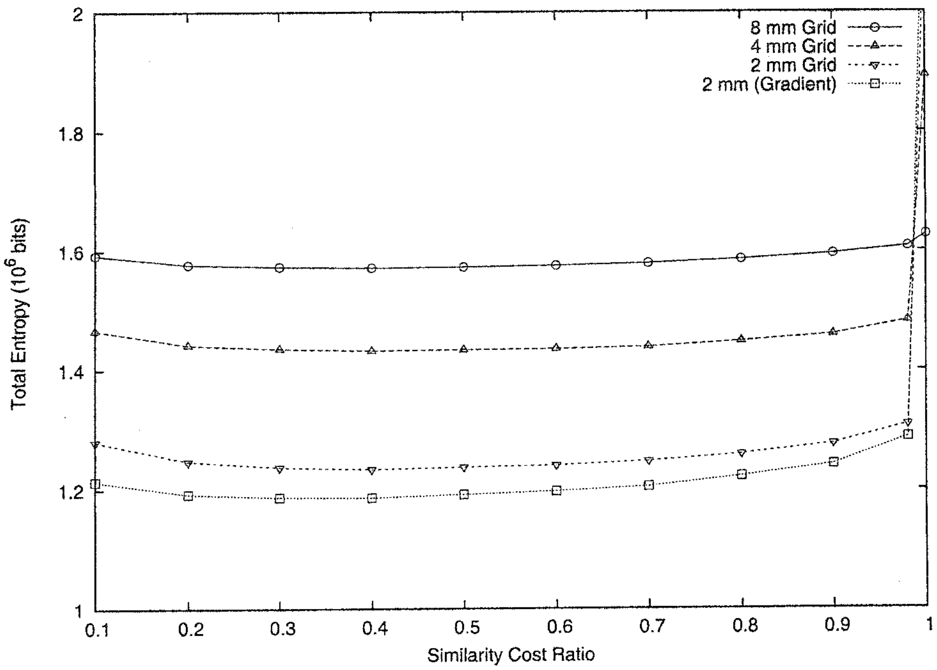


Figure 4.9: A second set of 10 subjects shows good performance in the same broad range of similarity values (using weight=1, stiffness=1) as exhibited in Figure 4.4. The parameter values are thus not specific to the particular data set.

matter of many gyri. The gray matter is also often well-matched and shows up with low variability, such as in the circled regions of Figure 4.10.

Boundaries between tissue types show high variability, some of which is due to misalignment and some of which is due to limitations of the classifier used to generate the segmentation. At a boundary, even a misalignment on the order of the voxel size is enough to change the labelling from an input image and hence the entropy value. However, some of the apparent variability is the result of imperfections in the tissue classification. Voxels in boundary regions frequent contain two (or more) tissue types, resulting in a signal intensity between the intensities of the two tissue types. Such partial volume voxels are more frequently misclassified. For example, the CSF is frequently misclassified as gray matter, leading to high entropy values in the CSF spaces of sulci.

A more sophisticated classifier could try to estimate the fraction of each type of tissue in a voxel [CHK91] and use that information to produce a more refined segmentation that includes classes for voxels containing more than one tissue. For example, a voxel that contains significant fractions of both gray matter and CSF could be labelled as a mixed "gray-CSF voxel". Enlarging the set of classes in this way

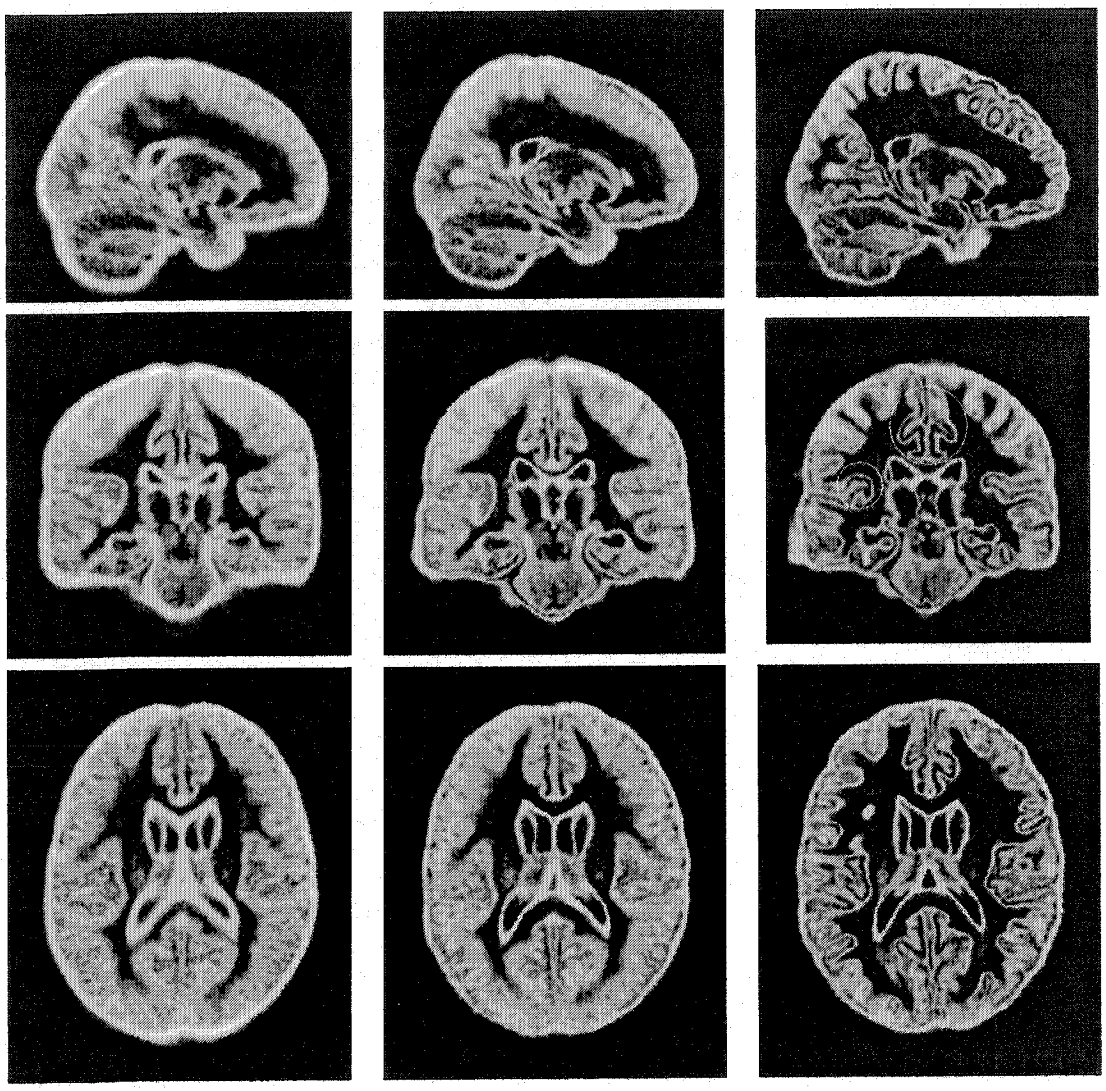


Figure 4.10: Entropy maps of 39 individuals after spatial normalization using 9-parameter affine registration, ANIMAL with the default parameters (and cross correlation data term), and ANIMAL using correlation coefficient data term and optimal parameter values (similarity=0.3, weight=1, stiffness=1). Voxels with more variability are brighter. Edges remain the most variable, both the white/gray interface and gray/CSF interface, producing an "outline" effect where the gray matter shows as less variable, bounded between two interfaces of high variability (examples are circled in the middle image of the third column). The third column shows a clear reduction in variability compared with the other two columns.

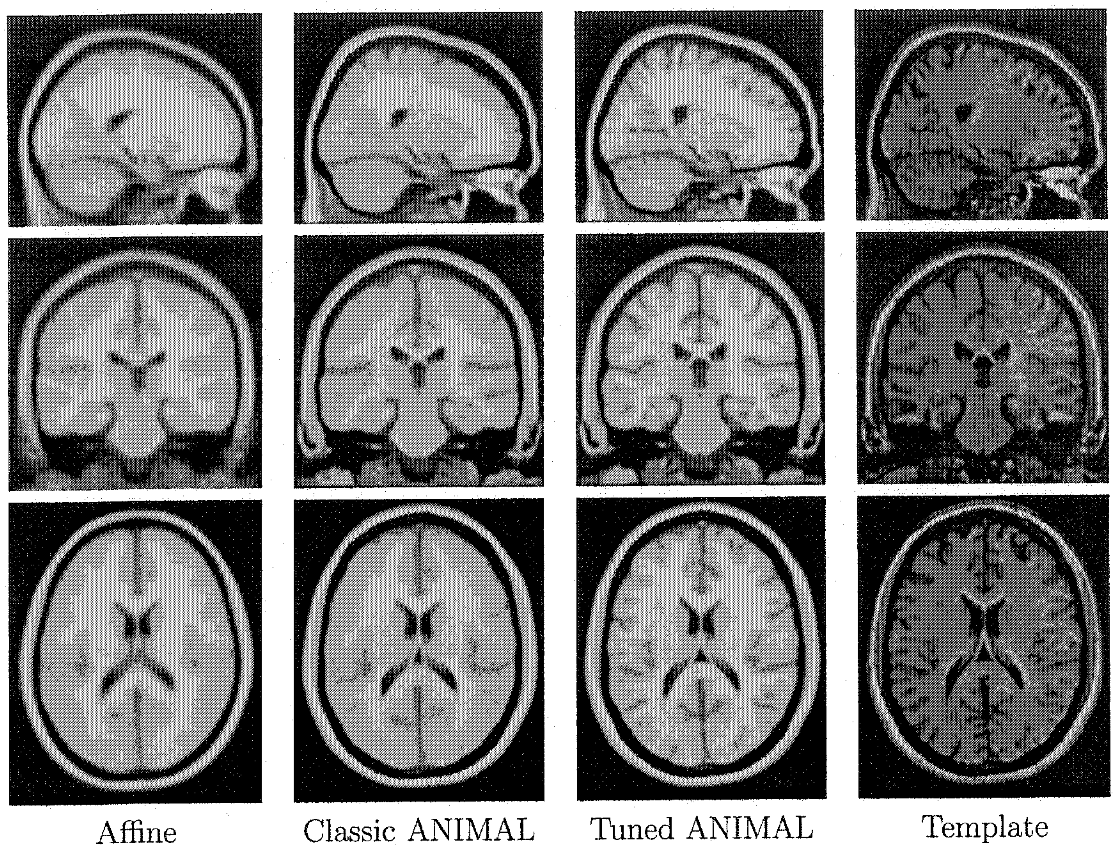


Figure 4.11: Intensity-averaged images of 39 individuals after spatial normalization using 9-parameter affine registration, ANIMAL with the default parameters (and cross correlation data term), and ANIMAL using correlation coefficient data term and optimal parameter values (similarity=0.3, weight=1, stiffness=1). The fourth column shows the template image. Note the sharpness of the third column compared to the first two columns, and the excellent matching of the third column with the template.

might improve the classifier performance, as it would allow labelling the boundary voxels as a mixed class rather than choosing one or the other pure classes. If so, this may remove some of the apparent variability at the cortical edges in Figure 4.10.

Figure 4.11 shows intensity-averaged images which become sharper with tuning, a qualitative display of the improvement in aligning fine detail.

4.1.2 Frontal Sulci

While total entropy of the tissue classification gives a good measure of overall matching, it is also of interest to know how well the spatial normalization succeeds in aligning specific structures, such as a particular sulcus. To quantify this, a human

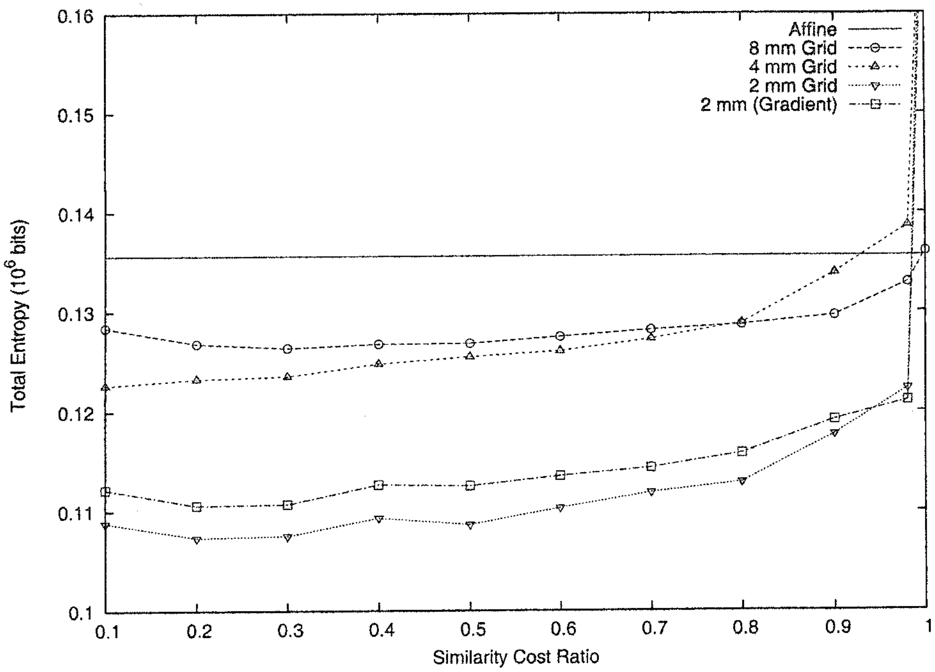


Figure 4.12: Residual variability of segmentation of frontal sulci (46 labels) as measured by total entropy, H, on a sample of 10 individuals after registration with ANIMAL. Plot shows results after each of the four iterations of the outer loop (weight=1, stiffness=1) along with the value for 9-parameter affine normalization, for reference. Note that similarity values in the range 0.2-0.6 produce good results, the same range as found using tissue class labelling shown in Figure 4.9. However, while the variability is reduced for each of the first three iterations of the outer loop, it *increases* on the fourth (gradient fit) iteration. The gradient fit will no longer be used.

anatomical expert manually identified 46 sulcal segments in the frontal lobes of the 40 ICBM images under study.

The total entropy of this set of labels (Figure 4.12) shows the same slowly-changing behaviour as a function of similarity cost ratio as obtained using the tissue labels (Figure 4.4). However, the fourth outer loop iteration, which uses a gradient fit, produces a larger total entropy than that of the third iteration (the 2 mm grid intensity fit). Since the gradient fit produces little overall reduction in variability (Figure 4.9), while increasing the variability for sulci, it will be omitted for the remainder of this thesis. As an added benefit, the running time of each registration is cut nearly in half.

For the experiments to follow, four sulci are used: central, superior and inferior precentral, and superior frontal; see Figure 4.13. Each sulcus is identified as the union of its constituent folds. The other labels are discarded, leaving just four labels, one

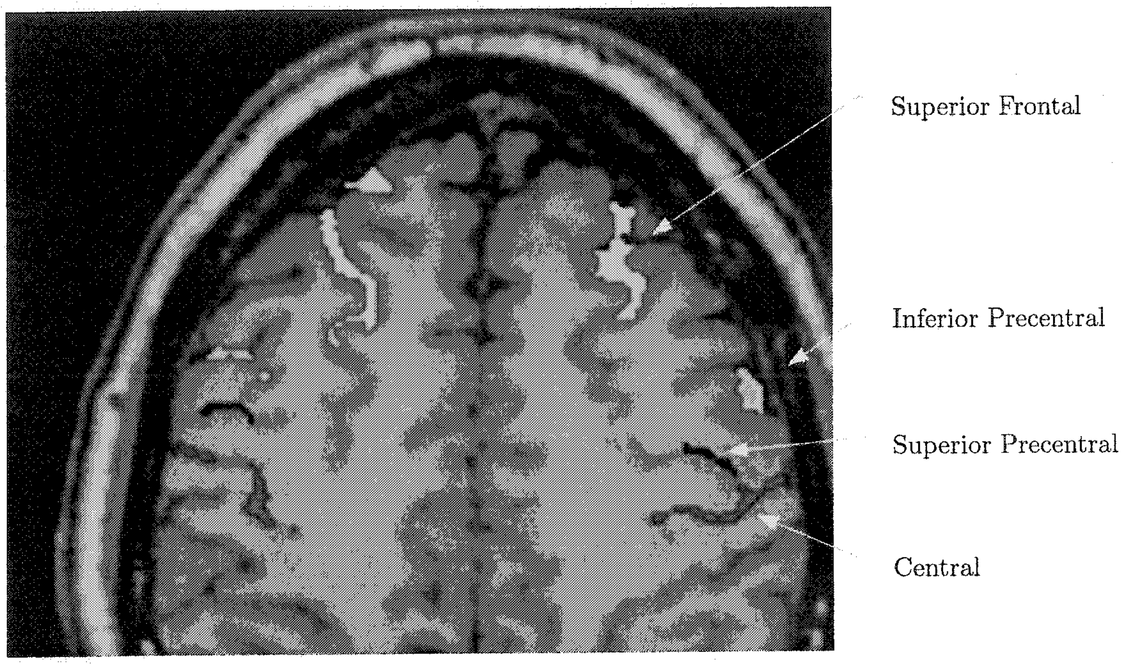


Figure 4.13: The frontal sulcus labels used to measure variability.

for each of the sulci. All 39 images (all except the template) are normalized and used in the subsequent analysis.

After spatially normalizing the inputs, the location of the central sulcus can be given probabilistically as follows. Each voxel v of a grid in the standardized space is assigned a value equal to the fraction of inputs for which the central sulcus overlaps v. This fraction is interpreted as the probability that v forms part of the central sulcus. The same procedure can be carried out for each of the other three sulci. Formally, for a random point set $X \subset \mathbb{R}^3$, define the *coverage function* p_X to be the probability that a point $x \in \mathbb{R}^3$ forms part of X, $p_X(x) = \mathsf{Pr}(x \in X)$ [SS94]. Spatial maps of the coverage function for anatomical features are also referred to as *Statistical Probability Anatomy Maps* (SPAMs) [CZPE01].

The coverage function for the central sulcus after a 9-parameter affine normalization and after nonaffine normalization using ANIMAL is shown in Figure 4.14. From the figure, it is evident that the nonaffine normalization produces a much tighter distribution in space (shown in the third column) than does the affine normalization (shown in the first column). It is also clear that the input central sulci are being matched to two distinct folds on the template. This is most evident for the left hemisphere on the $z = 34$ and $z = 44$ slices. In fact, all four sulci in left hemisphere are matched to more than one fold of the template. This also happens on the right hemisphere, though to a much lesser extent.

Variability of the central sulcus, as measured by total entropy, is smaller after nonlinear normalization, decreasing from 1.8×10^4 to 1.3×10^4 bits on the left hemisphere and from 1.6×10^4 to 1.1×10^4 bits on the right. In spite of this, the "width" of the left hemisphere distribution on the $z = 34$ image has not appreciably decreased. The decrease in total entropy, without any decrease in spatial width of the distribution, is understood with the help of a 1-dimensional example, illustrated in Figure 4.15. Consider the example as a linear cross-section through the distribution on the $z = 34$ plane. In this example there are two structure labels: the central sulcus has label 1 and other voxels have label 0. Suppose there are 4 input images, each of which has a single voxel labelled as central sulcus. The distribution on the left of Figure 4.15 has four voxels in which the probabilities are not all 0 or 1. Each of these four voxels has $p_0 = 3/4$ and $p_1 = 1/4$, giving entropy (Equation 4.1) of

$$-\sum_l p_l \log_2 p_l = -\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4} \approx 0.81$$

at each such voxel, for a total entropy $H = \sum_v H(L_v) \approx 3.2$. For the distribution shown on the right, there are two voxels with nonzero entropy. Each voxel has $p_0 = p_1 = 1/2$, giving a voxel entropy of

$$-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1,$$

for a total entropy of 2. This lower entropy is obtained no matter what is the separation between the two non uniformly-labelled voxels.

The phenomenon illustrated by this simple example is also occurring in the spatial normalization data shown in Figure 4.14. Consider the left hemisphere sulci on the $z = 34$ slice. The sulci after affine spatial normalization are located in a small region of space, but there is not much overlap at any particular voxel: the larger coverage function values are in the range 20-25%. After a nonaffine spatial normalization, shown in the third column of the figure, the coverage function values are in the range 30-35%. The coverage function value at a voxel is precisely the probability p_l, for l="central sulcus", that goes into computing the entropy $H(L_v)$ in Equation 4.1. Thus the affine result of Figure 4.14, with many voxels of low probability, is analogous to the left distribution of Figure 4.15. The nonaffine result of Figure 4.14, with fewer voxels covered, but with higher probability values is analogous to the right distribution of Figure 4.15. The entropy after nonaffine spatial normalization

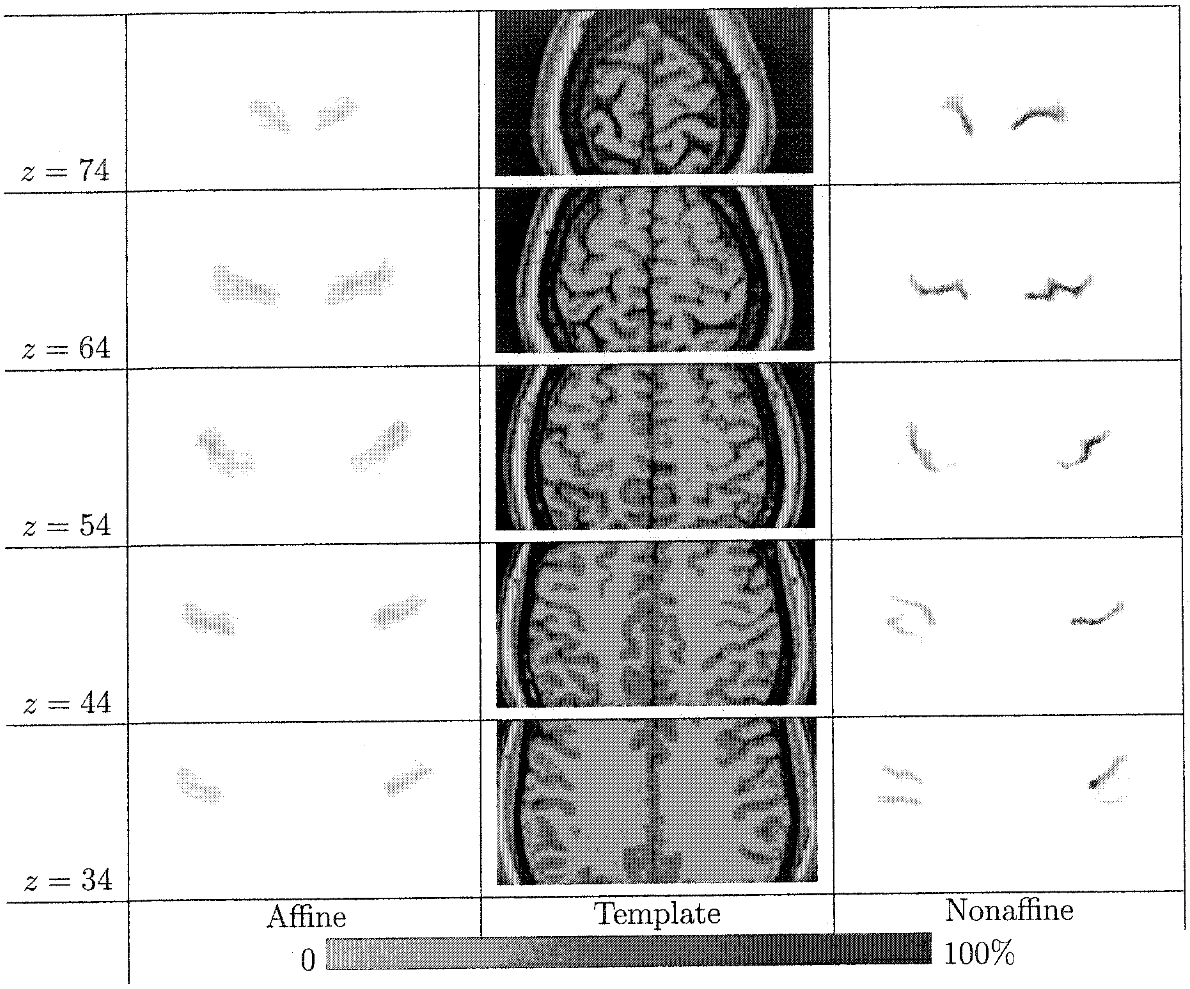


Figure 4.14: Coverage function for the central sulcus of 39 individuals, after 9-parameter affine spatial normalization (left), and nonaffine normalization using ANIMAL (right). The template images are shown in the centre column, for reference. Note that the maximum probability values are higher when using nonaffine normalization compared to affine normalization, but that the input sulci are being matched to two folds of the template, e.g. on the $z = 34$ slice.

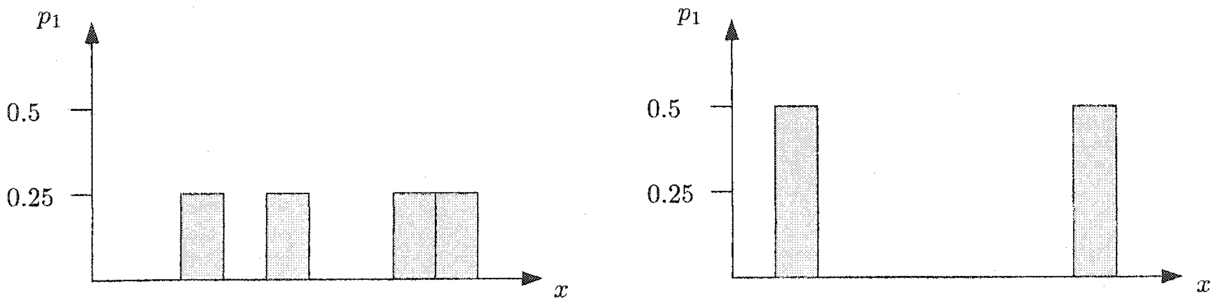


Figure 4.15: The total entropy is lower for the distribution on the right compared to the distribution on the left, even though the spread along the x-axis is larger for the distribution on the right.

is 1.3×10^4 bits, lower than that obtained after affine spatial normalization (1.8×10^4 bits). The entropy after nonaffine spatial normalization would continue to be lower, no matter how widely separated are the two concentrations of sulci appearing in the left hemisphere sulci of the $z = 34$ slice.

In summary, the voxels are more consistently labelled, but are not better localized in space. The total entropy is clearly not sensitive to this spatial variability. Thus, a different measure of variability is required to capture the spatial distribution of the structures in question.

4.2 Structure Variability

In univariate statistics, the variability of a quantity can be described using some measure of dispersion about a central value. Two common such measures are the standard deviation about the mean, and the mean absolute deviation,

$$\frac{1}{N}\sum_{i=1}^{N}|x - m|,$$

about the median m [GGD91, Chapter 8]. A similar approach is considered here for labelled brain structures, specifically for the frontal sulci described in Section 4.1.2.

Each structure under consideration is a closed and bounded point set of $\mathbb{R}^3$ (thus a compact point set), with nonzero volume. For example, a 1 mm^3 cubic voxel with centre at $(5, 7, 2)$ is considered as the point set $\{(x, y, z) : 4.5 \leq x \leq 5.5; 6.5 \leq y \leq 7.5; 1.5 \leq z \leq 2.5\}$. For the present set of experiments, the structures under study are finite unions of such voxels.

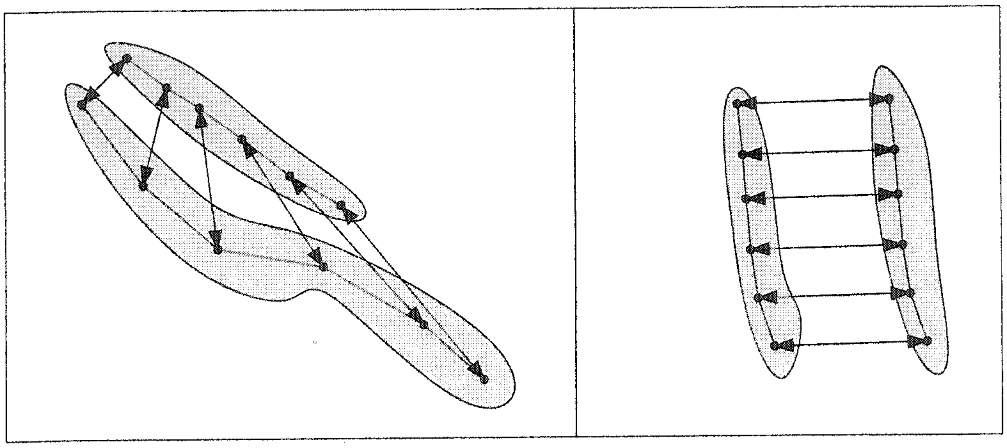


Figure 4.16: Structure distance defined using matched landmarks can be misleading. Though the pair of structures in the left panel appears closer together than the pair of structures in the right panel, the mean landmark distance is the same in both panels.

Defining a measure of dispersion for compact point sets requires selection of a central structure, and a measure of distance between structures. These two topics are addressed in the reverse order.

4.2.1 Distance Measure

In the brain imaging literature, a method commonly used to measure a distance between two structures involves identifying and matching one or more points on each structure. For example, a sulcus represented by a curve (e.g. the fundus) can be uniformly sampled with the points numbered sequentially along the curve. The i^{th} point on each curve is deemed to be corresponding [KLT+00] and the structure distance is then derived from the distances between corresponding points, e.g. by taking the mean. The analogous procedure can be carried out for a sulcus represented by a medial surface patch, in this case sampling on a two-dimensional grid [TST96].

Such an approach is simply one choice as to how the structures should be matched. In particular, the assumption is that the endpoints of one sulcus should match the endpoints of the other, even if the sulci are of very different lengths. These a-priori landmark matchings can assign a large distance to situations where (considered as continuous sets of points), the sulci could be considered to be close, see Figure 4.16.

Moreover, this kind of strategy breaks down when the structure in question is more complicated than a 1D curve or 2D surface patch. For example, sulci obtained by manual delineation of the CSF spaces will generally be like a "thick" surface patch, with varying thickness; see Figure 4.13, for example. Or consider measuring the distance between structures that consist of all gray matter in a brain. Obtaining homologous landmarks throughout the gray matter is essentially equivalent to registering the two cortices, but it is the registration itself that is to be quantified using the distance measure.

These difficulties are avoided by defining a measure based on the distance to the nearest point of the other structure rather than a prespecified point, a strategy used by e.g. [WGW+98, CLGE98]. This strategy of assigning distance removes the necessity of identifying and matching landmark points in the structures. In addition it can be usefully applied to complex structures like the gray matter, as will be shown in Chapter 8.

A suitable measure of distance between compact point sets is obtained by a modification of the well-known Hausdorff distance [Ruc96]. Let $A \subset \mathbb{R}^d$ be a compact point set and define the *distance transform* of A, denoted d_A, by

$$d_A : x \in \mathbb{R}^d \mapsto \inf_{a \in A} ||x - a||.$$

The distance transform is zero for all $x \in A$, and positive for $x \notin A$ (since A is closed). Consider the set of points at distance $\leq r$ from A, denoted

$$A_r = \{x \in \mathbb{R}^d : d_A(x) \leq r\}. \tag{4.4}$$

This set contains A as well as points near A. Given a second compact point set B, a measure of the nearness of A to B is obtained by considering the smallest r such that $A \subset B_r$. This value of r is given by $r = \rho(A, B)$ where ρ is defined as

$$\rho(A, B) = \sup_{a \in A} d_B(a). \tag{4.5}$$

This measures the closeness of A to B: all points of A are within distance $\rho(A, B)$ of B. This measure is not symmetric, as illustrated in Figure 4.17. The *Hausdorff distance* is defined to be the larger of the two asymmetric measures,

$$d_H(A, B) = \max\{\rho(A, B), \rho(B, A)\}. \tag{4.6}$$

Though it has the attractive property of being a metric on the space of compact sets, the Hausdorff distance is sensitive to outliers in the following sense [VH99]. Suppose

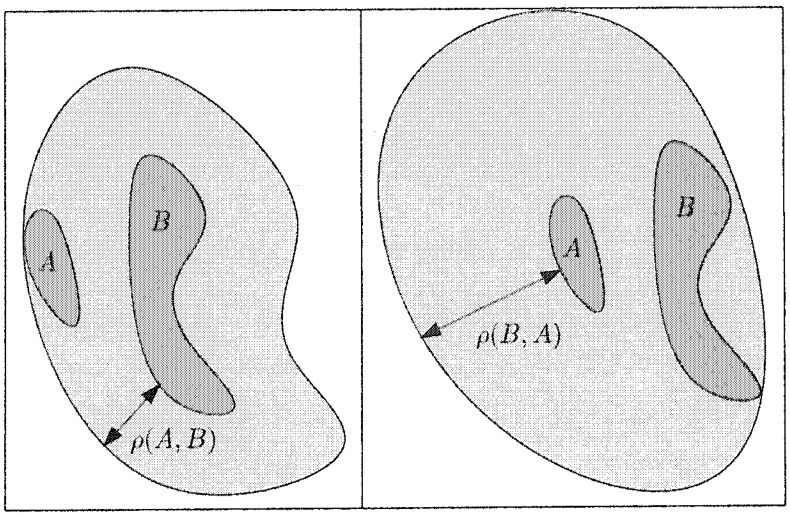


Figure 4.17: Distance $\rho(A, B)$ is the largest $d_B(a)$ for $a \in A$. The light grey area on the left is the set $\{x : d_B(x) \leq \rho(A, B)\}$ and it completely contains set A. On the right, the roles of A and B are reversed, illustrating that $\rho(B, A) \neq \rho(A, B)$.

A, B are given point sets, but due to noise only $A' \equiv A \cup P$ and B are observed, where P is some perturbation. For example, P may be some misclassified voxels. From Equation 4.5, $\rho(A', B) = \max\{\rho(A, B), \rho(P, B)\}$. Set P may be arbitrarily distant from B, hence $\rho(A', B)$ may be arbitrarily larger than $\rho(A, B)$, even though the perturbation P may be arbitrarily small in size (e.g. a single voxel). This defect can be mitigated, though not eliminated, by modifying the computation of $\rho(A, B)$, replacing the supremum in Equation 4.5 by a "quantile" value defined as follows.

For $q \in [0, 1]$, the q-quantile of a distribution is defined as a value x_q for which fraction q of the values are $\leq x_q$ and fraction $(1 - q)$ are $\geq x_q$. The median of a set is the $\frac{1}{2}$-quantile. Define the q-quantile distance function

$$\rho_q(A, B) = q\text{-Quantile}\{d_B(a) : a \in A\}.$$

This measures the nearness to B, not of set A as is the case for $\rho(A, B)$, but of the closest fraction q of the points of A. For example, set q=90%; then $r = \rho_q(A, B)$ is the distance such that 90% of A lies in B_r. Finally, define the q-trimmed Hausdorff distance (also known as the partial Hausdorff distance [Ruc96]) as

$$d_q(A, B) = \max\{\rho_q(A, B), \rho_q(B, A)\}. \tag{4.7}$$

In the experiments that follow, q-trimmed Hausdorff distance with q=90% is selected as the distance measure between sets. The reason for using this distance

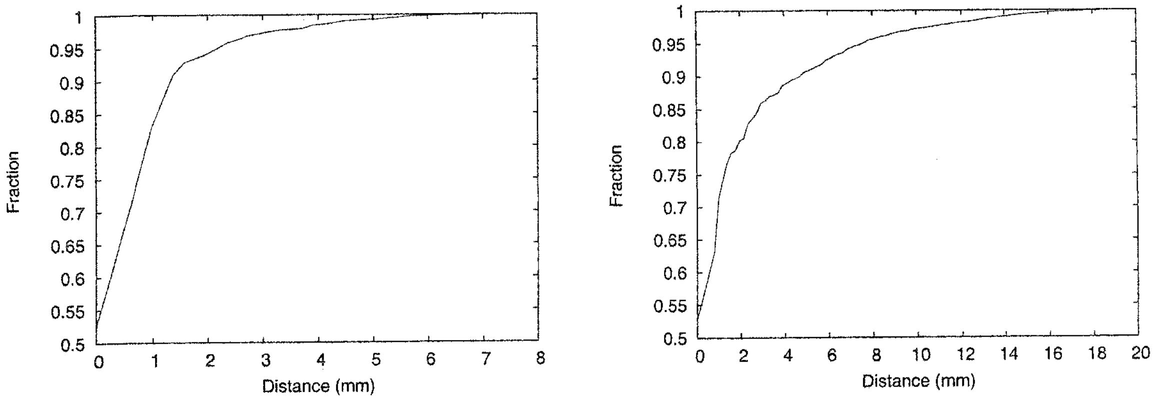


Figure 4.18: Cumulative distribution of distance sets $\{d_B(a) : a \in A\}$ and $\{d_A(b) : b \in B\}$ for typical pair of sulci A and B. The Hausdorff distances is the supremum of the two sets. The long tail is avoided by using the 90% trimmed distances.

rather than the Hausdorff distance is that the 90%-trimmed Hausdorff distance gives a more representative distance between points of the two sets. Figure 4.18 shows the distribution of distances $\{d_B(a) : a \in A\}$ and $\{d_A(b) : b \in B\}$ for a typical pair of input sulci, A and B. Using the 90% cutoff gives a representative distance, avoiding the long tail in the distribution.

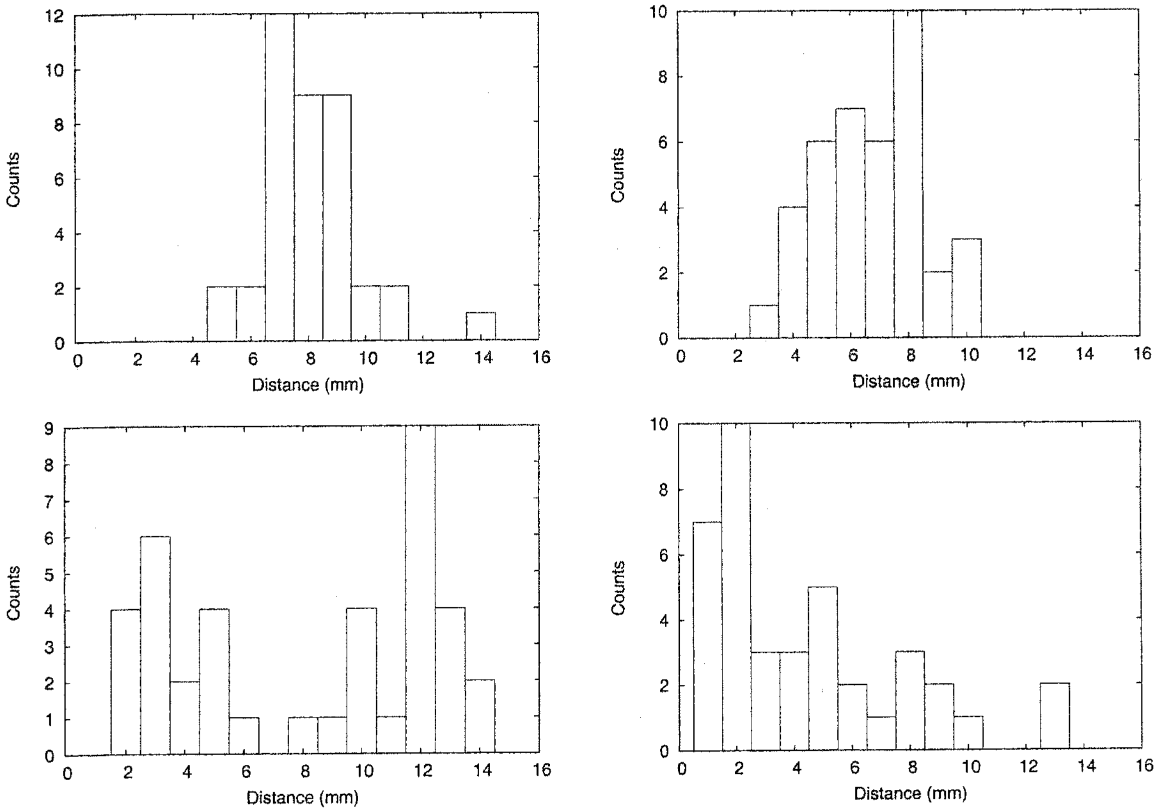
The implementation of this distance measure for experimental work presented in this thesis uses a discrete approximation of the structure (one point at the centre of each voxel) as well as a discrete approximation of the distance transform functions d_A, d_B [Bor84].

4.2.2 Dispersion about the Template

In some cases, the template image will have the same structures identified as do the population images. If so, the dispersion can be measured using the template structure as the central structure.

This is the case for the experiments presented so far. The template is one of the 40 images for which the four sulci are identified. For each of the sulci, 39 distances are computed: the distance between the template structure and one of the input structures. The effect of nonaffine spatial normalization is measured by computing the set of distances after 9-parameter affine registration, and again after running ANIMAL on the inputs.

For the central sulcus, the coverage function map in Figure 4.14 suggests that the distribution of distances after nonaffine spatial normalization is likely to be bimodal:



Left Central Sulcus

Right Central Sulcus

Figure 4.19: Distributions of the 90%-trimmed Hausdorff distance to the template central sulcus for 39 subjects. The distances are measured after a 9-parameter affine spatial normalization (top row) and after a nonaffine spatial normalization (bottom row). Note that the affine distributions are unimodal while the nonaffine are bimodal distributions.

some inputs are very near the template central sulcus while others are not, at least for the left hemisphere. The histograms of the 39 distances before and after nonaffine normalization, displayed in Figure 4.19, show clearly the bimodal distribution of distances for the left hemisphere after registration. In the right hemisphere, the distribution is shifted towards smaller values, though some values do become larger after nonaffine normalization.

Since the distribution is not necessarily unimodal, and may have a long tail (as in the lower right plot of Figure 4.19), the median value ($\frac{1}{2}$-quantile) is chosen to represent the average. In the plots to follow, the median is shown along with $\frac{1}{4}$-quantile and $\frac{3}{4}$-quantile to indicate the range of distance values. The median is quoted as the measure of dispersion.

Figure 4.20 shows the dispersion values after 9-parameter affine normalization

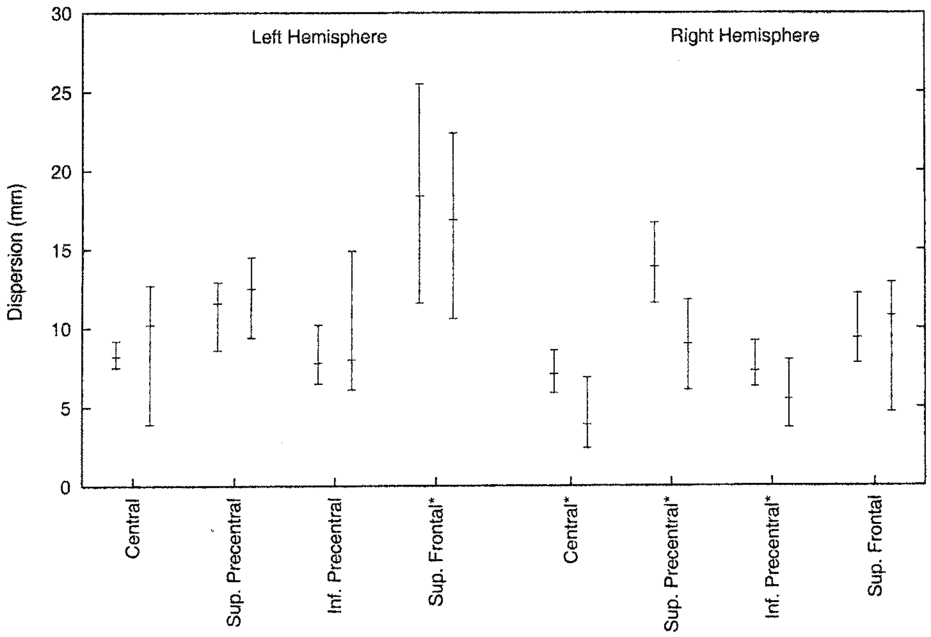


Figure 4.20: For each sulcus, the dispersion about the template sulcus is displayed after 9-parameter affine normalization (left bar of each pair) and after nonaffine normalization (right bar of each pair). The central mark indicates the dispersion value, which is the median of the set of distances from the template, while the error bars display the 25% and 75% quantile values, computed with 39 subjects. The sulci for which the medians obtained with affine normalization and with nonaffine normalization differ significantly (sign test at $p \leq 0.05$) are marked by an asterisk.

and after nonaffine normalization using ANIMAL. The nonparametric paired-sample sign test [Fre92] is used to test for significant change in the median value before and after nonaffine normalization. Of the 8 sulci, four show significant decrease after nonaffine registration while the remaining four show increases that are not statistically significant.

4.2.3 Mean of a Random Closed Set

The template structure may not be available. Or the template, if an arbitrarily chosen individual, may not be representative of the population. In such cases, a different notion of "centre" about which to measure dispersion is required. As discussed above, the dispersion for a real-valued distribution is measured about some notion of average (mean, median, etc). Similarly, consider the structure to be a random variable, X, whose value is a closed point set in $\mathbb{R}^3$. View the set of input structures as a sampling of X, from which a sample mean is computed to be used as the centre for dispersion measurements.

There is a considerable body of work on the theory of random closed sets, with several definitions for the mean of a random set, e.g. [SS94, Ser82, Mat75]. Some of the definitions are suitable only for convex or for star-shaped sets. The sulci considered here are not of this form, so such definitions of mean are not considered.

For a point sets $A \subset \mathbb{R}^3$, use $|A|$ to denote the volume. Define the *set difference* $A \setminus B$ as $A \setminus B = \{a \in A : a \notin B\}$ and the *symmetric set difference* $A - B$ as $A - B = (A \setminus B) \cup (B \setminus A)$. As previously discussed, the input structures are given as closed and bounded (hence compact) point sets of $\mathbb{R}^3$ that have non-zero volume. Let $\mathcal{K}$ denote the set of closed, bounded subsets of $\mathbb{R}^3$ that have non-zero volume, i.e. $\mathcal{K}$ is the set of compact set of $\mathbb{R}^3$ with non-zero volume.

Vorob'ev Mean

Let X be a random compact set of $\mathbb{R}^3$ with finite, nonzero expected volume, $\mathsf{E}|X|$. Recall that the value of the coverage function $p_X(x)$ is the probability that $x \in X$. For any t, define the set of points for which the coverage probability is at least t as $V_t = \{x \in \mathbb{R}^3 : p_X(x) \geq t\}$. It is easy to see that $t < u$ implies that $V_u \subset V_t$. Vorob'ev suggests to choose t such that the volume of V_t most closely matches the expected volume of X. Thus t is chosen to be the largest value for which $|V_t| \geq \mathsf{E}|X|$. The Vorob'ev mean $\mathsf{E}X$ is then defined to be V_t. This mean is the best-fitting set of volume $\mathsf{E}|X|$ as measured using symmetric set difference, $\mathsf{E}|X - \mathsf{E}X| \leq \mathsf{E}|X - B|$ for all B with volume $\mathsf{E}|X|$ [SS94, p.113].

For small structures with little overlap (i.e. low values in the coverage function), a drawback of this definition of mean is that the resulting mean structure often doesn't look like an individual input structure. Figure 4.21 shows three input structures that are roughly the same size and shape, lying roughly parallel to one another. Intuitively, one expects a "mean" to look something like one of the inputs and lie roughly in the middle. However, each of the non-white voxels for this input have coverage function value 1/3. The Vorob'ev threshold t cannot be larger than 1/3, otherwise V_t is empty. So $t = 1/3$ and the Vorob'ev mean is the union of the three inputs. This phenomenon happens in real-world data, too: the Vorob'ev mean for the left central sulcus after nonaffine normalization includes voxels from both the folds that appear on the $z = 34$ slice of Figure 4.14.

The Vorob'ev mean is more reasonable when the inputs are larger and more likely to overlap. In Chapter 8 the Vorob'ev mean of gray matter is used to select

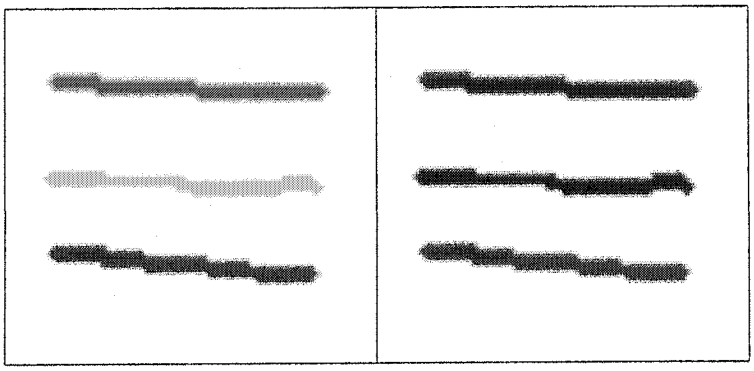


Figure 4.21: Three input structures are displayed in the left panel. The right panel shows the resulting Vorob'ev mean, which is not a natural result. A structure similar to the middle input is a more appealing mean.

a template. However, for smaller structures like sulci a different notion of mean is sought.

Fréchet Mean

Suppose that a metric function ρ is given on the space of structures $\mathcal{K}$ (the Hausdorff distance is a suitable metric). Then X is considered as a random element of the metric space $(\mathcal{K}, \rho)$. The following definition of a mean of a random variable that takes values in a metric space was introduced by Fréchet in 1948.

Let X be a random variable in the metric space $(\mathcal{K}, \rho)$ that satisfies $\mathsf{E}\rho^2(X, K) < \infty$ for all $K \in \mathcal{K}$. Any $M \in \mathcal{K}$ that satisfies

$$\mathsf{E}\rho^2(X, M) = \inf_{K \in \mathcal{K}} \mathsf{E}\rho^2(X, K) \tag{4.8}$$

is a mean value element of X [SS94]. The set of all mean value elements is called the Fréchet mean. For real numbers (metric space $(\mathbb{R}, d)$, where $d(x, y) = |x - y|$), the Fréchet mean reduces to the standard definition.

This mean has a natural appeal, but the big drawback to using it is that one must perform an optimization over the space $\mathcal{K}$. The next definition of mean can be seen as a modification of Fréchet's.

Distance Mean

Baddeley and Molchanov [BM98] suggest a method for computing a mean image by averaging distance transform functions. Let X be a random element of the metric space $(\mathcal{K}, d_H)$, where d_H is the Hausdorff distance defined in Equation 4.6. The

Hausdorff distance satisfies the identity

$$d_H(A, B) = \sup_{x \in \mathbb{R}^3} |d_A(x) - d_B(x)|, \tag{4.9}$$

which is proved, e.g. in [Bad92]. Using this, the optimization over $\mathcal{K}$ in Fréchet's mean (Equation 4.8) can be replaced by an optimization over the space of distance functions (functions $\mathbb{R}^3 \to \mathbb{R}$) with the uniform metric

$$\rho_\infty(f, g) = \sup_{x \in \mathbb{R}^3} |f(x) - g(x)| \tag{4.10}$$

as follows. Using Equations 4.9 and 4.10, write $d_H(A, B) = \rho_\infty(d_A, d_B)$, so the right hand side of Equation 4.8 (with $\rho = d_H$) can be written as

$$\inf_{K \in \mathcal{K}} \mathsf{E} d_H^2(X, K) = \inf_{K \in \mathcal{K}} \mathsf{E} \rho_\infty^2(d_X, d_K). \tag{4.11}$$

This reformulation confers no advantage, as the optimization of Equation 4.11 is no easier than the original optimization in Equation 4.8. The key is then to change the metric used on the distance functions which, of course, changes the metric space in which X lies and therefore changes the mean computed. Baddeley and Molchanov replace the uniform metric on distance functions by the L^2 metric,

$$\rho_2(f, g) = \left(\int_{\mathbb{R}^3} (f(x) - g(x))^2 dx \right)^{1/2}.$$

The advantage of this change is that the infimum over the space of distance functions has a unique solution, namely the expectation value of the distance function for X, $\mathsf{E} d_X$ [BM98]. When given a set of samples of X, $\{X_1, \ldots, X_N\}$, $\mathsf{E} d_X$ can be estimated as the sample mean,

$$\overline{d}(x) = \frac{1}{N} \sum_{i=1}^{N} d_{X_i}(x),$$

where d_{X_i} is the distance transform of input structure X_i. For the experimental work in this thesis, the distance transforms are always represented on a discrete grid, so the sample mean is simply the voxelwise mean.

This function $\overline{d}$ is a solution to the (modified) problem of finding the mean distance transform function. The final step is to map this back to a point set of $\mathbb{R}^3$ that can be identified as the mean value $\mathsf{E}X$. Unfortunately, the function $\overline{d}$ is not the distance transform of a point set (unless X is deterministic) [BM98]. A suitable point set can be obtained, however, by choosing a threshold t and defining

$$D_t = \{x \in \mathbb{R}^3 : \overline{d}(x) \leq t\}.$$

With a suitable choice of threshold, $\mathsf{E}X$ will be set to D_t. This can be interpreted in terms of the sample mean as selecting the points that are, on average, close to the input structures.

Baddeley and Molchanov propose a number of different ways to select the threshold value. Each method to obtain t is arrived at by optimizing some measure between the mean distance transform, $\overline{d}$, and the distance transform of $\mathsf{E}X$. Subsequent work by Lewis, Owens and Baddeley [LOB99] suggests a different strategy: choose t by optimizing a criterion based on D_t and the inputs, $\{X_i\}$. They suggest to minimize the mean absolute volume difference,

$$\frac{1}{N}\sum_{i=1}^{N} ||D_t| - |X_i||. \tag{4.12}$$

It is well-known (e.g. [GGD91]) that the mean absolute deviation about m,

$$\frac{1}{N}\sum_{i=1}^{N} |m - x_i|,$$

where m and x_i are real numbers, is minimized by choosing m to be a median value of $\{x_1, \ldots, x_N\}$. Thus, Expression 4.12 is minimized by selecting t such that $|D_t|$ is equal to the median volume of the input set $\{X_i\}$. This has the attractive property that the volume of the mean is equal to the average (median) of the input volumes, and is the criterion used in this thesis. For the implementation, the distance mean is discretized so $|D_t|$ will not generally exactly equal the median input volume, m; rather, t is chosen to be the smallest value such that $|D_t| \geq m$.

As an illustration of the improvement over Vorob'ev mean for small structures with little overlap, the same three inputs as used for Figure 4.21 and the distance mean are shown in Figure 4.22. In this example, the distance mean is more natural than the Vorob'ev mean. It is a single connected component, and located roughly where the middle input structure appears.

The distance mean for the central sulcus, after nonaffine spatial normalization, is shown in Figure 4.23. In both hemispheres, the mean structure is similar in appearance to an input central sulcus, albeit shorter and thicker.

The tendency of the distance mean to be shorter and thicker is actually a reflection

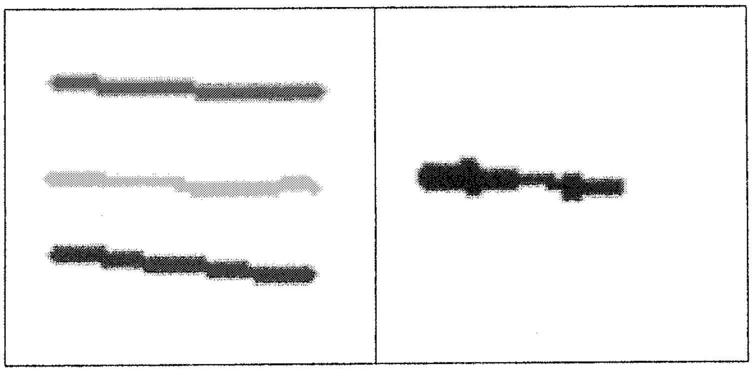


Figure 4.22: The left panel shows the same three input structures as in Figure 4.21. The resulting distance mean shown in the right panel is more natural than the Vorob'ev mean.

of the spatial variability of the inputs. If each input mapped exactly on top of all the others, then $\overline{d}$ would be zero at these points, and the volume of the set of points for which $\overline{d}$ is zero would be equal to the median volume of the inputs. In this case, the distance mean would be precisely the same as each input. If, on the other hand, the distribution in one region of the structure (e.g. the $z \leq 44$ slices in Figure 4.23) were more spread out than in another region (e.g. slice $z = 64$), the value of $\overline{d}$ in the former (more variable) region would tend to be larger than $\overline{d}$ in the latter (less variable) region. Points in the less variable region are more likely to be under the threshold and thus selected for the distance mean, at the expense of points in the more variable region.

4.2.4 Dispersion about the Distance Mean

In contrast to measuring dispersion using the template structure as the central structure, dispersion about the distance mean is expected to be a better measure of the input variability for two reasons. First, the distance mean is centered on the input data whereas the template structure could be from an extreme of the population variation. If the inputs were spatially concentrated in a small region while the template structure were an outlier, the dispersion measure could be quite large and thus unrepresentative of the actual dispersion of the input data. Second, as noted above the sample distance mean tends to be concentrated in the regions of low variability, away from regions of higher variability. This renders the dispersion about the distance mean more sensitive to variability of the inputs.

The distance mean for each of the eight structures before and after nonaffine spatial normalization is computed, and the dispersion measures about these means

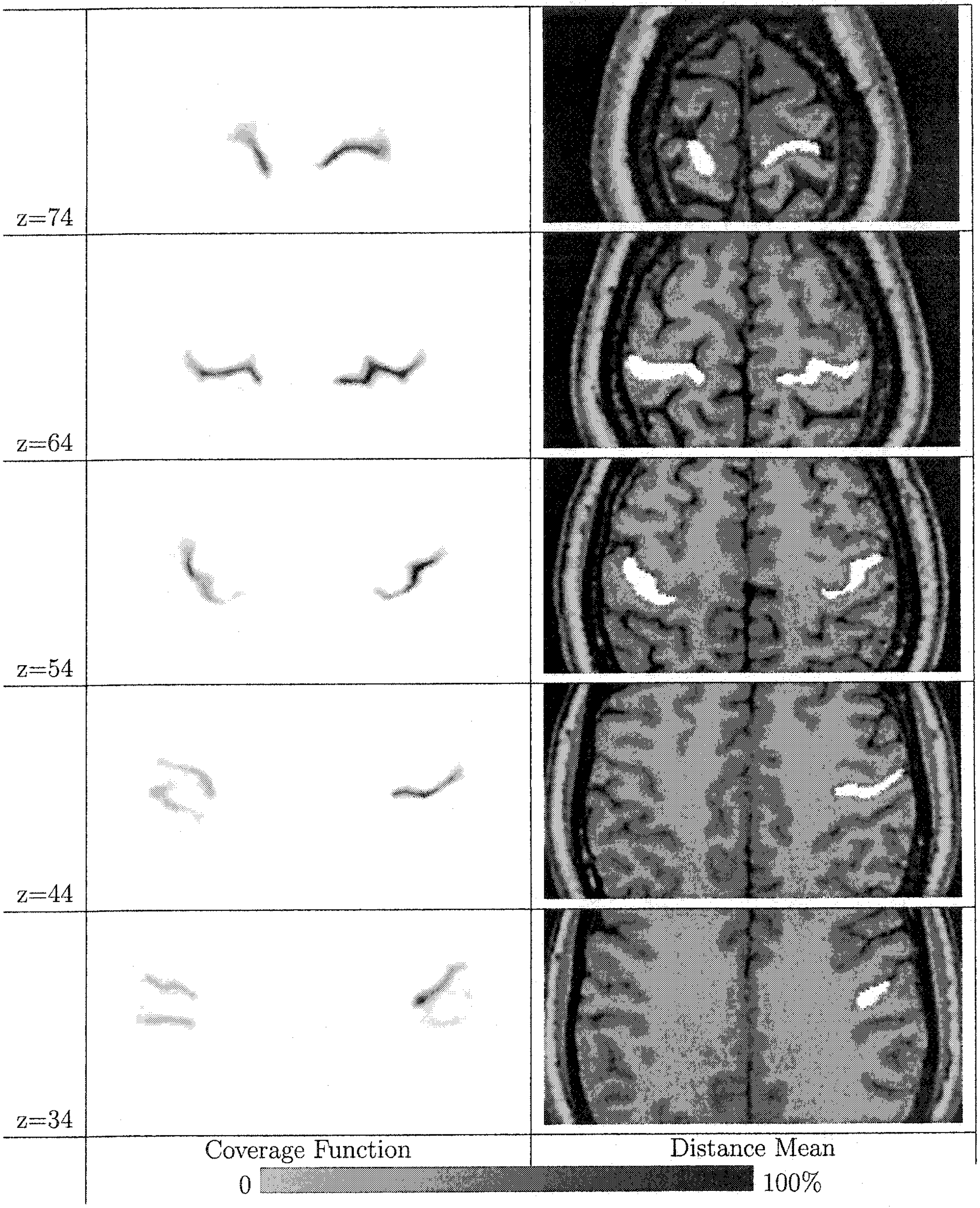


Figure 4.23: Coverage function and distance mean of the central sulcus after nonaffine normalization with 39 subjects. Notice that the distance mean in the left hemisphere does not extend down to the $z = 34$ and $z = 44$ slices.

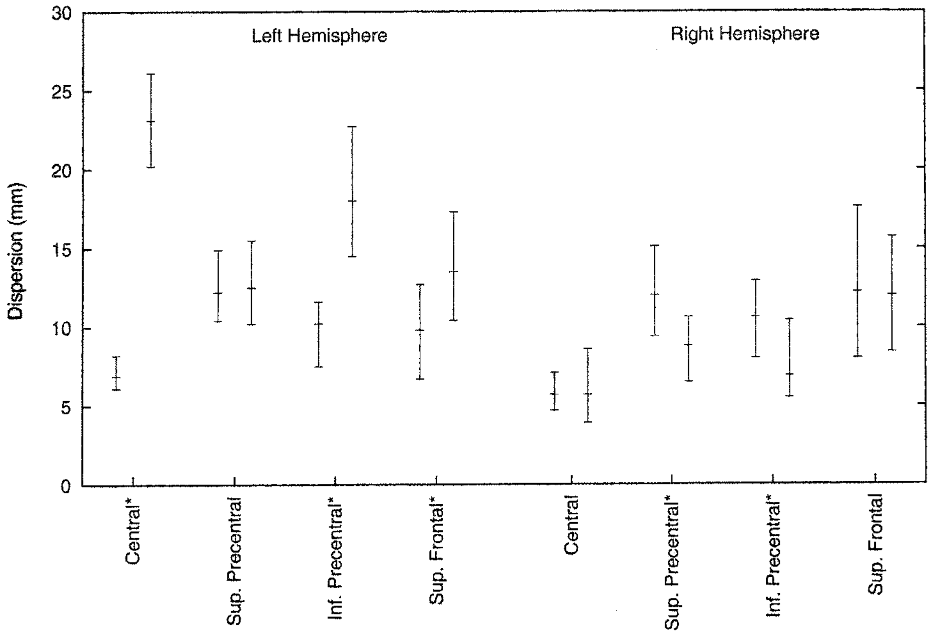


Figure 4.24: For each sulcus, the dispersion about the distance mean is displayed after 9-parameter affine normalization (left bar of each pair) and after nonaffine normalization (right bar of each pair). The central mark indicates the dispersion, which is the median of the set of distances from the distance mean, while the error bars display the 25% and 75% quantile values, computed with 39 subjects. The sulci for which the medians obtained with affine normalization and with nonaffine normalization differ significantly (sign test at $p \leq 0.05$) are marked by an asterisk.

are plotted in Figure 4.24. The biggest difference with respect to dispersion about the template is that 3 of 4 of the left hemisphere sulci show large increases in dispersion after nonaffine warping. This is due to the second effect, i.e. that the distance mean is concentrated in regions of lower spatial variability, as illustrated in Figure 4.23.

4.3 Localization Measure of Spatial Normalization

The goal of spatial normalization is to map a population into standard space such that corresponding structures are well-localized. Fischl et al. [FSTD99] measure localization by "computing the spatial spread over which a feature occurs across individuals". Their measure of spatial spread is the volume of the union of the structures after spatial normalization, divided by the mean volume of the structures. This measure captures some aspects of the spatial spread, but not all of them. Imagine spatially normalizing a population of 2 individuals and let X_1 be the first subject's

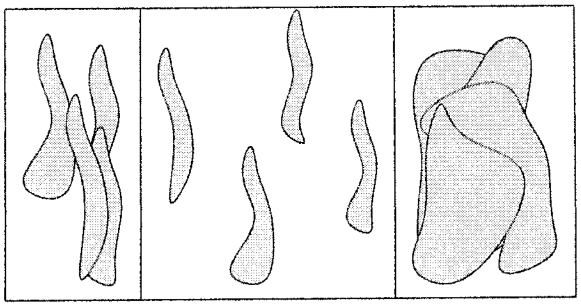


Figure 4.25: A good spatial normalization of four structures is shown in the left panel. In contrast, the middle panel shows a spatial distribution that is more disperse, and the right panel shows a spatial distribution in which the structures cover a larger area.

structure after normalization, while X_2 is the structure of the second subject after normalization. The spatial spread measure of Fischl et al. is given by

$$\alpha = \frac{|X_1 \cup X_2|}{\frac{1}{2}(|X_1| + |X_2|)}.$$

Assuming that the sizes of X_1 and X_2 are not appreciably different for the different spatial normalizations being compared, the denominator of α is roughly constant. The quantity α is then minimized when there is complete overlap, i.e. $X_1 \subset X_2$ or $X_2 \subset X_1$. The maximum value of α occurs when X_1 and X_2 are disjoint, which gives $\alpha = 2$. However, α is equal to 2 no matter how distant in space are X_1 and X_2. Thus, the measure does not completely capture the notion of "spatial spread". The measure α suffers from the same flaw as that of segmentation entropy, discussed at the end of Section 4.1.2 on page 83.

The desired measure of localization should capture the notion that the normalized structures should be neither too disperse, nor cover too large a volume. See Figure 4.25.

In contrast to the measure of Fischl et al., the dispersion measure discussed in Section 4.2 *is* sensitive to the spatial spread about a mean, e.g. the distance mean. However, the dispersion is *not* sensitive to the size of the mean: if input structures are mapped to large regions in a consistent fashion, the dispersion can be low. Thus while the dispersion may be part of the answer, it, too, is not the complete answer.

To proceed consider the situation for a distribution of real numbers, with mean $\overline{x}$ and standard deviation σ. A certain fraction of the population (depending upon the underlying distribution) is known to lie on the interval $[\overline{x} - \sigma, \overline{x} + \sigma]$. Let $M \equiv \{\overline{x}\}$ denote the mean point set and notice that the interval is M_σ, the dilation of M by

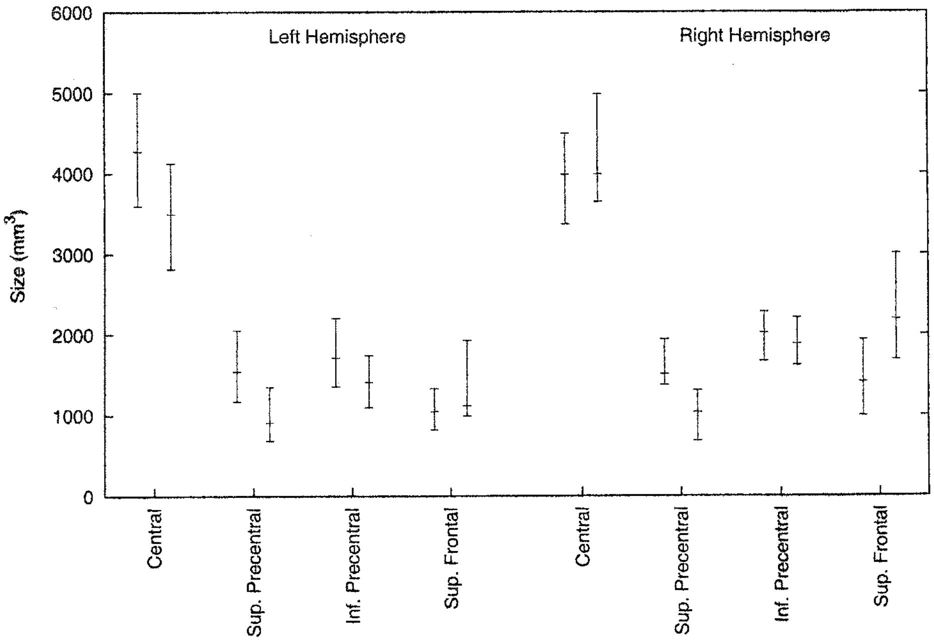


Figure 4.26: For each sulcus, the volume is displayed after 9-parameter affine normalization (left bar of each pair) and after nonaffine normalization (right bar of each pair). The central mark indicates the median while the error bars display the 25% and 75% quantile values, computed with 39 subjects.

σ defined by Equation 4.4. The size of the set M_σ (i.e. 2σ) is a measure of the localization: the smaller the size of M_σ, the better localized is the distribution. The localization for extended structures, such as the sulci considered in this thesis, can be measured in an analogous manner. Given a mean structure M (e.g. the distance mean), and a measure of dispersion about the mean σ (e.g. the dispersion defined in Section 4.2), the localization measure is taken to be the volume,

$$L = |M_\sigma|. \tag{4.13}$$

One way to think about this measure is that the distance mean, after dilating by the dispersion value, is a region that covers at least 90% of at least half the inputs. In contrast to simply measuring the union, however, the size of M_σ increases with dispersion.

Figure 4.26 shows that the structure volumes can be enlarged (e.g. superior frontal sulcus) or reduced (e.g. superior precentral sulcus) after normalization. The size of the distance mean is (approximately) the median value of the structure sizes so it, too, can be either enlarged or reduced by spatial normalization. The localization

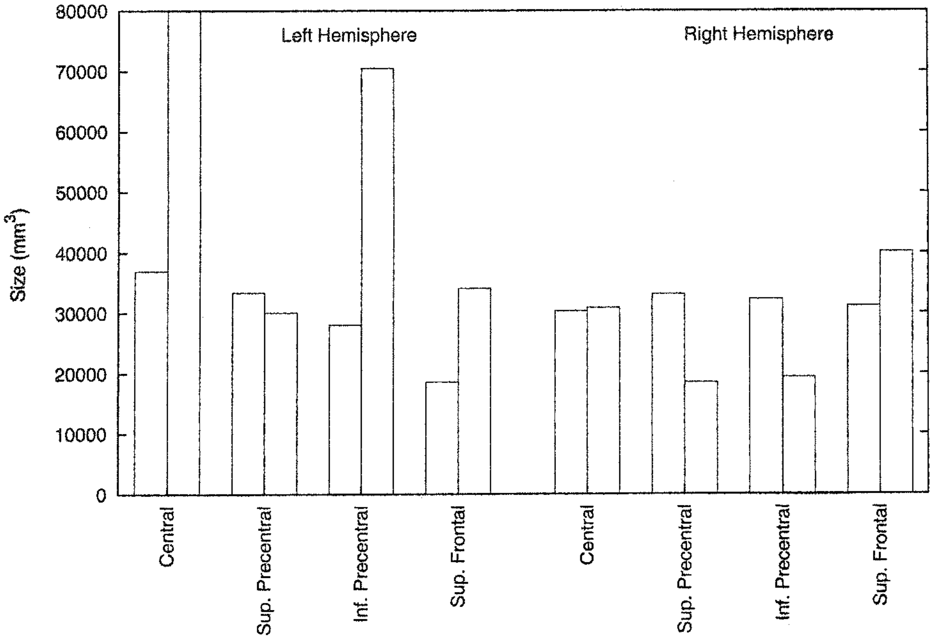


Figure 4.27: For each sulcus, the localization measure is displayed after 9-parameter affine normalization (left bar of each pair) and after nonaffine normalization (right bar of each pair) of 39 subjects. To distinguish better the values for most sulci, the vertical range is set to 80 000. The value for left central sulcus after nonaffine normalization is about 2×10^5 mm^3.

measures before and after nonaffine normalization are shown in Figure 4.27. The results, in terms of whether nonaffine normalization improved or degraded localization, generally agree with the measures of dispersion about the distance mean (Figure 4.24). Where the dispersion increases or decreases, so too does the localization value, with two exceptions. The left superior precentral shows about equal dispersion before and after nonaffine normalization. However, the size of the distance mean is considerably reduced afterwards, leading to a modest drop (improvement) in localization value. The right superior frontal sulcus shows about equal dispersion before and after nonaffine normalization with a larger distance mean afterwards, leading to a larger measure for localization. These examples indicate that the localization measure does indeed capture a quality distinct from the dispersion.

4.4 Previous Work

Measurements of variability (or the converse: alignment) after nonaffine registration most often appears in work validating a new algorithm, generally comparing the algorithm to one other algorithm (often an affine registration method). The goal of studies that compare nonaffine registration algorithms is often slightly different, namely to compare the impact of spatial normalization on analysis of functional data, e.g. [CSTM$^+$02, GAA97, MKFK94, SIO$^+$98, HAC$^+$02]. However, many of these studies also include some measure of anatomical variability as well.

The measurements of variability can be divided into two categories. The first category is measurements that compare a normalized image with the template, e.g. dispersion about the template discussed in Section 4.2.2. Such a measure can be misleading if the template is unrepresentative of the population, as discussed in Section 4.2.4. The second category of measurements avoids using the template. Examples of this approach are the total entropy discussed in Section 4.1 and the dispersion about a distance mean discussed in Section 4.2.4.

Template Agreement Measures

In validation work, perhaps because the goal is to show that the spatial transformation generates a good match between the source image and the target, the comparison most often measures agreement or discrepancy with the template. Comparative studies also use this type of pairwise measurement. A set of such pairwise measures can be averaged to show typical performance.

The measures can be as crude as simply checking that the volume of the brain, or some substructure such as the ventricles, agrees with the template volume after alignment, e.g. [CHPE95, KLT$^+$00, SD03]. Certainly this is not a very sensitive test of misregistration as a structure may have the same volume but be shifted with respect to the template structure, and it is not relied on as the only measurement.

Slightly more sophisticated tests measure segmentation agreement by counting the number of voxels on which the individual and the template segmentations agree (or disagree) after registration, e.g. [BK89, CRM94, CJM97, CSTM$^+$02, CHPE95, HB00, HBC$^+$01, HAC$^+$02, KPH$^+$96, KLT$^+$00, KLT$^+$02, SD03]. When applied to only two images, total entropy score defined in Section 4.1 is equal to the number of voxels for which the segmentations disagree. This is because in a pairwise measurement, voxels on which the registered image and the template agree contribute zero to the entropy. If the labels do not agree, then there are precisely two labels with

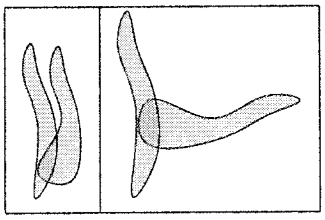


Figure 4.28: The two structures in the left panel are better aligned than the two in the right panel. The area of the union is the same in the left panel as in the right, as is the area of the intersection. Thus, the union and intersection measures do not completely capture the essence of "alignment".

$p_l = 1/2$ and the other labels have zero probability. The entropy for such a voxel is $2(-\frac{1}{2}\log_2\frac{1}{2}) = 1$, so the total entropy is just the number of voxels that disagree. Measures of overlap can be normalized by total number of voxels, the total number of voxels in the template structure, or the total number of voxels in the union of the two structures [GJC01].

Dispersion measures about template structures have also been used, for example with point landmarks [GBR$^+$99, KPH$^+$96, TT96] or with sulcal surfaces [CLGE98, HBC$^+$01] .

Group Measures

Variability measures that don't rely on a template are generally obtained by measuring a dispersion about a mean, using point landmark features [SAVK$^+$02, SKS$^+$99] or curves [KLT$^+$00, SKS$^+$99].

Warfield et al. [WRH$^+$01] introduce the total entropy (Equation 4.2) as a performance measure, borrowing the idea from Miller et al. [MMV00] who use the total entropy as the objective function to simultaneously co-register binary images of handwritten characters.

Other group measures of variability include the volume of intersection of all input structures used by Roland et al. [RGA$^+$97] for which a larger volume indicates better normalization. Fischl et al. [FSTD99] look instead at the volume of the union of all inputs, i.e., the volume with positive coverage function. In this case, a lower volume is best. However, there are qualitatively different situations, such as those illustrated in Figure 4.28, that cannot be distinguished by union nor by intersection. Woods et al. [WGW$^+$98] use sulcal and gyral curves to evaluate variability. The curves are rasterized, and at every voxel of a curve, the mean distance to the other $N-1$ homologous curves is computed. Rather than generating a single variability number, the distribution of these mean distances, over all curves is compared with the corresponding distribution of another spatial normalization method.

4.5 Summary

Two different strategies for measuring anatomical variability are presented in this chapter. Both measures can be applied to measure the performance of any registration method.

With either strategy, the variability is measured using a number of input images, each of which has an accompanying segmentation. In the work presented here, the segmentations carry anatomical information so anatomical variability is measured. However, the segmentations could equally well be functional regions in which case variability of a functional region would be measured. To compute the variability, a spatial normalization is first carried out, producing a set of transformations from the template to each input image. Each transformation induces a segmentation on a standard space grid, as illustrated in Figure 4.1.

Improvements to ANIMAL

The first strategy for measuring variability (Section 4.1) is based on measuring the consistency of the induced segmentations. After spatially normalizing the input data, each voxel v in the standard space grid maps to a certain point in subject i, which has label l_{vi} as shown in Figure 4.1. If the spatial normalization is truly matching homologous points, the labels thus associated with v will be consistently of the same class. The extent to which the labels are consistent is measured using total entropy, defined in Equation 4.2.

A segmentation of the entire brain into tissue classes gray matter, white matter, and CSF can be obtained automatically [Kol96, DFS99, JCD$^+$99]. Using such a segmentation allows a variability measure that is sensitive to label consistency across the entire brain. The entropy value measured using such a segmentation is thus a global measure of variability and it can be used to measure the performance of a registration algorithm for the purposes of spatial normalization. In Section 4.1.1, this performance measure is used to evaluate algorithm design choices and choose optimal values of numerical parameters for the ANIMAL algorithm (Section 3.7).

As distributed, the ANIMAL algorithm performs four iterations of the outer loop (Section 3.7.1), the first three using smoothed image data with control grid vertex spacing of 8 mm, 4 mm, and 2 mm. The fourth iteration uses smoothed gradient image data, with a 2 mm control grid. The entropy measure (see Figure 4.2) shows that the variability decreases after the first three iterations of the outer loop, but not after the fourth. Further investigation reveals that the node thinning heuristic that is designed to suppress the matching step for control mesh vertices located in non-brain regions of the image (Section 3.7.2) is not suitable for gradient image data. This heuristic suppresses matching at a control mesh vertex if the image intensity falls below 10% of the maximum intensity, which happens for almost all vertices when the image data is gradient magnitude, as shown in Figure 4.3. With node thinning, matching is only done on scalp and superficial cortex, and not on the internal structures of the brain. After the node thinning is removed from ANIMAL, the gradient fit iteration of the outer loop does indeed reduce variability, albeit not by much; see Figure 4.9. Ultimately, however, spatial normalization is concerned with minimizing the variability of structures like sulci. The entropy measured using a segmentation of frontal sulci rather than by tissue class (Section 4.1.2) shows in Figure 4.12 that variability using the gradient fit is larger than the variability using the intensity fit with the 2 mm grid. Thus, the gradient fit is omitted for results in this thesis.

The data term used by ANIMAL is also investigated using total entropy as a performance measure. The original ANIMAL algorithm uses normalized cross correlation (Equation 3.14) as the data term, which is equivalent to maximum likelihood under the assumption that the intensities of the two images are linearly related, as discussed in Section 3.2.2. The correlation coefficient (Equation 3.13), on the other hand, assumes an affine intensity relationship. An affine function has one more parameter than a linear function, which suggests it might be better able to match regions in which there are two tissue types such as near boundaries, as discussed in Section 4.1.1 on page 66. This is tested by replacing the cross correlation data term in ANIMAL by the correlation coefficient. The lowest variability achieved in the experiments performed with each data term is about the same, as show by Figures 4.4 and 4.5. However, these figures also show that the correlation coefficient data term still achieves low variability if the similarity cost ratio or the stiffness parameter is slightly non-optimal, whereas the variability when using cross correlation may be dramatically worse. In a situation where the algorithm is tuned using a set of test data and then put to general use with the selected parameters, this robustness to parameter choice is desirable. The optimal parameter choice for a given data set might indeed be slightly different from the parameter set in use. Thus, the correlation coefficient data term is used for the experiments in this thesis.

The total entropy is then used to obtain optimal values for the three major parameters of the ANIMAL inner loop (Algorithm 4): similarity cost ratio, stiffness, and weight. Figures 4.4 and 4.5 show that similarity=0.3, and stiffness=1 are optimal. The weight value, which doesn't affect the results greatly, is set to 1. These parameters are selected using a sample of 10 individuals, and Figure 4.9 shows that the performance is good for a second sample of 10 individuals using the same parameters.

Using the optimal parameters, a set of 39 individuals is spatially normalized to check whether the generated transformations are commonly non-injective. The results in Figure 4.8 show that there are pockets of non-injectivity. Overall, however, the Jacobian determinant is negative at only 0.2% of the control mesh vertices, so it is not a large problem.

In summary, using the total entropy as a performance measure for ANIMAL, in the context of spatial normalization, demonstrates that the following changes produce improved performance: remove node thinning, remove gradient fit, use correlation coefficient as the data term, and use parameters similarity=0.3, stiffness=1, weight=1. Figure 4.10 shows the quantitative improvement of these changes in terms of reduced entropy throughout the brain. Figure 4.11 shows the improvement qualitatively in terms of increased sharpness in the intensity average image.

Structure Variability

If homologous structures are being aligned by the spatial normalization, then gray matter is expected to align more often with gray matter, and similarly white matter with white matter. In other words, the label consistency measured by total entropy should decrease. However, the converse is not true: a decrease in entropy at a voxel because it is consistently assigned the label gray matter does not mean that the gray matter comes from the same sulcus in each individual. Figure 4.14 shows that this can happen, even for the central sulcus, which is relatively large and consistent in the population.

In order to quantify the variability of structures smaller than the totality of white matter or of gray matter, a more refined segmentation consisting of frontal sulci is used in Section 4.2. With the sulcus labelling, the total entropy will increase if a sulcus is matched to the wrong sulcus, in contrast to the tissue segmentation which will decrease because much of the tissue labels do match even if the sulcus is incorrectly matched to the wrong sulcus. However, as discussed at the end of Section 4.1.2 on page 83, the entropy of the sulcus labelling does not capture the spatial spread of the distribution of normalized sulci. The total entropy can decrease if the input sulci are well matched to, say, two folds on the template, regardless of how widely separated in space are the two folds, as illustrated in Figure 4.15.

To capture the spatial aspect of variability, a dispersion measure akin to median absolute deviation of univariate statistics is proposed in Section 4.2. Dispersion is defined as the median distance from a central structure to each of the input structures. The distance between structures is measured using trimmed Hausdorff distance (Equation 4.7). This distance measure is chosen because it is applicable to any closed point set, e.g. the gray matter object, and the "thick" sulcus structures used in this thesis. It does not require a point-to-point matching of the structures nor does it require any manual intervention.

Three methods for obtaining a central object about which dispersion can be measured are discussed in Section 4.2. One option is to measure dispersion about the template structure as in Section 4.2.2. This could be useful to gauge suitability of that template for the purpose of automated labelling, since labelling requires that the template structure be accurately matched to the corresponding structure in the population. The second option is to use the Vorob'ev mean structure, which is possibly useful when the input structures contain a lot of overlap, after normalization. The third option is the distance mean, which is useful for small structures that don't necessarily overlap much. Figures 4.21 and 4.22 compare the Vorob'ev and distance means on a synthetic example where there is not much overlap. The results of the distance mean are more natural. Thus the distance mean is used for dispersion measures of the frontal sulci.

Localization

A measurement of the localization of a given structure after applying spatial normalization should capture the notion that the normalized structures should be neither too disperse, nor cover too large a volume, as illustrated in Figure 4.25. From the notion of confidence intervals on the real line, which locate the region in which a given fraction of the population is concentrated, an analogous measure for extended objects is proposed. The localization measure is the size of the distance mean after dilating by the dispersion measure. This measure takes into account both the size of the mean as well as the dispersion about it. This is an improvement upon the volume of the union proposed by Fischl et al. [FSTD99] which isn't sensitive to the dispersion, as discussed in Section 4.3.

When comparing two competing spatial normalization methods, the smaller localization value is preferred.

Part II

2D Spatial Normalization

Chapter 5

Spatial Transformation

This chapter is concerned with defining an appropriate class of transformations for cortical surface spatial normalization, the 2D parallel of Chapter 2. Spatial normalization of the 2D cortical surface is conceptually the same problem as 3D spatial normalization, namely to obtain a transformation T from the standard space to an input cortical surface. However, there are two additional steps required, which are illustrated in Figure 5.1. First, the cortical surface mesh must be identified from the MR image; here there are several competing definitions of cortical surface mesh from which to choose. Second, the mapping from the surface situated in 3D to an auxiliary 2-manifold must be defined and computed.

5.1 Surface Mesh

Topology

The cortex of each hemisphere is a highly folded sheet of gray matter, which is attached to other structures deep in the brain such as the corpus callosum. The complete cortex consisting of two hemispheres can be modelled as a mesh using one of at least three different topologies. One option is to represent each cortical hemisphere using a topological disc [CDVE95]. However, this approach requires human neuroanatomical expertise to find the cortical boundaries deep in the brain [DECS99]. The second approach splits the MR data into two parts through the corpus callosum, which can be done automatically, e.g. [DFS99, RMPO$^+$02, RHXP02]. The surface of this combined object (one hemisphere plus half the corpus callosum) is then represented as a manifold without boundary, i.e., a topological sphere. The cortex is modelled as two such topological spheres, each of which includes an arbitrary "cap"

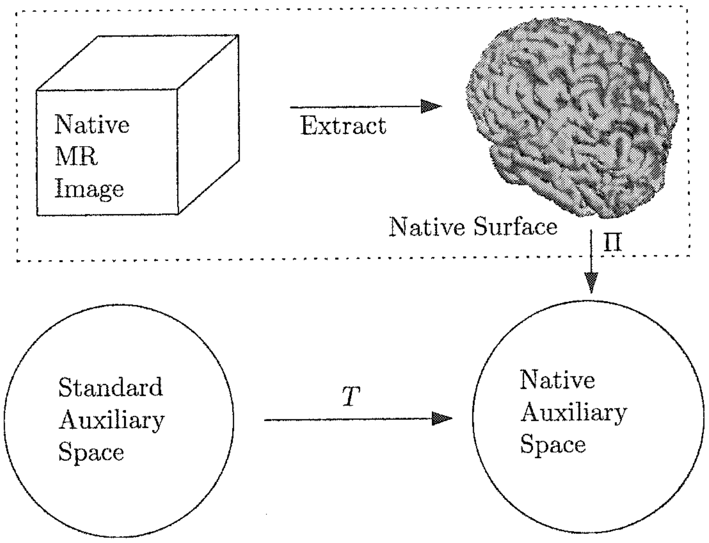


Figure 5.1: Starting with a native MR image, a cortical surface is extracted and then projected onto an auxiliary space (mapping Π) such as the plane or the unit sphere. After these two preliminary steps, the mapping T from standard auxiliary space to the native auxiliary space can be obtained. The two initial steps, enclosed in the dashed-line box, are unique to the 2D normalization problem.

through structures of the mid-brain [DFS99]. This cap does not represent cortex, but it is typically small in area and considered neglectable in subsequent analysis. The third approach uses a single topological sphere to cover both cortical hemispheres [MKAE00]. In this case, the surface consists of two cortical hemispheres, joined by the top part of the corpus callosum and an arbitrary cap through the brain stem below. These three possibilities are illustrated in Figure 5.2.

In the first two cases a separate auxiliary space is used for each hemisphere with left and right cortical hemispheres mapped separately to their respective auxiliary spaces, while the third approach maps both hemispheres to a single auxiliary space. From now on, the auxiliary space topology will be described simply as a disc or a sphere, understanding that the sphere may model one hemisphere or both.

Choice of Layer

Since the cortex actually has some thickness, there are several candidates that may be used as the cortical surface, illustrated in Figure 5.3. The interface between the white matter and the gray matter could be used, for example. Or the outer surface of the gray matter (also called the "pial surface" because the cortex is bounded by "pia matter") could be used. However, using either of these surfaces for brain mapping

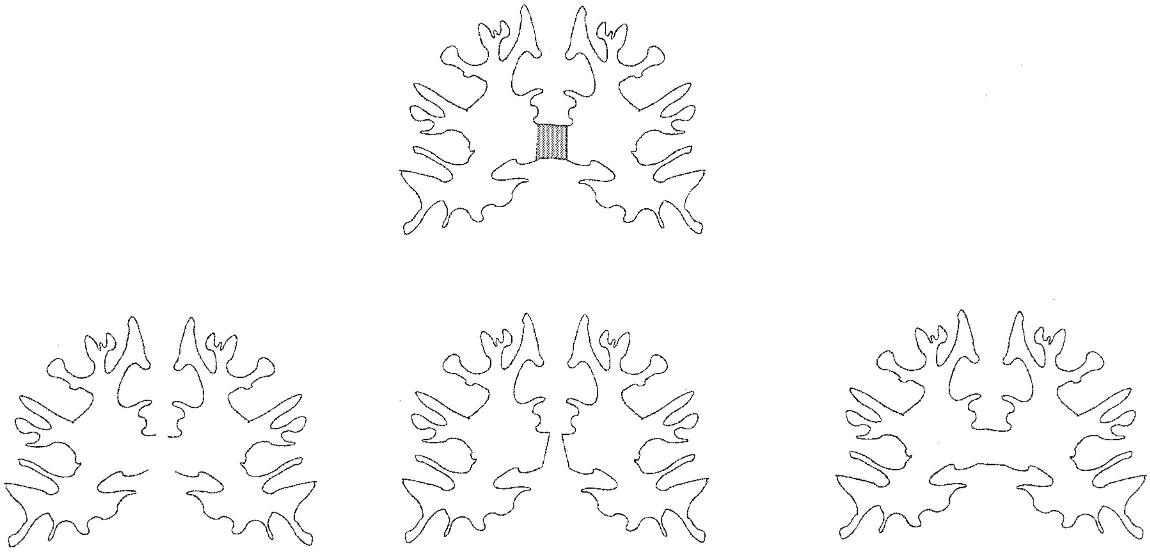


Figure 5.2: Schematic coronal slices through the cortical surface. The top figure shows the actual situation: cortex consists of two hemispheres attached to the corpus callosum (shaded). Topology of the model could be (bottom row, left to right): two discs, two spheres, or a single sphere. In order to interpret the bottom row, note that a slice through a topological disc (i.e. a manifold with boundary) is an open curve in the plane, while a slice through a topological sphere produces a closed curve.

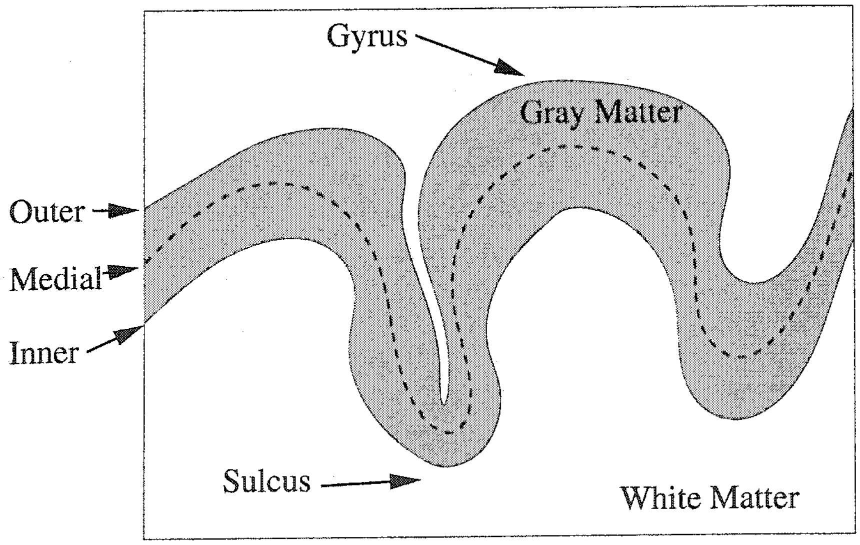


Figure 5.3: Three possible cortical surfaces: the outer surface is the external surface of the cortex, the inner surface is the gray matter/white matter interface, and the medial surface is between the two.

can be criticized (e.g. [DECS99]) as being unrepresentative of the cortical surface. Along a fold, one surface (outer surface at the crown of a gyrus, inner surface at the fundus of a sulcus) tends to have more surface area than the other surface. As a compromise, therefore, some groups use a surface through the middle of the cortical sheet. This surface cannot be discerned by a particular intensity on an MR image, but can be computed using a function that seeks the "centre of mass" of the gray matter [DB96, XPR+99] or using cues based on intensity and its gradient [VEDD+01]. Thus there are at least three reasonable choices for "the" cortical surface.

Computing the Mesh

There are several methods used to extract surfaces from volume data. An early method is to manually trace polygonal line contours in a stack of image planes and then link nodes between adjacent planes [JTH+89, She92]. This is still used with data obtained from histological sections [VEDD+01]. With MR or cryosection data, on the other hand, the surface reconstruction is generally done directly in 3D, employing one or both of the following two techniques.

One technique uses an intensity value that lies midway between the intensities of white matter and of gray matter. The locus of points in the (continuous) image that have this intensity value contains the interface between gray and white matter. Since other parts of the head, notably skin and fat, can appear with intensities in the same range as white matter, this set can contain many connected components. The set is usually post-processed (e.g. to remove all but the largest component), and the result is referred to as an *iso-value surface*, or *iso-surface*.

To ease the problem of choosing the iso-value the image is often pre-processed, classifying each voxel as to whether it lies in white matter, gray matter, or neither, using standard classification techniques, e.g. [Kol96, ZFE02]. The result is a classified image in which each voxel contains its class identifier, e.g. value 2 (white), 1 (gray), or 0 (background), as shown in Figure 5.4. On such an image the surface sought has a known iso-value of 1.5. Level-set methods [Set96, MS98] offer an alternate form of classification from which a surface can be generated.

Once the iso-value is chosen, a polyhedron with triangular facets is generated, typically using a variant of the marching cubes method [LC87]. Surfaces produced in this manner typically have a large number of triangles and biologically-implausibly sharp edges and corners. The surfaces are generally decimated and smoothed to alleviate this. In addition, with the exception of the topology-preserving segmentation of Mangin et al [MFB+95], the surfaces will generally have non-zero genus and re-

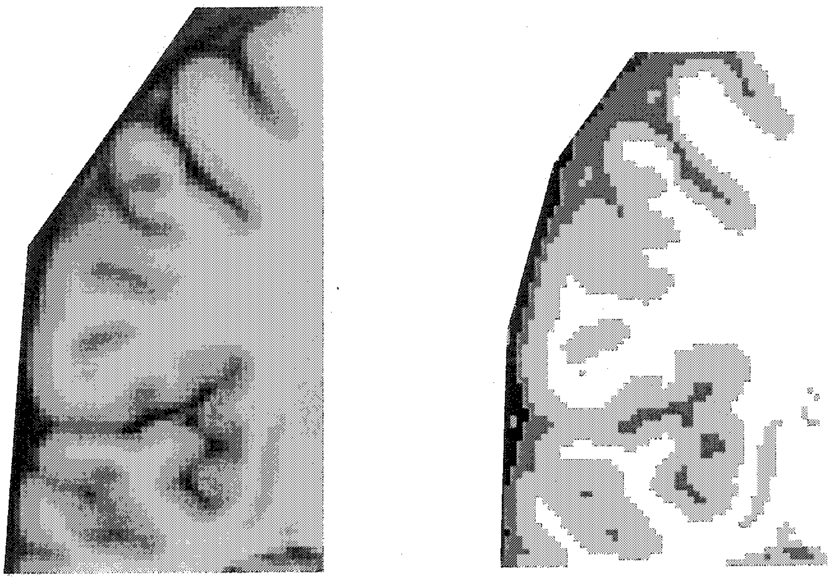


Figure 5.4: Left: the MR image. Right: voxels classified according to tissue class. The outer cortical surface boundary is difficult to discern from MR data, especially in sulci.

quire editing to bring the genus down to zero. The editing is typically either manual [DECS99] or based on intricate heuristics [FLD01, KG01, SL01, VEDD+01].

A second technique for computing a boundary surface is based on iteratively deforming a trial surface represented explicitly (e.g. as a polyhedron) until it conforms to the desired boundary. This *deformable surface* approach has the advantage that the surface topology can be fixed in advance (e.g. to be spherical) and the deformation process can maintain a simple surface at all times [Mac98], obviating the need for manual editing. The deformation process also has provisions for controlling the smoothness of the surface during deformation, rather than smoothing afterwards. One drawback of this approach is that it is much more computationally demanding than marching cubes. A fair comparison with iso-surfacing, however, must include the effort required to correct topological defects, which may take several hours on an SGI Octane [VEDD+01].

Computing the outer surface is problematic because the gap between opposing banks of many sulci is at or below the resolution of the image and thus is not visible. See Figure 5.4 for example. This is much less of a problem with the inner surface, so the inner surface is used for experiments presented in this thesis. The surface is obtained using the ASP software, developed by MacDonald [Mac98].

Cortical Surface Extraction Literature

The software suite of Van Essen's group [VEDD$^+$01] contains provisions for either manual contour tracing or automated iso-surface extraction and smoothing, using the medial surface in either case. Joshi et al. [JCD$^+$99] use an iso-surface method on the white matter segmentation. Malladi et al. [MS98] use a level-set technique for segmentation and then compute an iso-surface. Xu et al. [XPR$^+$99] compute the iso-surface of the gray/white interface after fuzzy segmentation, then use a deformable surface technique to find the medial surface. Dale et al. [DFS99] start with the surface of a white matter segmentation, and then deform it to seek the outer surface. None of the preceding guarantees a surface of genus 0 without post-processing. Davatzikos and Bryan [DB96] use a deformable mesh that seeks the medial surface, but it is not able to get into the depths of the sulci. Thompson and Toga [TT96] use a deformable mesh driven by the gradient image, thresholded and subsequently blurred. MacDonald et al. [MAE94, MAE98, MKAE00] use a deformable surface to find first the inner surface, then the outer surface, using the former to constrain the latter. Throughout the deformation, the surface remains geometrically simple and of genus 0.

5.2 Auxiliary Space

In contrast to the situation in 3D, the domain and range for the mapping T is not self-evident. While it is conceivable to use the native mesh just described as the domain and range of the mapping T, a simpler auxiliary space is generally used instead. There are at least three different reasons for using an auxiliary space in preference to the manifold of the native mesh.

The first reason for mapping to a simpler manifold is to allow visualization of surface data that is buried in the folds [VEDJM98, FSD99]. With a standard rendering of the native cortical mesh, a good deal of the cortex buried in the folds is occluded from view. In order to bring the buried cortex into view, the surface can be mapped to the plane (sometimes called *flat mapping*), to a sphere or ellipsoid, or to a smoothed version of the cortex [VEDJM98]. One method of smoothing the cortex is described in more detail in Section 6.2.2.

The second reason for mapping to a simpler manifold is to facilitate communication of surface locations of interest between laboratories. Mapping to the plane or the sphere allows standard coordinate systems to be used. Cartesian coordinates are used for flat maps [VEDJM98] while polar coordinates are used on the sphere

[VEDJM98, FSD99].

As the topic of this thesis is the registration problem, these two rationales for using an auxiliary surface will not be considered further. In any case, software allows one to use multiple surfaces and to map information from one representation to another [VEDD$^+$01]. For example, a coordinate system set up using, say, the sphere can be mapped on to a flat map intended for visualization.

The third reason to use an auxiliary surface is to simplify implementation of surface registration. All the 3D registration algorithms described in Chapter 3 employ a single coordinate system to locate points of $\mathbb{R}^3$. This simplifies algorithms tremendously, e.g. the distance computation between landmark points is a very simple operation in $\mathbb{R}^3$ compared to computing the geodesic distance on the native mesh. Similarly, the smoothing operations used in a 4-step algorithm (Algorithm 3) implemented on an arbitrary manifold requires an effective method for transporting a vector between tangent spaces of nearby points in order to integrate or average displacement vectors in a neighbourhood. This complexity is avoided by choosing a simpler auxiliary surface on which to work.

Geometry

If the cortex is modelled by a topological disc then the auxiliary space should also be a topological disc, i.e. a 2-manifold with boundary, or something into which a disc can be embedded such as the plane. The plane has the advantage that all the spatial normalization algorithms used for 3D can be straightforwardly adapted for the plane. Van Essen et al. [VEDJM98] apply the landmark matching of Joshi and Miller [JM00] to surfaces mapped into the plane.

If the cortex is modelled using a sphere, then the auxiliary space could be the unit sphere $\mathbb{S}^2$. There are a number of coordinate systems possible on the sphere, each with its own strengths and weaknesses. Polar coordinates, used by Thompson and Toga [TT96, TT99] and by Fischl et al. [FSTD99], suffer from coordinate problems at the poles where two points separated by a small distance have a large difference in the azimuthal (i.e. "longitudinal") coordinate. Davatzikos and Bryan [DB95, DB96, Dav97] use two unit squares in the plane with boundary conditions as illustrated in Figure 5.5. These boundary conditions also introduce extra complexities in the registration problem of computing T [VD99], for example smoothing $T(x)$ for x near an edge. These problems are avoided by embedding the sphere in $\mathbb{R}^3$ and representing each point using a unit-length 3-vector. This parameterization, used by Vaillant and Davatzikos [VD99] avoids the problems at the poles and at patch

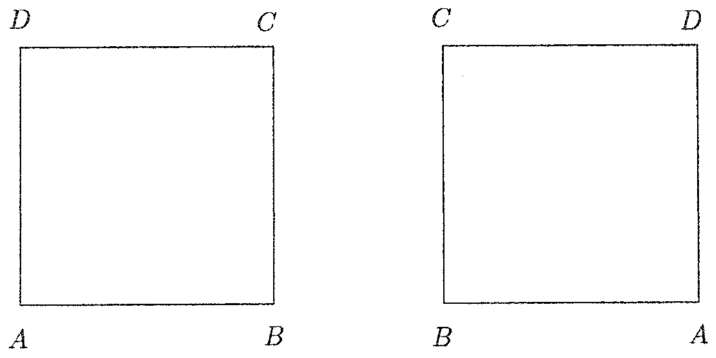


Figure 5.5: Parameterization of a spherical mesh using two unit square coordinate patches. Corners and edges with the same label are identified in the obvious manner.

boundaries described above, albeit at the cost of using 3 coordinate values to specify a point on a 2-manifold.

Projection of Native Mesh

The cortical surface is initially obtained as a mesh situated in 3D, whereas the auxiliary space is the plane or the unit sphere. The next step in surface normalization is to find the transformation Π that maps the surface bijectively to the auxiliary space as shown in Figure 5.1.

The domain of map Π is the triangulated mesh that represents the native cortex. A piecewise-linear map defined on a triangulated mesh is conveniently specified by its value at each vertex. The value of the map at a non-vertex point x is obtained by linear interpolation of the map values at the three vertices of the facet in which x lies, as described in Section 5.4.2 (Definition 5.4.5 in the plane and Definition 5.4.9 on the sphere). To specify Π, it suffices to find $\Pi(v)$ for each vertex v of the mesh.

The initial projection to the plane or the sphere can be obtained using any of several different techniques: iterative deformation (forward or reverse), conformal mapping, and harmonic mapping.

Iterative Deformation

Some methods for mapping the cortex to the plane or a sphere borrow from work on deformable surfaces. The idea is to apply forces at each mesh vertex and simulate its motion. The procedure is engineered to ensure that all the vertices eventually end up in a common plane or sphere. The value of map Π is obtained for vertex v by its final position after deformation.

The deformation generally comprises two stages, with a number of iterations performed at each stage. The iterations of the first stage smooth the mesh in $\mathbb{R}^3$, e.g. by applying a force that pushes each vertex towards the centroid of its neighbours. This force has (unless the point is coplanar with its neighbours) a component directed along the normal to the surface, which tends to make the mesh locally flatter. In addition, forces directed along the plane of each incident facet are added to discourage metric distortion. For example, a spring-like force directed along each edge penalizes compression or expansion of the edge from its initial length. Forces designed to penalize a change in facet shape, such as a change in angle or area, may also be used.

After the first stage of iterations, the mesh is projected into the auxiliary space. If the auxiliary space is planar, a suitable projection plane is chosen, e.g. by computing the mean of the facet normal vectors (of the mesh position after the first stage of deformations) and using a plane perpendicular to this mean normal direction. In the case of a spherical auxiliary space, the mesh vertices can be projected radially to the desired distance from the global centroid. Note that this projection may introduce overlaps, rendering Π non-injective.

The second stage of iterations keeps the mesh in the auxiliary plane or sphere. For planar space, this is achieved by simply using forces directed in the plane. In the case of the sphere, the mesh vertices may be allowed to move in $\mathbb{R}^3$ and are then projected back to the sphere after each iteration.

The projection step after the first stage of iterations may result in a mesh that is "folded", due to a triangle of the surface mesh having flipped over on top of an adjacent triangle. The second stage of iterations generally attempts to remedy this. In addition, the second stage is often used to reposition vertices to minimize the change in edge length, facet area, or facet angle between the initial mesh and the deformed mesh.

This approach is used by Van Essen's group [CDVE95, DVEA$^+$96, DECS99] by Dale, Fischl, and Sereno [DFS99, FSD99], and by Wandell, Chial, and Backus [WCB00]. Similar ideas were proposed to generate a surface mapping for shape morphing [KCP92]. A large amount of effort is spent dealing with the non-injective mapping that arises if a triangle is flipped over. Drury et al. detect this occurrence and manually re-start the procedure with different parameters. Dale et al. deal with this by incorporating an additional force term that penalizes flipped triangles during the second stage of iterations [FSD99]; setting its coefficient large enough will ensure that no triangles are flipped.

Reverse Iterative Deformation

Recall from Section 5.1 that one approach to computing a cortical surface from the image data is to iteratively deform a mesh (initially a sphere, say) so that it conforms to the desired interface of the 3D image. Treating the initial sphere of the mesh as the auxiliary space, the initial and final positions of the mesh vertices during surface extraction can be used to generate Π: if vertex v of the native cortex (final mesh) originated from point p_v on the initial mesh, set $\Pi(v) = p_v$.

MacDonald et al. [Mac98, MKAE00], Thompson and Toga [TT96, TT99], and Davatzikos and Bryan [DB95, DB96, Dav97] each use the map obtained after identifying the cortex using a deformable surface.

Conformal Mapping

A conformal (i.e. angle-preserving) map between two genus 0 surfaces always exists. Such a mapping is attractive since it is unique up to isometries, and is guaranteed to be bijective.

Angenent et al. use a finite-element approximation of a partial differential equation to map cortex (modelled as a topological sphere) to the unit sphere [AHTK99a, AHTK99b]. The core of this method is solving two systems of linear equations. The systems are large, but sparse and so can be solved rapidly; in [AHTK99b] they report computing a mapping for a surface composed of 430 000 triangles in less than 6 minutes on a Sun Ultrasparc 10.

Hurdal et al. [HBS+99] start with a circle packing of the cortex, modelled as a topological disc. A circle packing of a mesh associates a circle to each vertex with the property that two circles are tangent if their respective vertices are joined by an edge of the mesh. Radii of the circles for boundary mesh vertices are parameters; once chosen, the radii for internal vertices are fixed [CS03]. The position in the plane for each vertex is fixed by the radii and mesh structure, up to isometries. A circle packing (with no "branch points") will give a flattened mesh with consistently-oriented triangles, which is sufficient for Π as it only needs to be injective. By iteratively adjusting the boundary radii, the mapping can be made more nearly conformal.

Harmonic Mapping

A topological disc can always be harmonically mapped to a convex polygon in the plane. The map is constructed from an optimization problem that treats mesh edges as springs. This approach is used to generate correspondences between meshes for purposes of morphing [EDD$^+$95, KSK98]. Although a harmonic map is a homeomorphism, the map generated is usually a piecewise-affine approximation, so the possibility of adjacent facets overlapping still exists.

Graph Drawing Approaches

The surface mesh is a planar graph with triangular facets and the cyclic ordering of neighbours about each vertex is known. In the language of graph drawing [DBETT99], the mesh is an *embedded graph*, i.e. the cyclic ordering of the neighbours is specified. In fact, since the graph is triconnected, the embedding is unique, up to a reversal of the neighbour ordering. The problem of mapping the mesh to the plane can be reduced to producing a straight-line drawing of the graph. Floater approaches the problem in this manner [Flo97]. Alexa [Ale00] uses ideas from planar graph drawing to embed the mesh onto the sphere.

5.3 Low-Dimensional Warping

The choice of auxiliary space influences which classes of spatial transformations are suitable. One natural class of low-dimensional mappings for the sphere is rotations, which require 3 parameters.

In the plane the natural low-dimensional mappings are the affine transformations, just as is the case in $\mathbb{R}^3$ (Section 2.3). A general 2D affine transformation has six parameters, e.g. a 2×2 matrix A and a 2-vector b with the transformation written as $T : x \mapsto Ax + b$. Analogous to the 9-parameter subset of affine transformations used in $\mathbb{R}^3$ is the subset of transformations comprised of rigid body transformations followed by independent scaling of the x_1 and x_2 axes, requiring a total of $3 + 2 = 5$ parameters.

Such low-dimensional warpings can be used as the first stage of alignment. However, the experience in 3D (Chapter 4) suggests that a more flexible set of transformations is required to match fine detail, e.g. structures such as sulci.

5.4 High-Dimensional Warping

When using the plane as the auxiliary space, any 3D technique of Section 2.4, e.g. splines or mesh warping, can be adapted for use in a straightforward manner. For example, Van Essen et al. [VEDJM98] use the fluid landmark approach of Joshi and Miller [JM00] after mapping the cortical hemispheres to the plane.

The remainder of this section concentrates on the use of the sphere $\mathbb{S}^2$ as the auxiliary space, as the spherical topology introduces a few subtleties in the mapping function.

5.4.1 Global Transformations

On the sphere, basis functions with global support can be used to expand the mapping function. Spherical harmonics are a natural choice for basis functions, and have been used in this context by Thompson and Toga [TT96] and by Bakırcıoğlu, Joshi, and Miller [BJM99]. As in 3D, however, a local transformation is used as it allows better control of the transformation.

5.4.2 Local Transformations

Working in a spherical topology, it is possible to adapt the 2D mesh warping approach (Definition 2.4.1) by working in a 2D parameter space, such as polar coordinates. Davatzikos uses [Dav96] such a mesh warping on his pair of unit squares domain (Figure 5.5). However, the spherical topology imposes non-trivial boundary conditions so this approach will not be pursued. Instead, the sphere is embedded in $\mathbb{R}^3$ and a point on the sphere is specified using a unit-length 3-vector.

A local transformation is obtained by defining T piecewise on the sphere, in the same spirit as the mesh warping used in 3D. In contrast to mesh warping, it turns out to be advantageous to use triangular subdomains rather than quadrilateral since the control mesh is obtained from the projection of the (triangulated) native mesh to the sphere using Π (see Section 6.7). In addition, testing for injectivity of the map T (discussed in Section 5.5) is easier when the subdomains are triangular. The other difference vis-a-vis mesh warping is that it is more convenient to store, at each mesh vertex v, the value of the transformation $T(v)$ itself rather than a deformation vector $\Delta(v)$, as discussed in Section 6.7.

Warping a triangulated mesh can also be used to define a transformation in the plane. The required concepts and definitions are introduced with the planar version before describing the the warping method for the sphere.

Triangulation Warping in the Plane

Cartographers have long used isomorphic triangulated mesh pairs to register (2D) maps of the earth; see Goshtasby [Gos86], Saalfeld [Saa87], and references therein. The main idea is to map each triangle in the domain mesh bijectively to the corresponding triangle in the range mesh (see Figure 5.7). Though higher-order polynomials can be used [Gos87] to generate a smoother transformation, an affine map between triangle pairs is used here. Meshes employed in brain mapping generally have triangles that are small enough that extra smoothness at the triangle boundary will not bring any benefit, while imposing a cost in terms of more complicated injectivity conditions.

An affine map from $\mathbb{R}^2$ to $\mathbb{R}^2$ has six parameters. Using homogeneous coordinates, write the map as

$$\begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix} \longmapsto \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}. \tag{5.1}$$

The six parameters $\{t_{ij}\}$ are obtained by solving a system of three equations involving 2D points, which are conveniently given by two triangles. Let $A = (a_1, a_2, 1)$, $B = (b_1, b_2, 1)$, and $C = (c_1, c_2, 1)$ be coordinates of three non-collinear points in the plane. Similarly, let $A' = (a'_1, a'_2, 1)$, $B' = (b'_1, b'_2, 1)$, and $C' = (c'_1, c'_2, 1)$ be three non-collinear points. Let T be an affine transformation such that $T(A) = A'$, $T(B) = B'$, and $T(C) = C'$. Using Equation 5.1, these three equations are written as columns of a 3x3 matrix equation,

$$\begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} a'_1 & b'_1 & c'_1 \\ a'_2 & b'_2 & c'_2 \\ 1 & 1 & 1 \end{pmatrix}. \tag{5.2}$$

Since points A, B, and C are not collinear, the second matrix on the left hand side can be inverted to solve for the matrix of T. Since in addition points A', B', and C' are not collinear, the determinant of T is non-zero. The transformation T is therefore a valid affine transformation. Since the second and third matrices are non-singular, the solution for T is unique. The above is summarized in the following lemma.

Lemma 5.4.1 *Let A, B, and C be three non-collinear points in $\mathbb{R}^2$. Let A', B', and C' be three non-collinear points in $\mathbb{R}^2$. There is a unique affine transformation T such that $T(A) = A'$, $T(B) = B'$, and $T(C) = C'$.*

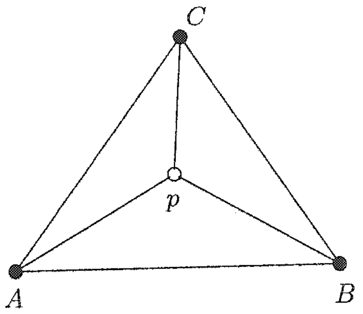


Figure 5.6: Areal coordinates (α, β, γ) for p are proportional to the signed area of triangles BCp, CAp, and ABp respectively. The coordinates are all nonnegative if, and only if, p lies in triangle ABC.

The mapping T can thus be called *the* affine map of ABC to $A'B'C'$.

Consider now the mapping of points of the triangle interior. Let ABC be a nondegenerate triangle in $\mathbb{R}^2$. Masses α, β, γ (with $\alpha + \beta + \gamma \neq 0$) placed at vertices of the triangle determine a unique point p that balances the plane. The triple (α, β, γ) is known as *barycentric coordinates* of p [Cox69, section 13.7]. The coordinates are homogeneous: for any $\lambda \neq 0$, $(\lambda\alpha, \lambda\beta, \lambda\gamma)$ specifies the same point. This freedom is used to normalize the coordinates such that

$$\alpha + \beta + \gamma = 1. \tag{5.3}$$

Barycentric coordinates so normalized are called *areal coordinates* and give a formula for the point, $p = \alpha A + \beta B + \gamma C$. Henceforth, all barycentric coordinates are assumed to be areal.

The coordinates are proportional to the (signed) area of triangles BCp, CAp, and ABp respectively [Cox69, section 13.7]. Thus the coordinates are all positive if, and only if, p lies on the interior of triangle ABC. Using Equation 5.3 to eliminate α,

$$\begin{aligned} p &= A + \beta(B - A) + \gamma(C - A) && (5.4) \\ &= A + \beta U + \gamma V, && (5.5) \end{aligned}$$

where $U = B - A$ and $V = C - A$.

Given the triangle vertices and point p, the barycentric coordinates are obtained by solving a 2x2 linear system. Take the dot product of $(p - A)$ with U and with V, obtaining the following system of equations,

$$\begin{pmatrix} U^2 & U \cdot V \\ U \cdot V & V^2 \end{pmatrix} \begin{pmatrix} \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} (p-A) \cdot U \\ (p-A) \cdot V \end{pmatrix}. \tag{5.6}$$

The determinant of the matrix is $U^2V^2(1 - \cos^2\theta)$, where θ is the angle between vectors U and V. Since the triangle ABC is assumed to be non-degenerate, θ is neither 0 nor π so this determinant is not zero; thus the system has a unique solution.

The barycentric coordinates are useful for interpolation, both for real functions defined on triangle ABC and for mapping from triangle ABC to $\mathbb{R}^2$. Let (α, β, γ) be the barycentric coordinates for point p with respect to triangle ABC, which can be computed using Equation 5.6. Let $A'B'C'$ be a second non-degenerate triangle and define the mapping

$$T : A + \beta(B - A) + \gamma(C - A) \mapsto A' + \beta(B' - A') + \gamma(C' - A'). \tag{5.7}$$

In other words, point $T(p)$ has the same barycentric coordinates with respect to triangle $A'B'C'$ as p has with respect to triangle ABC. Equation 5.6 shows that β and γ are affine functions of the coordinates of p and so therefore is the mapping T. The vertices A, B, and C have barycentric coordinates of $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$ respectively, so $T(A) = A'$, $T(B) = B'$, and $T(C) = C'$. In light of Lemma 5.4.1, the mapping T defined by Equation 5.7 is the unique affine map of ABC onto $A'B'C'$.

Note that Equations 5.4, 5.6, and 5.7 are all valid when p, A, B, etc. are points in $\mathbb{R}^3$. This is used later to construct a map between the unit sphere and itself.

For p on edge AB, the barycentric coordinate γ is zero. The map of Equation 5.7 reduces to $A + \beta(B - A) \mapsto A' + \beta(B' - A')$, i.e., it is an affine map from edge AB to edge $A'B'$. It is clear that edge AB is mapped to edge $A'B'$ in a manner that depends only on the edges involved (AB and $A'B'$) and does not depend on C or C'. Similarly, edge BC maps to $B'C'$ and edge CA maps to edge $C'A'$; in each case the map only depends on the edges involved. Interior points of triangle ABC comprise the set of points with positive barycentric coordinates, so interior points of ABC are mapped to interior points of $A'B'C'$. These observations are summarized in the following lemma.

Lemma 5.4.2 *The affine map of ABC to $A'B'C'$ maps interior points of ABC to interior points of $A'B'C'$. Each edge of ABC is mapped to the corresponding edge of $A'B'C'$, each such map depending only on the two edges involved.*

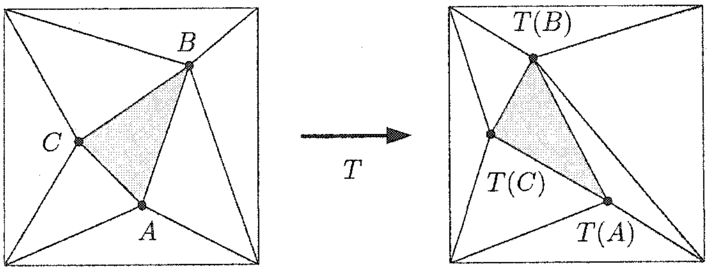


Figure 5.7: Plane triangulation warp. Each triangle is mapped using the unique affine transformation that maps corresponding vertices, e.g. vertex A to vertex $T(A)$, etc.

The preceding method for affinely mapping a single triangle is now extended to mapping a region of the plane, $\mathcal{D}$, employing a triangulation of the latter.

Definition 5.4.3 *Plane Triangulation. Let S be a finite set of non-degenerate triangles and let $\mathcal{D} = \cup_{\tau \in S} \tau$. Call S a* plane triangulation *if (i) the intersection of any pair of triangles is either empty, a common vertex, or a common edge, and (ii) the edges in S belonging to only one triangle form a simple closed polygon denoted $\partial\mathcal{D}$.*

Condition (ii) ensures that the domain is simply connected with boundary $\partial\mathcal{D}$. A piecewise-affine mapping is defined on $\mathcal{D}$ as follows.

Definition 5.4.4 *Plane Triangulation Warp. Let S be a plane triangulation with domain $\mathcal{D}$, as in Definition 5.4.3. Let $\{p_v : v \text{ a vertex of } S\}$ be a set of points in the plane such that for each triangle $ABC \in S$, points p_A, p_B, and p_C form a non-degenerate triangle in the plane. Define the plane triangulation warp $T : \mathcal{D} \to \mathbb{R}^2$ as follows. 1. If x is a vertex of S, $T(x) = p_x$. 2. Otherwise, x is located in triangle $ABC \in S$; let $T(x)$ be given by Equation 5.7, the unique affine mapping of $\triangle ABC$ to $\triangle p_A p_B p_C$.*

This mapping is also known as a *triangulation map* [Saa87]. The mapping is well-defined since the mapping of each vertex is given, and the vertex mapping determines the map for points on a triangle interior. Given point x, the triangle containing it is located using a standard planar point location algorithm, for example using a trapezoidal map [dBvKOS97]. For edges common to two triangles, Lemma 5.4.2 ensures the maps agree because each map reduces to the same affine mapping between edges. The mapping T is continuous everywhere, and Section 5.5 discusses the conditions under which it is a homeomorphism. This kind of mapping easily extends to a mapping between isomorphic simplicial complexes in higher dimensions and is used in 3D brain matching, e.g. [AAF00].

The barycentric coordinates are convenient for defining a piecewise-linear interpolation of a function on the domain of a triangulation.

Definition 5.4.5 *Plane Triangulation Interpolation. Let S be a plane triangulation with domain $\mathcal{D}$, as in Definition 5.4.3, and V be the vertex set of S. Given a set of real values at each vertex, $\{f_v \in \mathbb{R} : v \in V\}$, a continuous function $f : \mathcal{D} \to \mathbb{R}$ is obtained as follows. 1. If x is a vertex of S, then $f(x) = f_x$. 2. Otherwise, x is located in triangle $ABC \in S$; let $f(x) = \alpha f_A + \beta f_B + \gamma f_C$, where (α, β, γ) are the barycentric coordinates of x relative to triangle $\triangle ABC$.*

This definition is easily extended to range sets other than $\mathbb{R}$. For example, Equation 5.7 is precisely a plane triangulation interpolation with values in $\mathbb{R}^2$, so the plane triangulation warp is simply a special case of plane triangulation interpolation.

Triangulation Warping on the Sphere

On the sphere, triangulation warping follows the same pattern as in the plane: the spherical domain is divided into "spherical" triangles and a 1-1 mapping between corresponding triangles is used.

Let A, B, and C be three points on the sphere $\mathbb{S}^2$ that are not coplanar with the origin. Using projection from the origin, i.e. the mapping $x \mapsto x/||x||$, the planar triangle ABC is put into 1-1 correspondence with a triangular patch of $\mathbb{S}^2$ bounded by three shortest geodesics, from A to B, from B to C, and from C to A. Each geodesic bounding the patch is denoted an *edge arc*, and the triangular patch of $\mathbb{S}^2$ is known as the *spherical triangle* $\bigcirc\!\!\!\!\!\!\triangle ABC$. Given the 1-1 mapping between the planar triangle $\triangle ABC$ and the spherical triangle $\bigcirc\!\!\!\!\!\!\triangle ABC$, barycentric coordinates can be used to refer to either $q = \alpha A + \beta B + \gamma C \in \triangle ABC$ or to $p = q/||q|| \in \bigcirc\!\!\!\!\!\!\triangle ABC$.

Define a mapping from point p in spherical triangle $\bigcirc\!\!\!\!\!\!\triangle ABC$ to point $T(p)$ in spherical triangle $\bigcirc\!\!\!\!\!\!\triangle A'B'C'$ as follows. First, project p to point q on the planar triangle ABC and compute the barycentric coordinates of q using Equation 5.6. Equation 5.7 then gives a corresponding point q' on the planar triangle $A'B'C'$. Finally, q' is projected to point on the sphere, denoted p'. Define $T(p)$ to be p'. Figure 5.8 illustrates this procedure. Note that each of these maps is a homeomorphism, so the composed spherical triangle to spherical triangle mapping is a homeomorphism. Call this mapping the *barycentric spherical triangle mapping* from $\bigcirc\!\!\!\!\!\!\triangle ABC$ to $\bigcirc\!\!\!\!\!\!\triangle A'B'C'$. These observations are summarized in the following lemma.

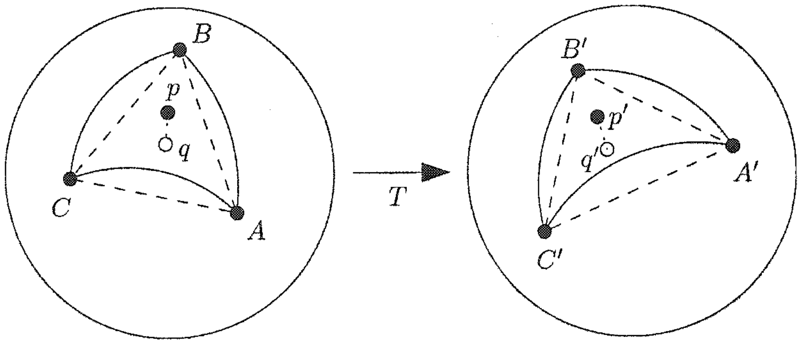


Figure 5.8: Map from spherical triangle $\bigcirc\mkern-14mu\triangle ABC$ to spherical triangle $\bigcirc\mkern-14mu\triangle A'B'C'$ is defined using the mapping between a spherical triangle and the planar triangle using the same vertices, and the affine map between planar triangles ABC and $A'B'C'$. The value of map T at p is obtained by composing maps $p \mapsto q \mapsto q' \mapsto p'$.

Lemma 5.4.6 *The barycentric spherical triangle mapping from $\bigcirc\mkern-14mu\triangle ABC$ to $\bigcirc\mkern-14mu\triangle A'B'C'$ is a homeomorphism. Interior points of $\bigcirc\mkern-14mu\triangle ABC$ are mapped to interior points of $\bigcirc\mkern-14mu\triangle A'B'C'$. The mapping on each edge arc is independent of the remaining vertex.*

The last statement follows from observing that the edge arc, say from A to B, maps to the line segment AB, and Lemma 5.4.2. The preceding method for mapping a single spherical triangle is now extended to mapping the entire sphere, employing a triangulation of the latter.

Definition 5.4.7 *Sphere Triangulation. Let S be a finite set of spherical triangles such that $\cup_{\tau \in S} \tau = \mathbb{S}^2$. Call S a* triangulation *if (i) the intersection of any pair of spherical triangles is either empty, a common vertex, or a common edge arc, and (ii) each edge arc in S belongs to exactly two spherical triangles.*

Given a sphere triangulation, a continuous mapping from the sphere to itself is obtained as follows.

Definition 5.4.8 *Sphere Triangulation Warp. Let S be a sphere triangulation. Let $\{p_v : v \text{ a vertex of } S\}$ be a set of points on $\mathbb{S}^2$ such that for each spherical triangle $\bigcirc\mkern-14mu\triangle ABC \in S$, points p_A, p_B, and p_C are not coplanar with the origin and thus comprise the vertices of a spherical triangle. Define the sphere triangulation warp $T : \mathbb{S}^2 \longrightarrow \mathbb{S}^2$ as follows. 1. If x is a vertex of S, $T(x) = p_x$. 2. Otherwise, x is located in spherical triangle $\bigcirc\mkern-14mu\triangle ABC \in S$; let $T(x)$ be given by the barycentric spherical triangle mapping of $\bigcirc\mkern-14mu\triangle ABC$ to $\bigcirc\mkern-14mu\triangle p_A p_B p_C$.*

A continuous real function on the sphere can be defined using interpolation as follows.

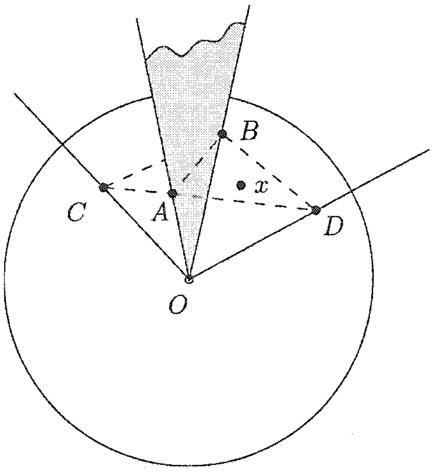


Figure 5.9: Locate spherical triangle containing $x \in \mathbb{S}^2$ by testing x against plane OAB, etc.

Definition 5.4.9 *Sphere Triangulation Interpolation. Let S be a sphere triangulation with vertex set V. Given a set of real values at each vertex, $\{f_v \in \mathbb{R} : v \in V\}$, a continuous function $f : \mathbb{S}^2 \rightarrow \mathbb{R}$ is obtained as follows. 1. If x is a vertex of S, then $f(x) = f_x$. 2. Otherwise, x is located in spherical triangle $\circledtriangle ABC \in S$; let $f(x) = \alpha f_A + \beta f_B + \gamma f_C$, where (α, β, γ) are the barycentric coordinates of the projection of x onto the planar triangle $\triangle ABC$.*

In order to implement sphere triangulation warping or sphere triangulation interpolation, an algorithm to locate the spherical triangle $\circledtriangle ABC$ containing $x \in \mathbb{S}^2$ is required. One possible algorithm is sketched here. Observe that $x \in \circledtriangle ABC$ if, and only if, x lies in the infinite triangular pyramid with apex at the origin, O, and edges passing through points A, B, and C. This property can be verified by testing the location of x with respect to the planes through points OAB, OBC, and OCA. If x lies on the correct side of all three planes, then $x \in \circledtriangle ABC$. If x lies on the opposite side of, say, plane through OAB, then x is closer to the adjacent triangle that shares edge AB than it is to $\circledtriangle ABC$. Figure 5.9 shows an example where x is on the opposite side from $\circledtriangle ABC$ of the plane through OAB and is therefore closer to $\circledtriangle ADB$. The triangle containing $x \in \mathbb{S}^2$ is obtained starting from an arbitrary triangle by testing x against the three planes associated with the triangle and updating to a triangle that is closer if x fails a plane test. This is similar to the strategy used in the Lin-Canny algorithm to find the closest features on a pair of convex polyhedra [Lin93] and its robust descendants such as [Mir98].

5.5 Injectivity

In 3D, the mesh warping functions use cubic polynomial interpolation for each subdomain (Section 2.4.2). With cubic interpolation, a spatial transformation can be noninjective in two ways [CL00]. First, a transformed subdomain may overlap another transformed subdomain. Second, the mapping of a single subdomain may itself be noninjective. The situation is somewhat easier using triangulation warps because the second situation does not occur. In the plane, mapping of each triangular subdomain is affine (Lemma 5.4.1) and thus injective. The mapping of a spherical triangle is also injective, due to Lemma 5.4.6.

The conditions under which a plane triangulation warp is injective are given by Saalfeld [Saa87]. Let S be a plane triangulation with domain $\mathcal{D}$, and let $T : \mathcal{D} \rightarrow \mathbb{R}^2$ be a plane triangulation warp. Consider a triangle of S and its image under T. If the counterclockwise ordering of vertices for a particular triangle is ABC in S and the counterclockwise ordering in the image is $T(A)T(B)T(C)$, then T is said to preserve the orientation of triangle ABC. Say that the mapping T is *orientation preserving* if it preserves the orientation of all triangles.

Theorem 5.5.1 *(Saalfeld [Saa87]) Let $T : \mathcal{D} \rightarrow \mathbb{R}^2$ be an orientation preserving triangulation mapping. If $T|_{\partial\mathcal{D}}$ is one-to-one, then T is a homeomorphism.*

Saalfeld proves this theorem by observing that the area bounded by the simple polygon $T(\partial\mathcal{D})$ is given by the sum of signed areas of triangle $T(\tau)$ for each $\tau \in S$. Since the mapping preserves the orientation of each triangle, the signed areas are all positive so the area of $T(\mathcal{D})$ is equal to the sum of absolute area of each triangle $T(\tau)$ and therefore no two triangles overlap. If T reverses the orientation of each triangle, the analogous theorem is also true.

The situation for sphere triangulation warp turns out to be more complicated because the sphere has no boundary. Meisters and Olech [MO63] prove a very general theorem on homeomorphisms between manifolds, from which a useful corollary is derived. A crucial property for a map to be injective is contained in the following definition.

Definition 5.5.2 *Locally one-to-one. A continuous mapping $T : X \rightarrow Y$ of one topological space X to another is said to be locally one-to-one at a point x provided x possesses a neighbourhood N such that $T|_N$ is a one-to-one mapping. A mapping T is said to be locally one-to-one on X provided T is locally one-to-one at each point of X.*

If T is differentiable, the classical inverse function theorem (see, e.g. [PM91]) provides a method for checking this property. Let T be a differentiable map $\mathbb{R}^n \to \mathbb{R}^n$ with continuous derivatives in an open set $U \subset \mathbb{R}^n$ and the Jacobian is nonzero at $x \in U$. Then T is locally one-to-one at x. Triangulation warping is defined piecewise and is not necessarily differentiable on the boundaries of subdomains, so the inverse function theorem cannot be used to prove local one-to-oneness.

The following form of a classical theorem, known as Brouwer's theorem on the invariance of domain, is used by Meisters and Olech [MO63].

Theorem 5.5.3 *Let X be an arbitrary subset of an n-manifold $\mathbb{M}^n$, and let T be a homeomorphism of X onto a subset $T(X)$ of an n-manifold $\mathbb{M}'^n$. Then $T(\mathit{Int}X) = \mathit{Int}T(X)$ and $T(\partial X) = \partial T(X)$.*

The following is theorem 2 of Meisters and Olech, slightly simplified to avoid some extra notation unnecessary for this work.

Theorem 5.5.4 *(Theorem 2 of [MO63]) Let X be a compact subset, with connected boundary, of an n-manifold $\mathbb{M}^n$, $n \geq 2$. Let T be a continuous mapping of X onto a proper subset $T(X)$ of an n-manifold $\mathbb{M}'^n$ such that T is locally one-to-one on X. Then if $T|_{\partial X}$ is one-to-one, T is a homeomorphism.*

The theorem of Meisters and Olech cannot be used directly to prove that a sphere triangulation mapping T is a homeomorphism because the theorem requires that the domain and its image be proper subsets of the manifold in question, i.e., the sphere. In order to apply the theorem, remove one spherical triangle τ from the domain of T and apply the theorem to the mapping T restricted to the smaller domain $X = \mathbb{S}^2 \setminus \mathrm{Int}\,\tau$. The condition that $T(X)$ is a proper subset of the sphere must be separately checked.

Let S be a sphere triangulation and $T : \mathbb{S}^2 \to \mathbb{S}^2$ be a sphere triangulation warp defined using S. Consider a spherical triangle of S and its image under T. If, viewed from outside the sphere, the counterclockwise ordering of vertices for a particular spherical triangle is ABC in S and the counterclockwise ordering in the image is $T(A)T(B)T(C)$, then T is said to preserve the orientation of spherical triangle $\bigcirc\!\!\!\!\!\triangle ABC$. Similarly, the edge arcs joining a vertex v to each of its neighbouring vertices have a particular counterclockwise cyclic ordering, as viewed from outside the sphere. If the edge arcs have the same cyclic ordering after mapping, T is said to preserve the cyclic ordering of neighbours of v. Say that the mapping T is *orientation preserving* if it preserves the orientation of all spherical triangles and also preserves the orientation of neighbours about each vertex.

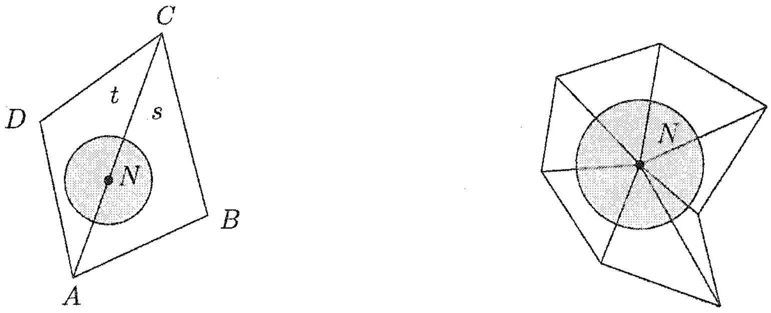


Figure 5.10: Neighbourhoods for proof of Theorem 5.5.5.

Theorem 5.5.5 *Let $T : \mathbb{S}^2 \rightarrow \mathbb{S}^2$ be an orientation preserving sphere triangulation warp. Let τ be a spherical triangle and p a point on the interior of τ. If $T(p)$ does not belong to the image $T(\tau')$ for any spherical triangle $\tau' \neq \tau$, then T is a homeomorphism.*

Proof: Let $X = \mathbb{S}^2 \backslash \operatorname{Int} \tau$, which is a compact subset of $\mathbb{S}^2$ with connected boundary $\partial\tau$. Let T' denote transformation T restricted to X. By hypothesis, $T'(X)$ does not contain $T(p)$, so $T'(X)$ is a proper subset of $\mathbb{S}^2$. The map $T|_\tau$ is a homeomorphism by Lemma 5.4.6, so $T|_{\partial\tau}$ and hence also $T'|_{\partial X}$ is one-to-one. Now argue that T' is locally one-to-one, so that Theorem 5.5.4 can be applied to T'.

Consider a point $x \in \mathbb{S}^2$. There are three possibilities: x lies on the interior of a spherical triangle, x lies on the relative interior of a spherical triangle edge arc, or x is a vertex. Argue, for each case in turn, that T' is locally one-to-one at x.

If x is on the interior of spherical triangle t, then x possesses a neighbourhood $N \subseteq t$ and $T'|_N$ is the mapping associated with t. All such triangle-to-triangle maps are homeomorphisms by Lemma 5.4.6, so $T'|_N$ is one-to-one.

In the case that x lies on the relative interior of an edge arc shared with τ, the neighbourhood in X can be chosen to lie in exactly one spherical triangle, and the argument proceeds as in the preceding case. Suppose instead that x lies on an edge arc shared by spherical triangles $s = ABC$ and $t = CDA$ (left side of Figure 5.10). Choose a neighbourhood N small enough that it contains no edge arcs other than AC . Neighbourhood N is therefore contained in $s \bigcup t$. Consider the disjoint partition $N = (N \bigcap \operatorname{Int} s) \bigoplus (N \bigcap \operatorname{Int} t) \bigoplus (N \cap AC)$. Each of $T'|_{N \bigcap \operatorname{Int} s}$, $T'|_{N \bigcap \operatorname{Int} t}$, and $T'|_{N \cap AC}$ is one-to-one. Since T' is orientation preserving, $T'(B)$ and $T'(D)$ are on opposite sides of the edge arc through $T'(A)T'(C)$. Hence $T'(\operatorname{Int} s)$ is disjoint from $T'(\operatorname{Int} t)$ and both are disjoint from $T'(AC)$. Thus $T'|_N$ is one-to-one.

The final case is that x is a vertex of the triangulation (right side of Figure 5.10).

Choose N small enough that it intersects no edge arcs of the triangulation other than those incident to x. Split N into a disjoint union of subsets (i) lying in the interior of a spherical triangle incident to x, (ii) lying in the relative interior of an edge arc incident to x, and (iii) the set $\{x\}$. As in the previous case, T' restricted to each subset of N is one-to-one. Since T' preserves cyclic ordering of the neighbours of x, the images of the subsets are pairwise disjoint. Thus $T'|_N$ is one-to-one.

Using Theorem 5.5.4, conclude that T' is a homeomorphism. It remains to show that $T'(X)$ and $T(\operatorname{Int}\tau)$ are disjoint so that the two maps can be put together to obtain a homeomorphism defined on all of $\mathbb{S}^2$, namely the mapping T.

Mapping T' is a homeomorphism on $X \subset \mathbb{S}^2$ onto a subset of $\mathbb{S}^2$. By Theorem 5.5.3, T' maps the boundary of X to the boundary of set $T'(X)$. The boundary of X is $\partial\tau$, which is mapped to the boundary of the spherical triangle $T(\tau)$. There are two open sets of $\mathbb{S}^2$ that have $\partial T(\tau)$ as boundary: the spherical triangle interior, $\operatorname{Int} T(\tau)$, and the exterior, $\mathbb{S}^2 \setminus T(\tau)$. By Theorem 5.5.3, T' maps $\operatorname{Int} X$ to one of these two sets. It cannot be the spherical triangle interior, $\operatorname{Int} T(\tau)$, since the latter contains $T(p)$ and $T(p) \notin T'(X)$ by hypothesis. Thus $\operatorname{Int} T'(X) = \mathbb{S}^2 \setminus T(\tau)$. The sets $T'(\operatorname{Int} X)$ and $T(\operatorname{Int}\tau)$ are thus disjoint.

Since both T' and $T|_\tau$ are homeomorphisms and agree on their common boundary (i.e. $\partial\tau$), the two maps can be put together to obtain a homeomorphism, namely T.

□

In contrast to the situation in the plane, the sphere triangulation mapping T must preserve not only the orientation of the spherical triangles, but also the cyclic ordering of neighbours. A simple example shows that this extra condition is necessary. Consider a triangulation of the domain sphere that comprises a vertex at each pole and $2k$ vertices ($k \geq 3$) equally spaced about the equator, with $2k$ spherical triangles in each hemisphere as shown on the left of Figure 5.11. Let T be a sphere triangulation warp for which each pole vertex is mapped to itself, while the sequence of vertices $v_1 \dots v_{2k}$ is stretched so as to wrap around the equator twice, as pictured on the right of Figure 5.11. The mapping preserves the orientation of each spherical triangle $v_A v_i v_{i+1}$, but it is clearly not a 1-1 map. The cyclic order of vertices at the poles is not preserved.

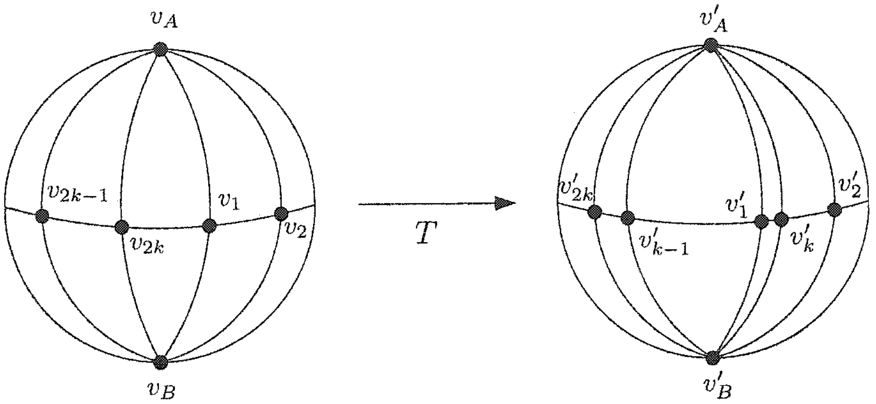


Figure 5.11: Spherical triangulation warp with $v'_i = T(v_i)$. This example shows that preserving the orientation of each spherical triangle is not sufficient to guarantee an injective spherical triangulation mapping.

5.6 Summary

In contrast to registration of 3D images, surface registration requires two extra steps, illustrated in Figure 5.1, before the transformation can be obtained. First, the cortical surface must be identified from the MR image. Second, the mapping from the surface situated in 3D to an auxiliary 2-manifold must be obtained.

As shown in Figure 5.3, there are at least three plausible manifolds that can be used as the cortical surface: the inner interface between gray matter and white matter, the outer interface between gray matter and CSF, or a surface located in the middle. In principle, any of the three can be used as the surface. For this work the inner surface is chosen, largely because the outer surface (and thus also the medial surface) is more difficult to detect on typical MR images as shown in Figure 5.4.

Biologically, each cortical hemisphere surface is a topological disc. However, the hemisphere boundaries are deep in the brain next to the corpus callosum and require manual processing to identify [DECS99]. Automatic cortical extraction software avoids identifying the hemisphere boundaries and instead produces a surface with extra, non-cortical, regions. One method splits the hemispheres and produces a topological sphere for each hemisphere that consists of cortex plus a cap through the corpus callosum. The second approach produces a single topological sphere that includes both hemispheres along with the top of the corpus callosum and a cap through the brain stem. The three alternatives are shown in Figure 5.2. The experiments presented in this thesis obtain the cortical surface using the ASP software, developed by MacDonald [Mac98]. This software operates by deforming a spherical mesh onto both hemispheres of the brain, obtaining a single-sphere model of the cortex.

As discussed in Section 5.2, the cortical surface mesh situated in $\mathbb{R}^3$ is inconvenient, so a simpler auxiliary space is used. There are three general reasons to prefer using an auxiliary space [VEDJM98]: (1) to aid visualization, (2) to facilitate a global coordinate system that allows communication of results between laboratories, and (3) to simplify the registration algorithm. It is the third reason that is important for this work. The auxiliary space topology should be compatible with the topology of the cortical surface mesh, which generally means a topological disc (or the plane) or a topological sphere is used. As mentioned above, this work uses a single topological sphere for the cortical mesh and a single sphere is therefore used as the auxiliary space.

After the native surface mesh has been generated, a mapping from the native mesh into the auxiliary space is required. There are a number of methods for doing so, described in Section 5.2. For this work the reverse-iterative deformation method is most appropriate, since the surfaces are generated by deforming an ellipsoid (easily mappable to $\mathbb{S}^2$) to the native mesh. This method obtains the mapping from mesh vertex v to $\mathbb{S}^2$ by simply recording the sphere location that corresponds to the point on the initial ellipsoid that ends up at v at the end of the surface extraction process.

Once all data from the source and target surfaces have been mapped onto $\mathbb{S}^2$, the auxiliary space, the mapping sought is $T : \mathbb{S}^2 \rightarrow \mathbb{S}^2$. This mapping is parameterized as a high-dimensional warping in order to have enough flexibility in matching different individuals, the same reasons that a mesh warping is used in 3D (see Chapter 2). For the work in this thesis, the sphere triangulation warp (Definition 5.4.8) is used.

There is nothing in the definition of triangle mesh warping that requires the mapping to be injective, just as there is nothing to require injectivity of a 3D mesh warping. In contrast to the situation in 3D, however, injectivity can be easily tested using the results of Theorem 5.5.5. This theorem ensures an injective mapping if the orientation of each triangle and the cyclic ordering of neighbours for each vertex are preserved under the mapping and, in addition, one point in some spherical triangle τ does not appear in spherical triangle $T(\tau')$ for all $\tau' \neq \tau$.

Chapter 6

Searching for an Optimal Transformation

The purpose of this chapter is to review the general surface mapping problem, mainly highlighting the differences from the corresponding 3D problem described in Chapter 3. In addition the algorithm for 2D surface registration proposed in this thesis is detailed.

Surface registration is set up as an optimization, as is done for volume registration, and thus the feature, objective function, and transformation search strategy must be chosen. As discussed in Section 5.2, the spatial transformation sought is mapping from the auxiliary space (the plane $\mathbb{R}^2$ or the sphere $\mathbb{S}^2$) to itself. In this chapter, the auxiliary space is sometimes denoted generically by $\mathbb{X}^2$. Thus, the spatial mapping is two steps removed from the input images as shown in Figure 6.1. From source image I, the native mesh is extracted (M_I) and then projected using mapping Π_I onto $\mathbb{X}^2$. Similarly, target image J gives rise to native mesh M_J which is mapped to the auxiliary surface using Π_J. Using the data from the source and target images mapped onto the auxiliary surface, $T : \mathbb{X}^2 \rightarrow \mathbb{X}^2$ is obtained by optimization.

6.1 Coordinate Invariance

As discussed in Section 5.2, the auxiliary space is introduced for computational convenience. The fundamental mapping between cortices is the mapping between the native surfaces, shown as mapping W in Figure 6.1. As long as Π_J is invertible, it is equivalent to compute mapping T or to compute mapping $W \equiv \Pi_J^{-1} \circ T \circ \Pi_I$. However, T is easier to work with since the domain is chosen to have a simple geometry such as the plane or the sphere.

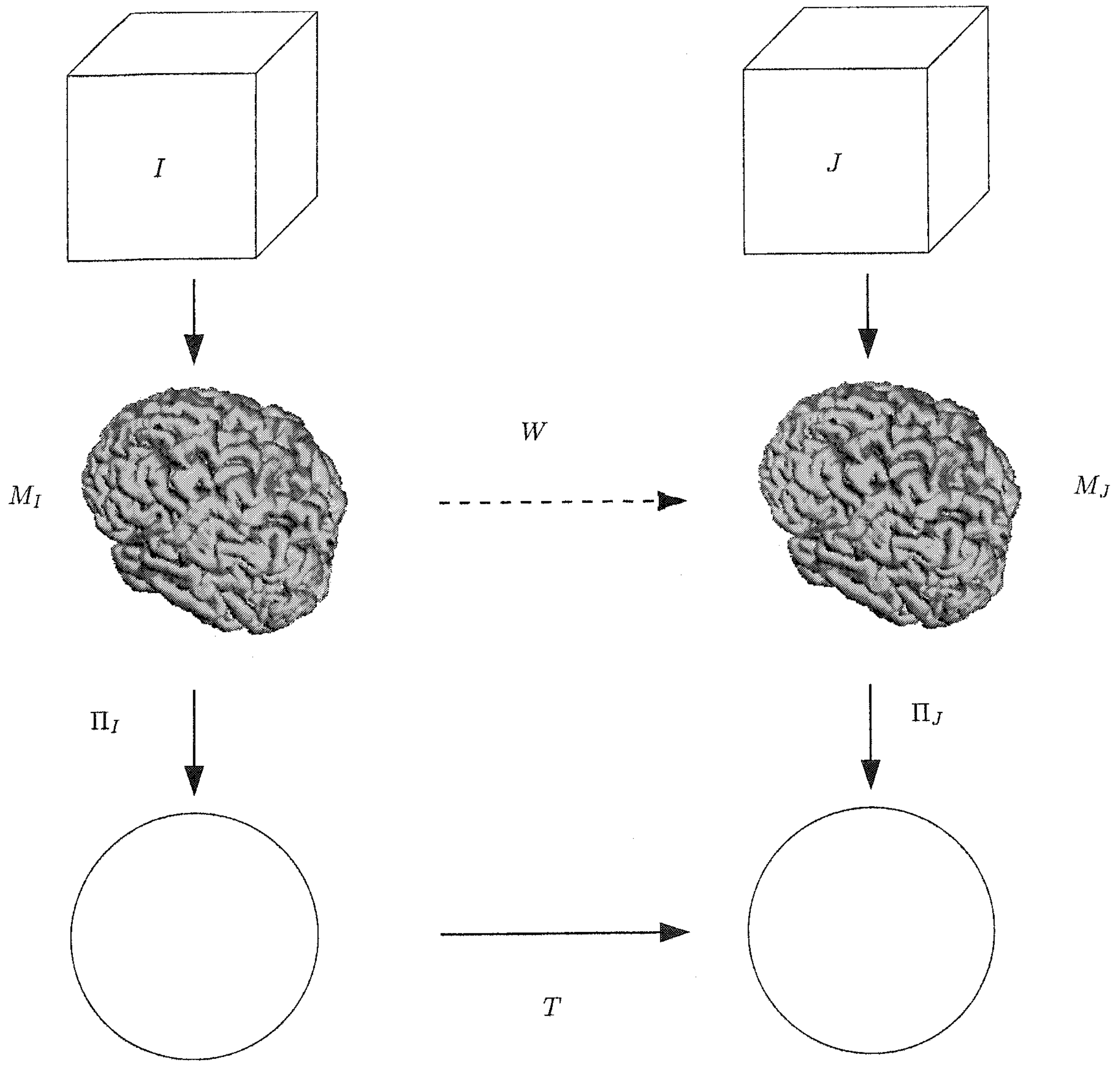


Figure 6.1: From the 3D image I the native mesh (M_I) is extracted and then mapped into the auxiliary space using Π_I. Image J undergoes analogous steps. The mapping T maps the auxiliary space to itself. Assuming Π_J is invertible, then $W \equiv \Pi_J^{-1} \circ T \circ \Pi_I$ maps the native mesh M_I to M_J.

From the point of view of coordinate invariance, it is mapping W that should be invariant under changes of coordinate system. In principle, the same map W should be obtained no matter what coordinate systems are imposed on the cortical meshes M_I and M_J. In 3D, the coordinate system is always assumed to be a right-handed Cartesian coordinate system, so only invariance under transformations between right-handed Cartesian coordinate systems is required. On the 2D surfaces, the coordinate systems are not Cartesian, so the objective function should be invariant under a general change of coordinates (any invertible function). This can be accomplished by ensuring the objective function is a scalar under general coordinate change, which in turn means that the integrand must be a "scalar density" [LR75]. For differential regularization terms, this means replacing each partial derivative by its corresponding covariant derivative, as suggested by Thompson [TT99, TWMT00].

In practice, most approaches work on the auxiliary space rather than the native mesh [Dav97, VEDJM98, FSTD99, VD99] without discussing coordinate invariance. This strategy can be viewed as considering the projections Π_I, Π_J as canonical so that T is the fundamental mapping. This is the point of view adopted in this thesis. It is the variational problem for T on the auxiliary space (rather than for W on M_I) that is required to be invariant under coordinate changes. For the plane, as in 3D, Cartesian coordinates are assumed and invariance under rigid-body transformations is required. For the sphere, invariance under rotations of the sphere is required.

6.2 Data Terms

Data terms can be categorized, as was done for 3D image registration, depending on whether a partial matching is given or not, i.e. whether the features are labelled or unlabelled.

6.2.1 Labelled Features

Both interpolating and approximating transformations driven by labelled features are possible, as is done in 3D. The only change is to use distance measured in the auxiliary space for the data term. Landmark matching can be done using the plane [VEDJM98] or using the sphere [BJM99] as the auxiliary space. The data term is

$$\Phi_{LM} = \sum_{k=1}^{n} \alpha_k d^2(T(p^k), q^k), \qquad (6.1)$$

where d is the metric function for the auxiliary space, $\{(p^k, q^k)\}$ is the set of matched point landmarks, and α_k is a weighting reflecting the confidence of localizing pair k. This is the 2D analogue of Equation 3.5.

6.2.2 Unlabelled Features

As with 3D matching, unlabelled features can be geometric objects or a continuous function defined on $\mathbb{X}^2$. In 3D, the latter was typically the image intensity function or its gradient magnitude. Since the surface is assumed to represent an iso-intensity surface of the original data (Section 5.1), the intensity values are not useful for surface matching. Some other feature must be obtained, and the only data available is the geometry of the native surface mesh. Three possible features are described below.

In each case, the feature value for the source brain is computed and stored for each vertex of the projection of mesh M_I into the auxiliary space $\mathbb{X}^2$, i.e. $\Pi_I(M_I)$. The continuous function over $\mathbb{X}^2$ is then obtained using triangulation interpolation described in Section 5.4.2 (Definition 5.4.5 or Definition 5.4.9). The feature value is thus a continuous function denoted as $S : \mathbb{X}^2 \to \mathbb{R}$ for the source (template) surface and as function $R : \mathbb{X}^2 \to \mathbb{R}$ for the target surface. For convenience, define $\widetilde{R} = R \circ T$.

Geometric Features

Surfaces can be represented and matched just using point features, or points along with extra information such as surface normal, curvature estimates [FA96, LBS99]. The data term used is a minimum-distance term similar to Φ_{ICP} (Equation 3.7). Curves [BGKM98] and sulcal regions can also be identified on the surface mesh [RHP00, RHXP02]. Either could potentially be used as a feature.

Mean Surface Curvature

The surface curvature reflects the type of folding (e.g. sulcal or gyral) at each surface point, so surface curvature is a reasonable feature for matching. There are several related quantities that have "curvature" in the name. The following definitions are standard and can be found in any textbook on differential geometry (e.g. [Gug63]). The curvature of a plane curve is explained first.

Let $\mathcal{C}$ be a curve in the plane with *arc-length parameterization* $x(s)$, that is $||x'(s)|| = 1$, where the prime denotes differentiation of x with respect to its parameter. Let point $P \in \mathcal{C}$ have parameter value s_P, i.e. $P = x(s_P)$. Define the unit-length tangent vector $t = x'(s_P)$ and a unit-length vector n perpendicular to t,

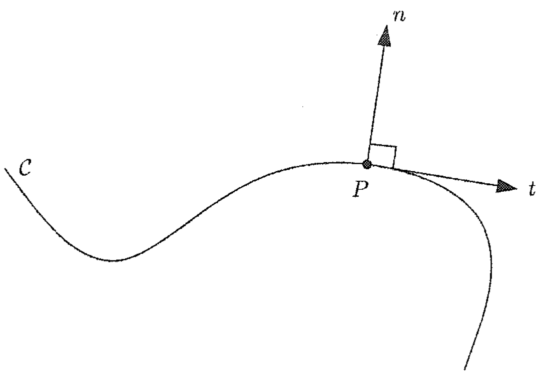


Figure 6.2: Plane curve $\mathcal{C}$ defined by arc-length parameterization $x(s)$ with $x(0) = P$, $x'(0) = t$, has curvature κ_P at P defined by relation $x''(0) = \kappa_P n$, where n is a unit-length vector perpendicular to t. The curvature at P in this figure is negative.

as illustrated in Figure 6.2. The curvature at P, κ_P, is defined by the relationship

$$x''(s_P) = \kappa_P n. \tag{6.2}$$

Note that there are two unit vectors perpendicular to t and the sign of κ_P depends on which vector is chosen to be the normal vector, n.

While a planar curve has a unique tangent vector, the situation at a point P on a surface $\mathcal{S}$ is complicated by the fact that the tangent to the surface is not unique. Indeed, a curvature can be defined for each tangent vector as follows. Let n be a unit-length normal to the surface at P, and t be a unit-length vector tangent to $\mathcal{S}$ at point P. The three points $(P, P + t, P + n)$ define a plane Π which is perpendicular to the surface at P. The intersection of Π with $\mathcal{S}$ defines a curve $\mathcal{C}$, and the *normal curvature* (some authors use the term *directional curvature*) at P in the direction t, denoted $\kappa_P(t)$, is defined as the curvature at P of the planar curve $\mathcal{C}$. The normal curvature can be calculated using an arc-length parameterization of $\mathcal{C}$ and Equation 6.2.

The normal curvature at a point is either constant as a function of t (e.g. for all tangent vectors, the normal curvature is zero if $\mathcal{S}$ is a plane) or varies between a maximum and minimum curvature. Let κ_P^1 and κ_P^2 be the two extreme values of the normal curvature at P. With no loss of generality, choose $|\kappa_P^1| \geq |\kappa_P^2|$. The values κ_P^1 and κ_P^2 are referred to as the *principal curvatures* at P. The corresponding tangent directions are always perpendicular. In Figure 6.3, for example, tangent vector t points approximately in the direction of largest principal curvature while the orthogonal direction in the tangent plane is the direction of smallest principal

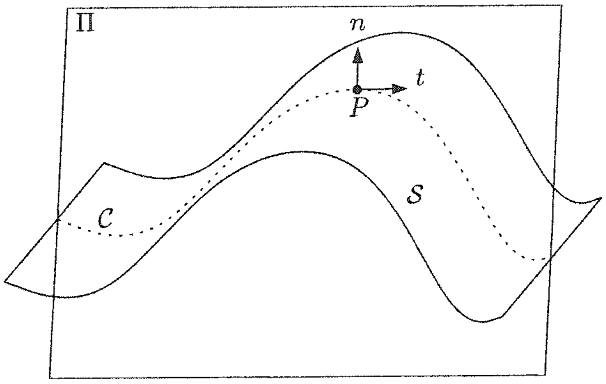


Figure 6.3: Normal curvature $\kappa_P(t)$ of surface at point P in direction t defined by arc-length parameterization of plane curve C, $x(s)$, with $x(0) = P$, $x'(0) = t$, and $x''(0) = \kappa_P(t)n$. The curvature $\kappa_P(t)$ in this figure is negative.

curvature (approximately zero in this figure). The *mean curvature* at P is defined as the mean of the normal curvatures, averaged over all tangent directions. The mean curvature has a simple formula,

$$\kappa_P^m = \frac{1}{2}(\kappa_P^1 + \kappa_P^2). \tag{6.3}$$

As with plane curves, the sign of normal curvature depends on the normal vector chosen. For the cortex, being consistently orientable, define the curvature with respect to the outward normal. With this convention, the sign of κ_P^1 (and hence also of κ_P^m, since $|\kappa_P^1| \geq |\kappa_P^2|$) at a sulcus fundus is positive, whereas the sign of κ_P^1 is negative on a gyral crown. Figure 6.4 shows the value of κ_P^m evaluated (using [Tau95]) at each vertex of the cortical surface.

The pattern of sulci and gyri is evidently reflected in the structure of Figure 6.4. Since sulcal fundi and gyral crowns have mean curvature of opposite sign, registration based on matching mean curvature should tend to match sulcus to sulcus and gyrus to gyrus. However, there is evidently a lot of noise in the curvature map as well, due to small-scale oscillations in the mesh vertex positions, so the data needs to be smoothed when used for registration [AF99b].

Curvature (or a thresholded version [Dav97]) can be used directly in a registration algorithm for matching cortical surfaces. Alternatively, the curvature can be used to guide manual delineation of features. Sulcal curves for landmark matching, for example, can be obtained by tracing the sulcus on a curvature map similar to Figure 6.4 [VD99, Dav96, Dav97].

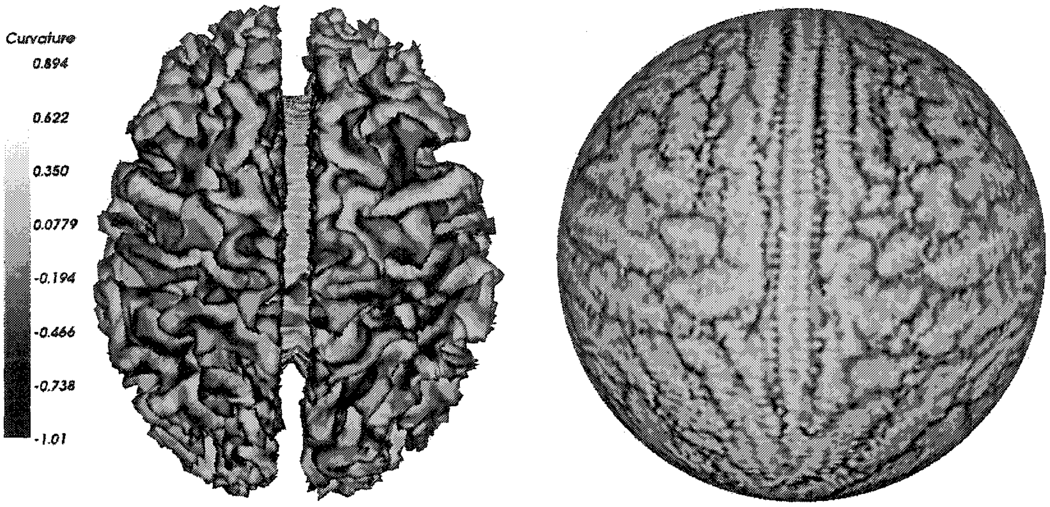


Figure 6.4: Mean curvature, evaluated (using [Tau95]) on typical cortical surface (gray matter / white matter interface). Left shows the curvature values on the folded cortex. Right shows the values mapped onto the unit sphere. Both views are from the top.

Average Convexity Feature

Fischl et al. [FSD99] describe a method of "inflating" a cortical surface that is essentially a mesh smoothing process such that the sulci open up, enabling visualization of functional activity occurring within sulci. The idea is to generate a smoother version of the surface while minimizing the change in distance between vertices. Let (V, E) denote the vertex and edge set of the surface mesh, $x(v) \in \mathbb{R}^3$ denote the position of vertex v, and d_{uv} denote the distances between vertices u and v in the initial (unsmoothed) mesh. The inflated surface is obtained by minimizing the following function

$$J_s = \frac{1}{|V|} \sum_{(u,v)\in E} ||x(u) - x(v)||^2 + \frac{\lambda_d}{4|V|} \sum_{v\in V} \sum_{u\in N_v} (||x(u) - x(v)|| - d_{uv})^2$$

over the positions $\{x(v) : v \in V\}$, where $|V|$ is the size of the vertex set, and N_v is an unspecified neighbourhood of v that should contain a "significant section of the manifold" rather than be just the 1-ring neighbourhood [FSD99]. A value of 0.1 is suggested for λ_d. The minimization of J_s is stopped when the average distance of neighbours to the tangent plane at v drops below some threshold for all v. The first term in J_s tends to pull v towards the centroid of its 1-ring neighbours, thus smoothing the mesh. The second term strives to keep $||x(u) - x(v)||$ near its value in the starting mesh.

At each iteration of the minimization, each vertex is displaced by some amount which is resolved into components tangent to and normal to the surface (at that iteration). Fishl et al. point out that vertices with positive mean curvature (i.e. sulcal fundi) tend to move outwards due to the smoothing term, while vertices with negative mean curvature (gyral crowns) tend to move inwards. Summing the (signed) displacements in the direction normal to the surface provides a measure of how "sulcal-like" or "gyrus-like" the surface is at v, a measure they dub "average convexity": a vertex that consistently moves outwards will have a large positive average convexity value, whereas a vertex that consistently moves inwards will have a large negative value.

The intersubject surface registration of Fischl et al. [FSTD99] uses squared difference of average convexity as the data term. They observe that the average convexity value "is useful for quantifying the folding pattern of a surface, as it captures large-scale geometric features, while being relatively insensitive to the small folds that typically occur on the banks of a sulcus. This is in contrast to mean curvature which attains equally high values for small secondary and tertiary folds in a surface as for the primary folds."

The average convexity, however, is dependent not just on the original surface geometry but also on the details of the smoothing algorithm, e.g. the definition of the neighbourhood N_v and the stopping conditions. The next section describes a feature value that is independent of smoothing algorithm details.

Crown Distance Transform

The intuition behind this feature, like the mean curvature and average convexity features, is that it is desirable to match points along the crown of a gyrus on the source surface with points along the crown of a gyrus in the target surface. Similarly, the fundus of a sulcus should match the fundus of a sulcus. In contrast to matching by mean curvature or by average convexity, other points on the bank of a sulcus are matched according to their fractional distance towards the bottom of the sulcus, e.g. a vertex halfway down the sulcus in the source mesh should match a point halfway down the target sulcus. Throughout this section, distances are measured on the native meshes M_I and M_J.

Suppose v is located in a sulcus of the source mesh. Consider two shortest geodesic paths, one from v to the gyral crown and the other from v to the fundus. Let the length of the first of these paths be $S(v)$ and let $S_D(v)$ be the sum of the

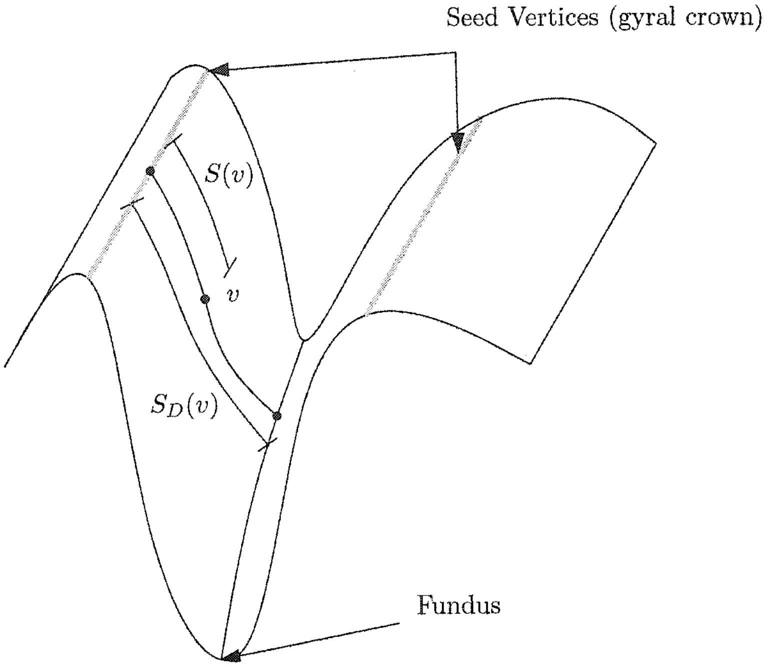


Figure 6.5: Vertices labelled as gyral points are given value zero. The remaining vertices are given a value equal to the (approximate) geodesic distance transform from the gyral crown points. Matching is based on the fractional depth $S(v)/S_D(v)$.

two path lengths. Distance $S_D(v)$ can be regarded as the depth of the sulcus along a path through v. Let R and R_D be the analogous distance functions for the target surface. The fractional depth of vertex v on the source mesh is $S(v)/S_D(v)$ as shown in Figure 6.5. The desired matching is to a point $T(v)$ on the target surface such that $S(v)/S_D(v) \approx R(T(v))/R_D(T(v))$. This relation can be expressed as (recall $\widetilde{R} = R \circ T$)

$$\widetilde{R}(x) = \alpha(x) + \beta(x)S(x) + \epsilon(x), \tag{6.4}$$

where ϵ represents random noise, $\beta(x) = R_D(T(x))/S_D(x)$, and α should be zero. However, α is left free to compensate for practical difficulties in computing $S(x)$ and $R(x)$. Note that points near to v should have depth values close to $S_D(v)$ and similarly points near $T(v)$ should have depth values close to $R_D(T(v))$. Thus, assume that α and β are slowly-varying functions and, as discussed in Section 3.2.2, the maximum likelihood estimate for T corresponds to maximizing the regional correlation coefficient, $\Phi_{CC}(S, R, T)$ given by Equation 3.13. If desired, the parameter α in Equation 6.4 could be forced to zero by using the normalized cross correlation, Φ_{XC} (Equation 3.14).

For the computation of Φ_{CC}, it is only the gyral crown distance transforms, S

and R that appear and not S_D and R_D. Thus, the feature value stored at each vertex is the geodesic *distance transform* from a set of gyral "seed" vertices, i.e. the length of the shortest geodesic path from v to a vertex in the seed set. Computation of a geodesic distance transform is addressed shortly, after considering how to identify the seed vertices. The vertices chosen to be the seeds should in principle be all vertices located on all the gyral crowns.

Existing methods used to locate such gyral vertices of a surface mesh employ image processing techniques. Rettmann et al. [RHXP02] start with a binary edge map image that is produced by assigning the value 1 to any voxel that intersects the cortical surface mesh. Then a deformable mesh (as discussed in Section 5.1) with a high tension force that makes it resistant to bending is used to seek out the boundaries of the edge map. The high tension force prevents the deformable surface from penetrating deeply into the sulci, but sometimes causes the surface to break through gyral regions so the implementation is modified to add an extra barrier force [RHXP02]. Any vertex on the original cortical surface that lies within 2 mm of the second deformable surface is deemed to be gyral. Cachia et al. [CMR+03] start with the white matter binary mask, i.e. an image that is 1 at each white matter voxel and zero otherwise. They apply a 3D morphological closing to fill in the sulcal regions and then perform a 3D erosion of the closed mask that changes any voxel within 5 mm of the mask boundary to the background value. Any mesh vertex lying outside the closed, eroded mask is selected to be a gyral crown vertex.

Other approaches do not employ image processing techniques. For example, one way to identify gyral vertices is to compute the mean curvature, then smooth and threshold it. As discussed previously, gyral vertices have negative mean curvature and so can be identified on that basis. However, negative mean curvature is not exclusive to gyral crown points: other bumps deep inside a sulcus can have negative mean curvature so a different characterization is sought.

The stiff deformable surface approach of Rettmann et al. [RHXP02] produces a surface that can be considered a "relaxed" or "slack" version of the convex hull. As Rettmann et al. observe [RHXP02], the convex hull alone generates too few gyral vertices. The problem is illustrated on the left side of Figure 6.6, where points C and D, which ought to be labelled gyral, are not part of the convex hull. Figure 6.7 shows the convex hull vertices of a typical cortical surface. There are relative few such vertices and certainly do not outline gyral crowns well.

The α-shape [EM94] is a discrete geometry notion that can be considered as a relaxation of the convex hull, with the degree of relaxation controlled by the para-

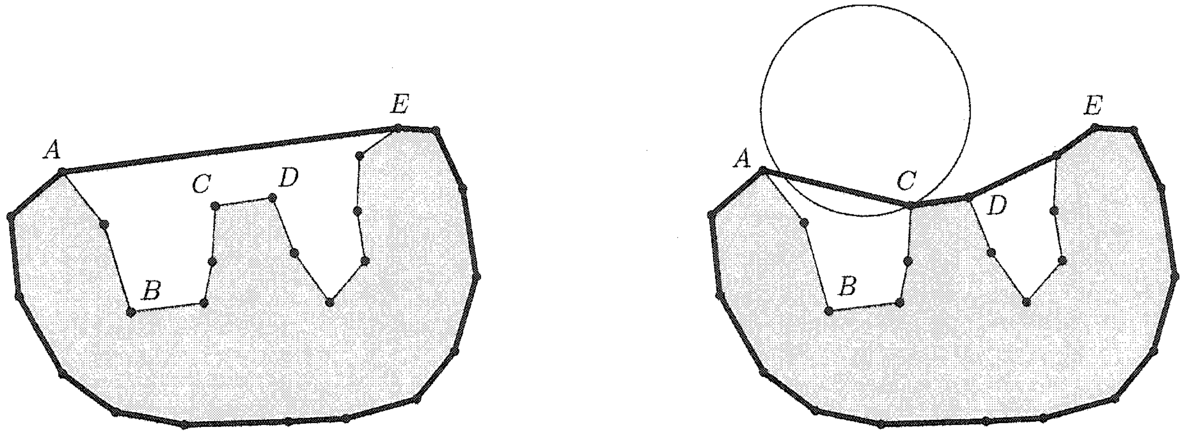


Figure 6.6: Left figure shows convex hull, which does not include points C and D on its boundary. The α-shape on the right includes C because there is an α-ball with C on the boundary and no points on the interior. However, α is large enough that B is still excluded from the boundary.

meter α. Let S be a set of points in $\mathbb{R}^3$. Define an *α-ball*[1] to be a closed ball of radius $\sqrt{\alpha}$, i.e. the point set $\{x \in \mathbb{R}^3 : ||x - p|| \leq \sqrt{\alpha}\}$, for some $p \in \mathbb{R}^3$. The points of S that form the vertices of the α-shape of S are those that lie on the boundary of an α-ball with no point of S on the interior of the α-ball. Such vertices are said to be *α-exposed*. Note that convex hull vertices are always α-exposed, since for every point x on the boundary of a convex set, there exists a ball of radius $\sqrt{\alpha}$ for any value of α that intersects the convex set only at x. Thus the vertices of the α-shape with $\alpha = \infty$ are the vertices of the convex hull. As α is decreased, more points become α-exposed until at $\alpha = 0$ all points are α-exposed. On the right side of Figure 6.6, α has been selected so that C is now α-exposed, but α is still large enough to prevent sulcal vertices such as B from being α-exposed.

Using α-balls of radius 10 mm (i.e. $\alpha = 100$), a value empirically chosen using visual inspection of the results, gives the pattern shown in Figure 6.8. Here there are good outlines of several major gyri. However, there are also vertices in the point set lying underneath the brain on the arbitrary "cap" through the brain stem (described in Section 5.1). These should not form part of the seed set, so they are masked out as follows. The distance transform is computed using the convex hull points as the seed vertices and any vertex with a distance transform value greater than 35 (a value chosen empirically) is removed from the set of vertices of the α-shape. The

[1] The definition given here follows the Computational Geometry Algorithms Library (CGAL) [CGA03], as the implementation of α-shape provided by CGAL is used. In the original definition of Edelsbrunner and Mücke, α is the radius of the α-ball.

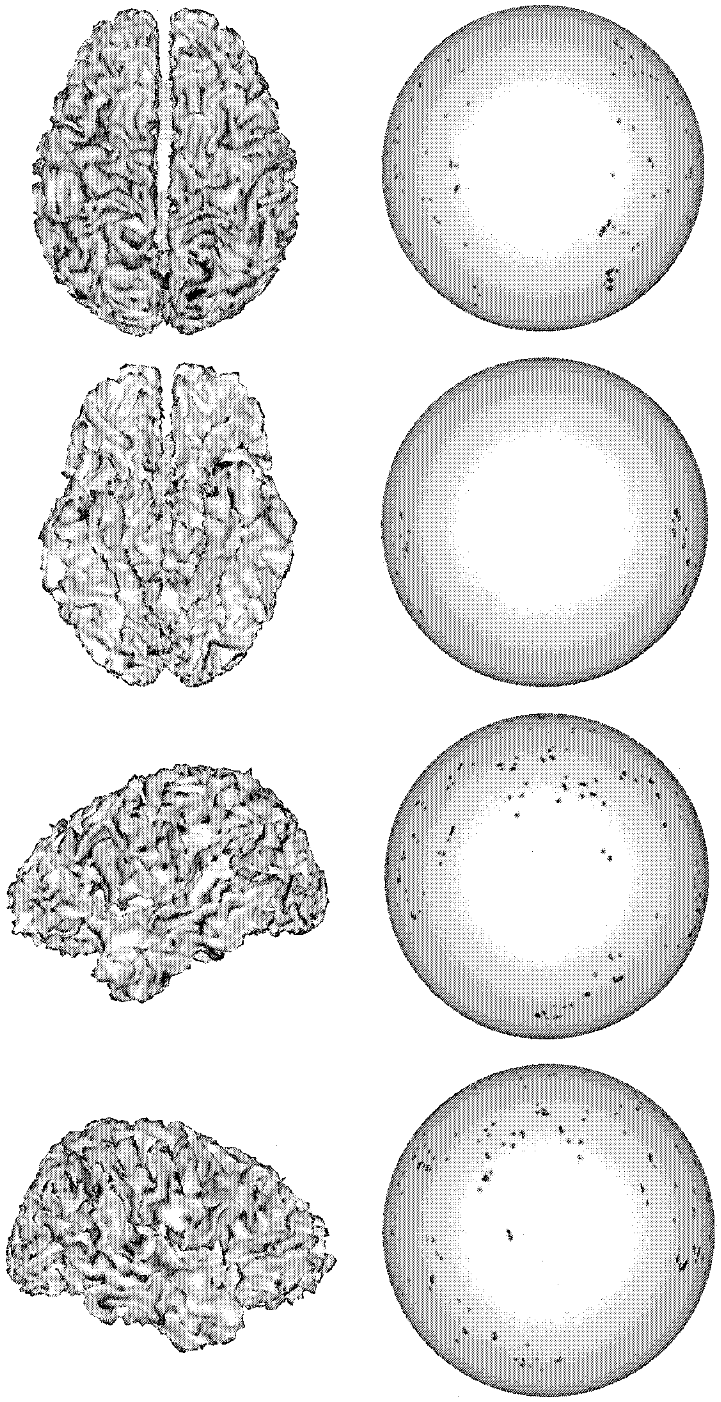


Figure 6.7: Left column shows convex hull points of the cortical surface in red. Right column shows corresponding view on $\mathbb{S}^2$. The convex hull points are distributed very sparsely across the surface, so are not representative of gyral crowns.

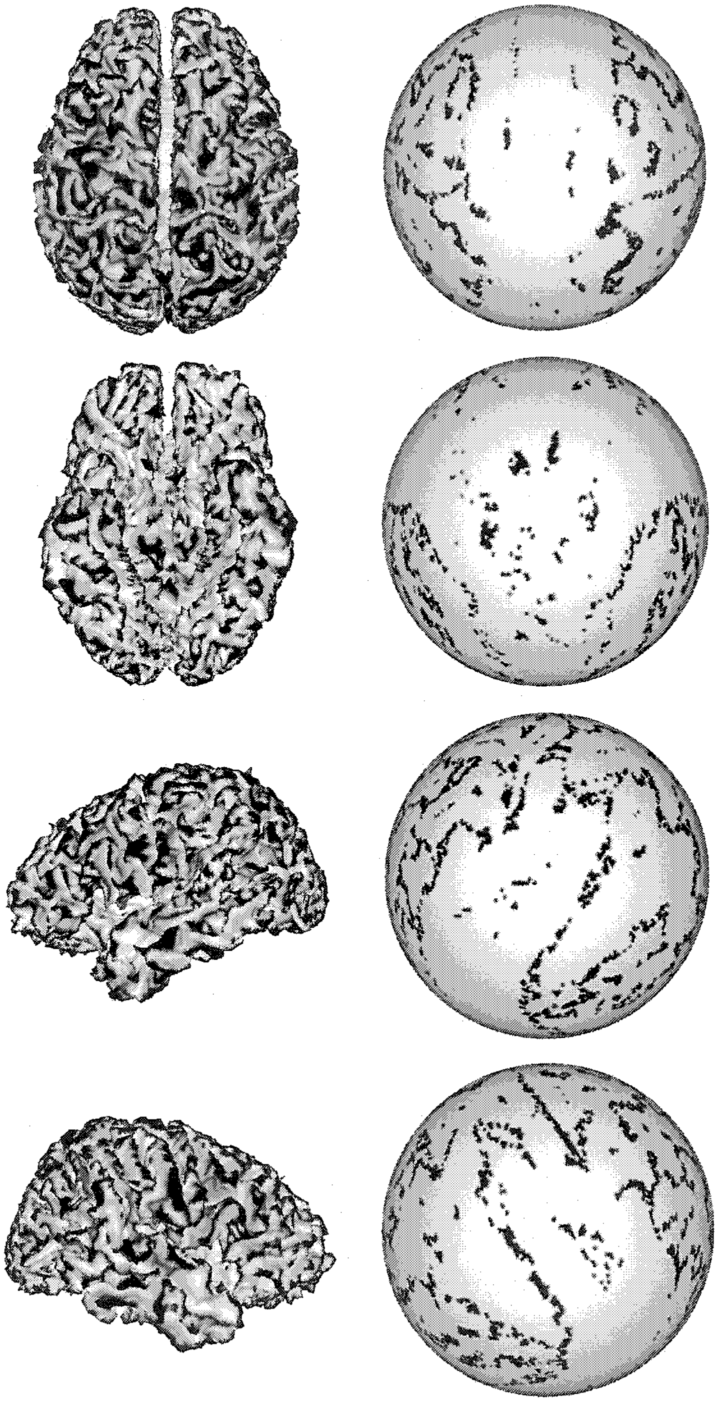


Figure 6.8: Vertices of α-shape ($\alpha = 100$). The major gyri are outlined reasonably well, but there are also points in the unphysical "cap" through the brain stem (second row), which are not desirable.

remaining vertices form the seed set used for the crown distance transform feature, shown in Figure 6.9.

The crown distance transform from the final seed set is illustrated in Figure 6.10.

Geodesic Distance Transform

The distance transform used in practice is an approximation to the geodesic distance, as computing true geodesic distance is computationally costly and difficult to code robustly [LMS01]. The approximation is computed using the mesh graph with the weight for each edge set to the Euclidean length of the edge. An additional vertex, s, is inserted and attached to each seed vertex with a zero weight edge. Then Dijkstra's single-source shortest path algorithm [CLR90] is run with s as the source vertex. To improve the accuracy of the approximation the mesh is refined before running Dijkstra's shortest path algorithm. First, m equally-space extra points are inserted on each edge, then a new edge is inserted for each pair of extra points that either (i) are adjacent on an original edge, or (ii) are on different edges both of which are incident to a common (original) facet. The approximate shortest path computed by this algorithm consists of line segments which are either an edge of the original surface, or cross a facet of the original surface as illustrated in Figure 6.11. The set of original surface edges encountered along the path is important for the error bound.

Lanthier et al. [LMS01] prove that the approximate path length is bounded by the true path length plus $kL/(m+1)$ where L is the longest edge of the *original* surface and k is the number of segments comprising the approximate path. In their paper, Lanthier et al. note that this is a pessimistic bound. A better estimate of the error, assuming that the source and terminal vertices of the path are vertices of the original surface (which is the case for the distance transform computation) is

$$\frac{1}{m+1}\sum_{j=1}^{k} l_j,$$

where l_j is the length of the j^{th} original-surface edge encountered along the approximate path. This is a better estimate because, as shown in Figure 6.12, most of the edges in a typical cortical surface are much smaller than the largest edge.

In order to select a value for m, an arbitrary cortical surface is chosen and the distance transform is computed with $m = 0, 1, \ldots, 7$ at which point the storage

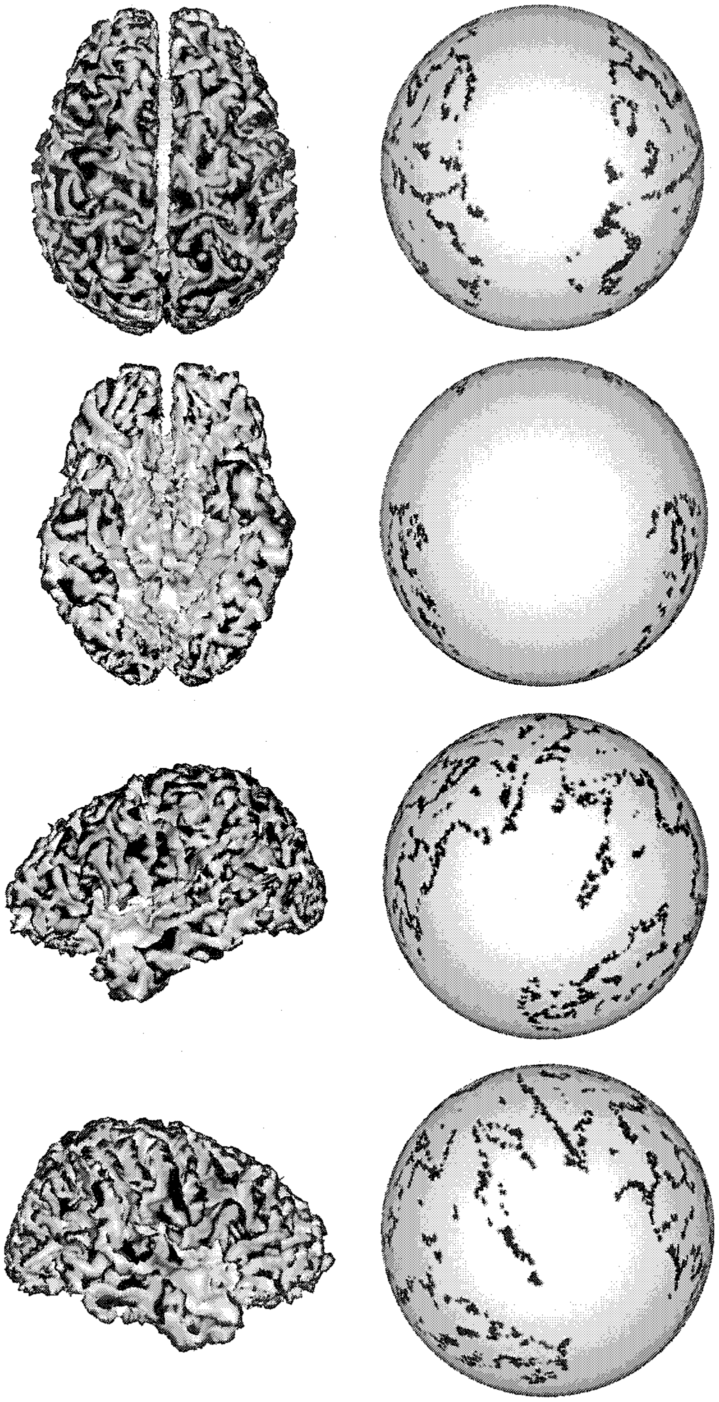
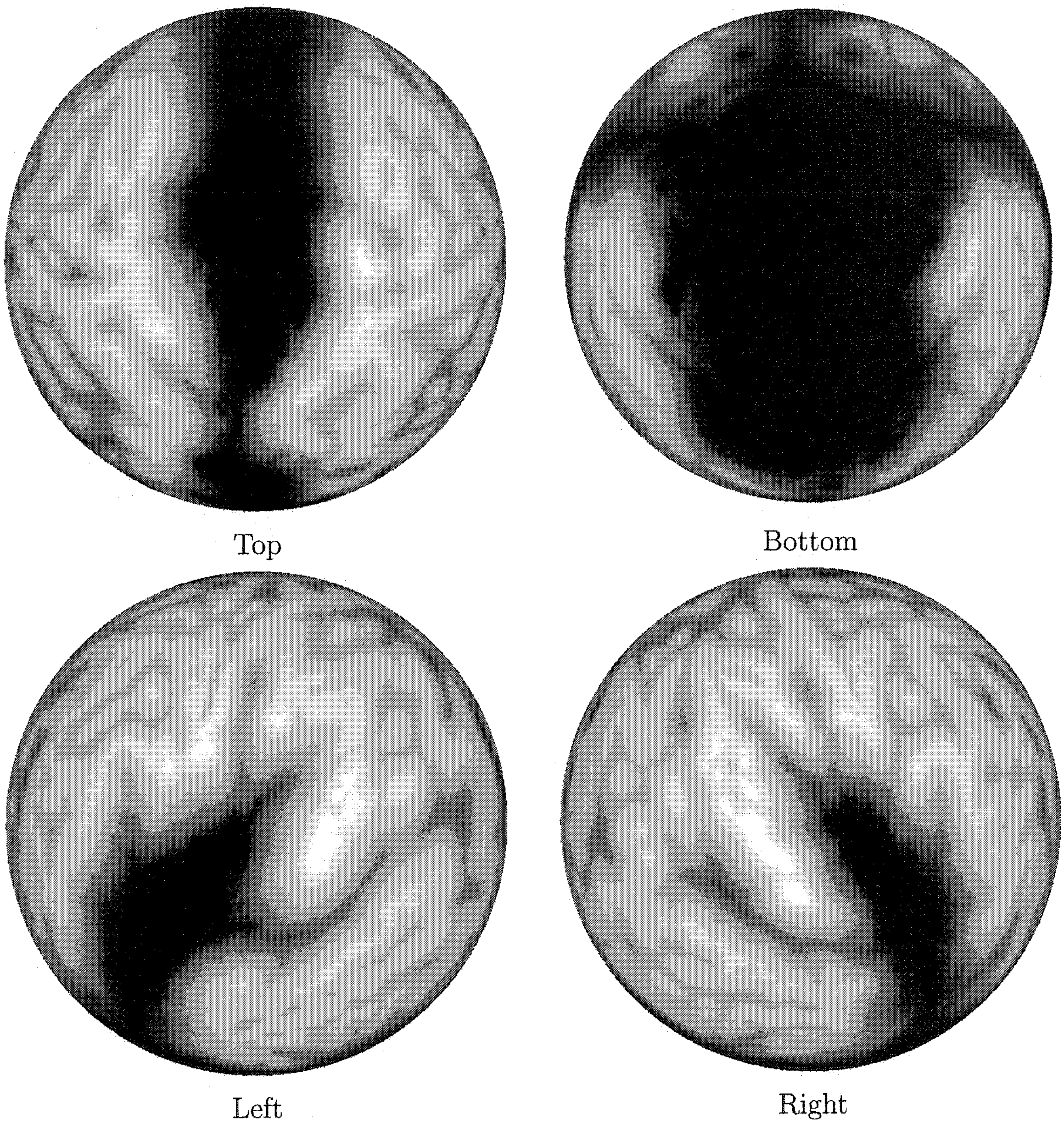


Figure 6.9: Vertices of α-shape ($\alpha = 100$), with deep points removed. The unphysical points of the brain stem cap have been removed. Compare the second row with the second row of Figure 6.8.



Top

Bottom

Left

Right

Figure 6.10: Gyral crown distance transform mapped onto $\mathbb{S}^2$. The intensity scale runs white (0 mm) to black (40 mm or more), so the gyral regions are shown in white and sulcal in black. Due to the masking out of deep vertices, most vertices in the bottom view have distances 40 mm or greater and so appear black.

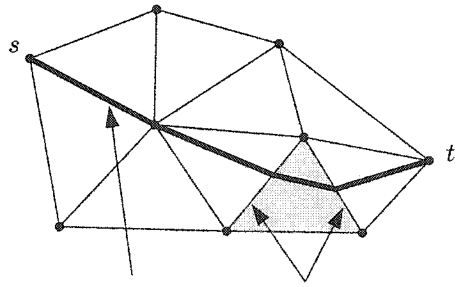


Figure 6.11: Heavy line shows shortest path from s to t on a polyhedral surface (only part of which is shown). The path consists of line segments that either follow an edge (first segment) or cross a facet (shaded). The path encounters some edges of the surface (indicated by arrows). The extra points and extra edges are not shown.

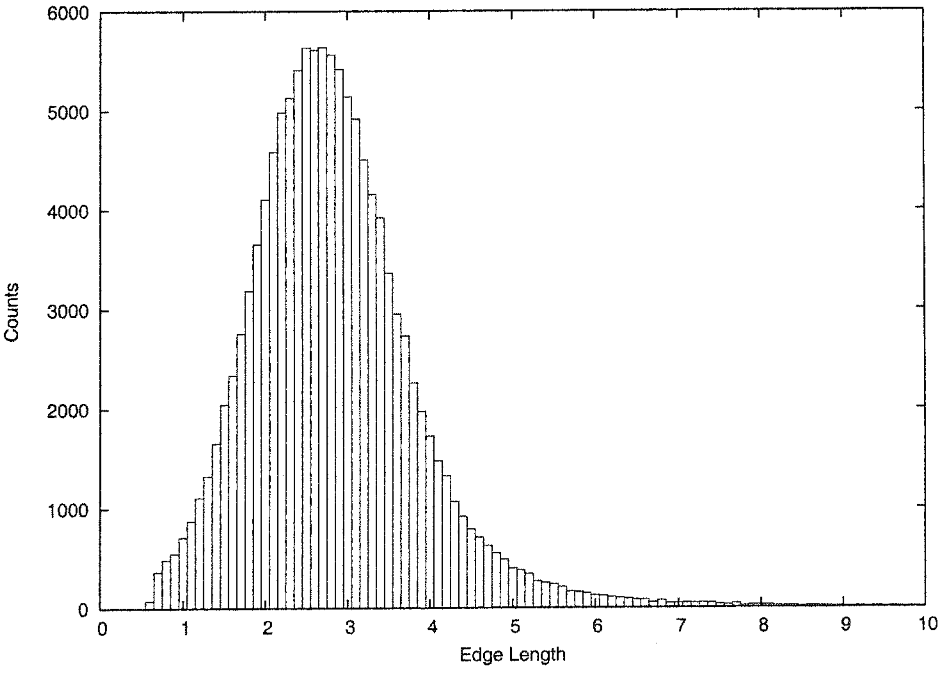


Figure 6.12: Histogram of edge lengths (122880 edges) of a typical cortical surface. 86 edge lengths were > 10 (largest edge has length 18.2). Note that the vast majority of the edge have length < 5 (27% of the maximum length) so the error bound based on the largest edge length is likely to be an overestimate.

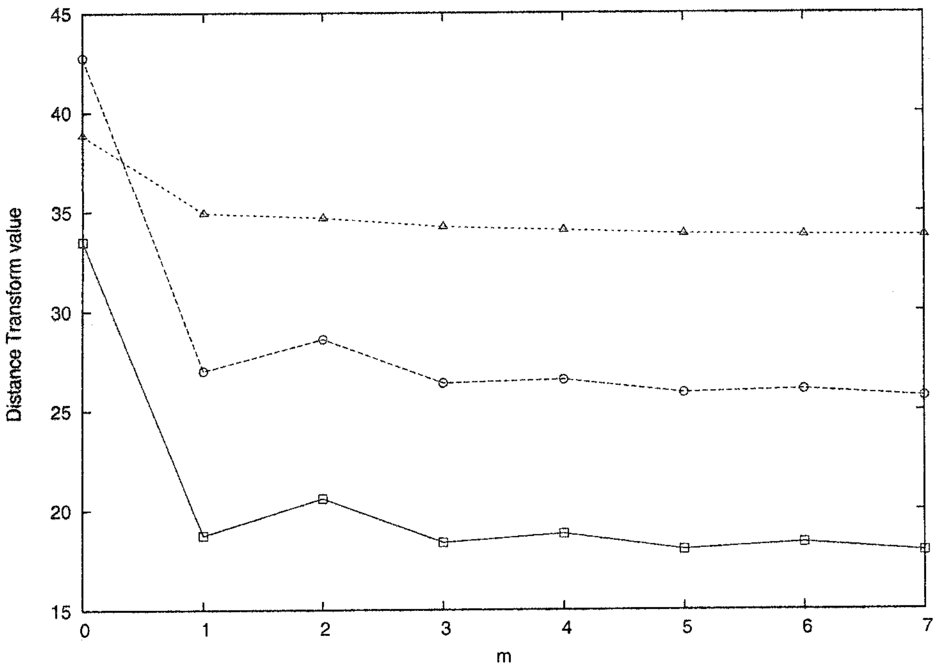


Figure 6.13: Plot of distance transform values for three vertices, as a function of m, the number of extra points inserted on each edge. One vertex was chosen because the difference between $m = 0$ and $m = 7$ was largest. Another had the largest difference between values for $m = 6$ and $m = 7$. The third was chosen arbitrarily. The values converge after about $m = 4$, so using 4 or more extra points should produce reliable distance transform values.

requirement (>1GB) exceeds the machine memory. Figure 6.13 shows that the values converge after about $m = 4$ or 5, which is in agreement with the observations of Lanthier et al. The value $m = 5$ is used for the experiments in this thesis. Figure 6.14 shows that the accuracy lost by stopping at $m = 5$ rather than $m = 7$ is generally less than 0.2, which is on the order of 1% of typical distance transform values. Moreover, the error bound grows with the path length k, so errors for vertices with small distance transform values are expected to on the low end of Figure 6.14 while large errors will be associated with large distance transform values. The relative accuracy is thus expected not to be correlated with distance transform values.

6.2.3 Multiple Features

As with 3D normalization, it is possible to use both labelled and unlabelled features. For example, Davatzikos [Dav97] uses a feature derived from thresholding three curvature values (the two principal curvatures and Gaussian curvature) at

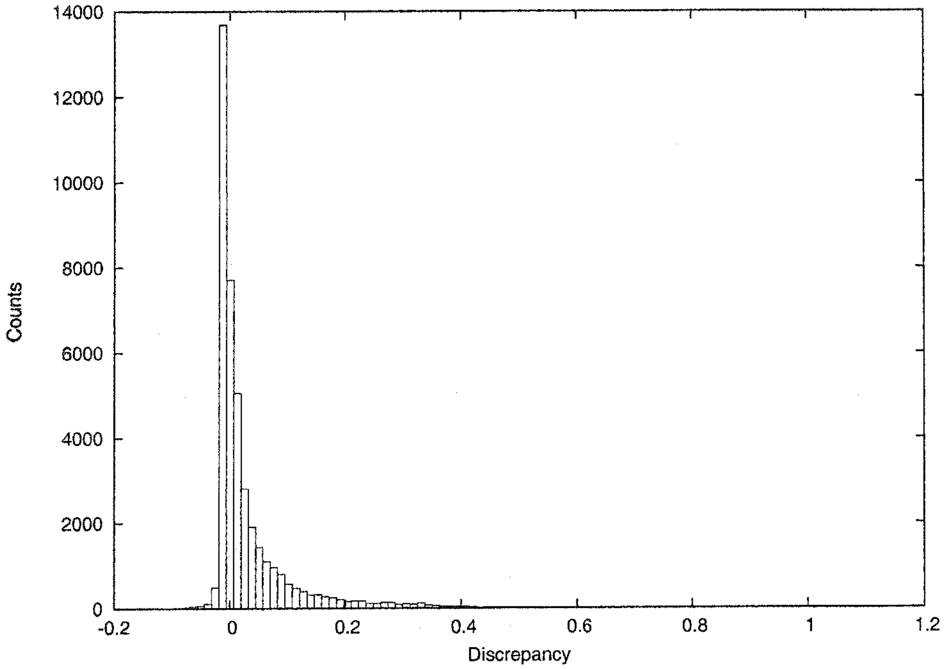


Figure 6.14: Histogram of distance transform difference between $m = 5$ and $m = 7$. Most values are < 0.2, which can be taken as indicative of the typical error made in the path length when computed using $m = 5$.

each vertex of the control mesh in addition to a landmark term Φ_{LM} (Equation 6.1) of points on matching sulci.

6.3 Model Terms

6.3.1 Regularization Rationale

The rationale is the same as described in Section 3.3.1 namely to impose prior knowledge of the solution characteristics on T, either as a specific distribution $\mathsf{P}(T)$ or a generic "smoothness" requirement using a differential regularizer.

6.3.2 Low-Order Differential Regularizers

The low-order regularizers discussed in Section 3.3.2 can be adapted by changing the partial derivatives to covariant partial derivatives in order that the result be invariant under a general change of coordinates as discussed in Section 6.1 [LR75]. This approach is not pursued here; rather the regularization is implemented as a smoothing operation of a two-step registration method (Section 6.4.2).

6.4 Iterative Registration

In general, the algorithms for 3D (Section 3.4) carry over with minimal modifications, e.g. derivatives should be covariant and distances should be measured on the appropriate 2-manifold rather than in $\mathbb{R}^3$.

6.4.1 Basic Algorithm

The generic algorithm for surface mapping is analogous to Algorithm 1. As in 3D, the idea is to iteratively estimate the parameters of the transformation T.

Landmark Matching

Algorithms used in 3D, e.g. [JM00] can be easily adapted to the plane auxiliary space [VEDJM98]. Landmark matching on the sphere auxiliary space is also used [BJM99].

Discretized Euler-Lagrange Equations

Elastic matching can be done using the Euler-Lagrange Equation 3.23 in the auxiliary space [Dav97]. The Euler-Lagrange equations can also be applied to deform the auxiliary sphere in $\mathbb{R}^3$, projecting the deformed surface back to $\mathbb{S}^2$ after each iteration [VD99].

Discretized Objective Function

Directly optimizing the objective function Φ as done in 3D is possible.

When a sphere triangulation warping is used for T, the parameters that need to be estimated are locations on the sphere, $\{T(v) : v \in V\}$, where V is the control mesh vertex set. If the three Cartesian coordinates are used as parameters for $T(v)$, the problem becomes a constrained optimization as points must lie on the sphere, i.e. $||T(v)|| = 1$ for all v. In general, constrained optimization is more computationally expensive than unconstrained optimization [NW99]. However since the constraint is a simple one, if the displacements are small enough with an unconstrained optimization in $\mathbb{R}^3$, the vertices $\{T(v)\}$ can simply be projected to the sphere after each iteration. This is done, e.g. by Fischl et al. [FSTD99].

The algorithm used for the experiments in this thesis avoids constrained optimization in a different manner, using a two-step registration method described next.

6.4.2 Two-Step Registration

Recall that a two-step registration formally uses two transformations, T and U, each a map $\mathbb{S}^2 \rightarrow \mathbb{S}^2$. The algorithm (Algorithm 2 on page 48) iterates over a matching step

$$U = \arg\min_{U'} \Phi_D(U') + \Psi(T - U'),$$

where Φ_D is the data term and $\Psi(T - U)$ penalizes deviation of transformation U from T, followed by a smoothing step that sets T to be a smoothed version of U. One of the advantages of this algorithm is that, when used with a regional data term, the objective function is separable as discussed in Section 3.4.2.

If the auxiliary space is the plane, any two-step registration used for 3D, e.g. ANIMAL, can be adapted straightforwardly.

Using the sphere auxiliary space, an awkward constrained optimization can be neatly avoided by using a suitable penalty term, Ψ. Suppose the objective function is separable (e.g. a regional data term is used) and consider the terms related to control mesh vertex v. The matching step searches for $U(v)$ in a neighbourhood of point $T(v)$. Suppose the penalty is designed so that $U(v)$ is restricted to the hemisphere centred on $T(v)$. This is not hard to do; for example, let $\Psi(T - U) = \sum_v \psi(T(v), U(v))$, where ψ becomes infinite when the angle between $T(v)$ and $U(v)$ reaches $\pi/2$. With the search space restricted to a hemisphere, it can be parameterized using two variables, by projecting the points of $\mathbb{R}^3$ to the tangent plane at $T(v)$. Thus, $U(v)$ is obtained using a two-dimensional unconstrained optimization. This is the approach used by the algorithm proposed in this thesis.

6.4.3 Four-Step Registration

Recall that a four-step method (Algorithm 3) differs from a two-step method in that the corrections $U - T$ can be smoothed as well as U itself. As in the 3D work, this thesis concentrates on producing the best performance possible from a two-step method. If that matching is determined to be inadequate, future work can use a four-step method.

6.5 Coarse-to-Fine Hierarchy

For sphere triangulation warping, the control mesh is often refined using *quadrisection* in which each triangle is replaced by four by joining the midpoints of the three edges, as shown in Figure 6.15. The surface meshes used for work in this thesis all

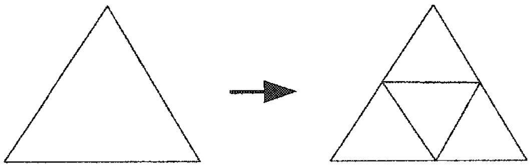


Figure 6.15: Quadrisection: each triangle is split into four by joining the midpoints.

start out as an icosahedron, the polyhedron with 20 equilateral triangular facets and 12 vertices, each of degree five. The vertices of this icosahedron all lie on $\mathbb{S}^2$. After performing a quadrisection on each facet, a polyhedron with 80 facets is obtained. Each new vertex is initially the midpoint of an edge which is then projected onto the sphere $\mathbb{S}^2$. This procedure of quadrisection followed by projection to $\mathbb{S}^2$ can be repeated, each time generating a mesh with four times as many facets as the previous. Each new vertex inserted has degree six.

6.6 Brain Surface Registration Literature

6.6.1 Labelled Features

The approaches that employ labelled features all rely on manual identification of the landmarks.

Thompson et al. [TT96, TMM+97] work with polar coordinates on the sphere as the auxiliary space. Writing the transformation as $T : (\theta, \phi) \mapsto (\theta + t_1(\theta, \phi), \phi + t_2(\theta, \phi))$, each t_1, t_2 is given by a finite sum of spherical harmonic functions. The coefficients of the expansion for t_2 are chosen to minimize the expression

$$\sum_{i=1}^{N} (\phi_i + t_2(\theta_i, \phi_i) - \phi_i^*)^2 , \tag{6.5}$$

where (θ_i, ϕ_i) are the coordinates of the i^{th} landmark of the surface, and (θ_i^*, ϕ_i^*) are the coordinates of the corresponding template landmark. The coefficients for t_1 are obtained in a similar manner. Note that Expression 6.5 penalizes geodesic distances unequally and thus fails to be invariant under rotation of the sphere as discussed in Section 6.1. Suppose points A and B share the same ϕ coordinate value, and points A^* and B^* likewise have the same ϕ value. Then both (A, A^*) and (B, B^*) contribute equally to the sum in Expression 6.5 even though one pair might be nearer to the pole and thus be much closer together on the surface, as illustrated in Figure 6.16. The matching will expend a disproportionate amount of energy matching landmarks

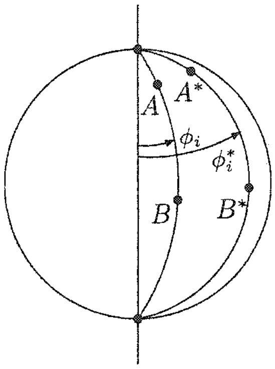


Figure 6.16: Equal penalty is imposed by Expression 6.5 for A and for B, though the former is much closer to its homologue.

near the poles, to the detriment of landmarks nearer the equator.

The Van Essen group employs large deformation landmark matching (Section 3.2.1) to align cortices after mapping to the plane or to the sphere [VEDJM98, DECS99, VELD$^+$01]. This technique is originally developed in 3D Euclidean space by Joshi and Miller [JM00] and on the sphere by Bakırcıoğlu, Joshi, and Miller [BJM99]. Thompson advocates the same approach in later work [TT99, TMV$^+$01].

In his earlier work, Davatzikos [Dav96] elastically matches landmark points using the two-unit-square parameterization of the cortex (Figure 5.5). Vaillant and Davatzikos [VD99] use $\mathbb{S}^2$ as the auxiliary space and obtain T by deforming the source mesh in $\mathbb{R}^3$, projecting it back to the sphere after each iteration. The forces used to deform the mesh are derived from distance between matched landmark points (the derivative of Equation 3.5) and elastic forces (Equation 3.24).

6.6.2 Unlabelled Features

Fischl and co-workers [FSTD99] use the average convexity feature described in Section 6.2.2 with a squared difference data term. The regularization terms are quadratic in change of signed facet area and change of edge length; the change is measured between the initial and deformed source meshes. The optimization is done in $\mathbb{R}^3$ [FSD99], projecting back to the surface at each step.

Related to the registration problem is work that searches for a cortical surface mapping in order to transfer labels from the template to an individual. The four papers mentioned here do not use an auxiliary surface, but compute mapping directly between native meshes (mapping W of Figure 6.1). Feldmar and Ayache [FA96] adapt ICP to include surface curvature in the data term and to search for non-rigid transformations. Sandor and Leahy [SL97] model the superficial surface of cortex using B-spline patches and fit to individual using deformable model techniques. Jaume, Macq, and Warfield [JMW02] describe a method in which the model mesh is coarsened using "progressive meshes" and the coarse model mesh is fit as a deformable surface (in $\mathbb{R}^3$) to the individual. The model is then refined by reversing the coarsening process, and the reinserted vertices are moved along the surface normal to the nearest point on the target. These three approaches search for matches in 3D and so are not constrained to search in surface neighbourhood of adjacent vertices: the possibility of jumping across to the opposite bank of a sulcus exists. In addition, it may be difficult to match to surface locations deep in sulci. Wang, Peterson and Staib [WPS00] start with an extremely coarse model of the cortex (69 vertices, 134 triangles), and locate corresponding vertices on the target surface. The arc joining a pair of neighbouring vertices is a geodesic path on the surface. The source and target meshes are repeatedly refined by subdividing each arc and using quadrisection to build isomorphic triangulations on the two surfaces. The result is that the matching is driven only by the coarse mesh of 69 vertices, which is very little data.

6.6.3 Multiple Features

Davatzikos [Dav97] uses a feature derived from thresholding three curvature values (the two principal curvatures and Gaussian curvature) at each vertex of the control mesh in addition to a landmark term Φ_{LM} (Equation 6.1) of points on matching sulci.

6.7 Algorithm for 2D Experiments

This section elaborates on the algorithm proposed in this thesis for registering cortical surfaces.

The algorithm is based on the unit sphere auxiliary space. Thus the assumption is that the native surfaces are of spherical topology, as discussed in Section 5.1. However, no assumption is made as to whether the surface contains one cortical hemisphere or both hemispheres, nor as to whether the inner, outer, or medial cortical surface is used (see Section 5.1). Moreover, the only assumption about the type of feature used for matching is that it is a scalar value defined at each point on the sphere. The algorithm is very general in these respects; it can be used to register any data that can be mapped to the sphere. The feature data could be anatomical or it could be functional, for example. For experiments described in Chapter 7, both the mean curvature and the crown distance transform features described in Section 6.2.2 are used.

The transformation $T : \mathbb{S}^2 \to \mathbb{S}^2$ is parameterized as a sphere triangulation warp (Definition 5.4.8 on page 127). Working on the sphere, coordinate invariance in the auxiliary space is achieved by using a two-step algorithm where the matching and smoothing steps each integrate over small circular neighbourhoods of the sphere.

The experiments presented in the sequel all use the inner cortical surface with both hemispheres mapped to a single unit sphere, obtained using the ASP code of MacDonald [MKAE00, Mac98]. The cortical surface extracted using ASP is a mesh of 81920 facets and 40962 vertices. This mesh is obtained by starting with an icosahedron (20 facets, 12 vertices, each of degree 5) and performing quadrisection six times. The projection of the native mesh onto $\mathbb{S}^2$ using reverse iterative deformation (see Section 5.2) is precisely this sixfold quadrisection of the icosahedron. This is the mesh used for the feature data interpolation functions S and R, described in Section 6.2.2.

There are thus three separate meshes involved in registration, each of which is embedded on the sphere $\mathbb{S}^2$: the control mesh for T, the source data mesh for interpolating source data function S, and the target data mesh for interpolating the target data function R. Each of these meshes is implemented using the halfedge data structure of CGAL [Ket99, CGA03].

6.7.1 Outer Loop

Control Mesh Refinement

As discussed in Section 6.5, the control mesh is obtained by repeated quadrisection of an icosahedron. The finest level mesh is designed to coincide with the data mesh extracted by ASP [Mac98], which is a sixfold quadrisection with $4^6{\cdot}20 = 81920$ facets. Initial experimentation indicates that using a very coarse mesh is not effective, so the initial mesh used starts with a threefold quadrisection ($4^3 \cdot 20 = 1280$ facets). Thus there are four levels of the outer loop, with $4^3 \cdot 20$, $4^4 \cdot 20$, $4^5 \cdot 20$, and $4^6 \cdot 20$ facets for the control mesh. After each level, the current transformation is interpolated onto the finer control mesh for the next iteration.

The initial mapping is obtained by a low-dimensional warping, e.g. a rotation of the sphere as discussed in Section 5.3. However, for the experiments performed here, the initialization is simpler still. Each MR image is initially registered using a 9-parameter affine transformation to a common target. Then a standard ellipsoid is deformed by ASP to identify the cortical surface in the image. Since the images are aligned, a given vertex v of the initial mesh tends to end up in a nearly homologous location on each image [MKAE00]. This means that the initial transformation on the coarsest control mesh can simply be set to the identity mapping. More evidence for this is shown in the next chapter, in the discussion around Figure 7.15.

Data Refinement

When the feature value map requires smoothing so as to reduce noise, the smoothing is done in a coarse-to-fine manner in step with the control mesh refinement. The coarse control mesh is used with heavily-smoothed data, with the amount of smoothing being reduced after each level of the hierarchy. Let $D(v)$ denote the data value at vertex v. The data smoothing replaces, in parallel for all v, $D(v)$ by

$$(D(v) + a \sum_{u \in N_v} D(u))/\alpha, \tag{6.6}$$

where N_v is the set of neighbours of v, $\alpha = 1 + a|N_v|$ is a scaling factor to ensure the weights add to 1, and a is a constant, selected using visual inspection to be 0.4 for experiments in this thesis. The smoothing operation is iterated 128 times at the coarsest level of control mesh, 32 for the next, then 8, and finally twice at the level of the finest control mesh.

The result of smoothing the mean curvature data term shown in Figure 6.4 is shown in Figure 6.17. The crown distance transform data (see Figure 6.10) is deemed sufficiently smooth, so no extra smoothing is done.

6.7.2 Inner Loop

The inner loop is a two-step algorithm patterned after ANIMAL, displayed as Algorithm 5.

Step 1: Matching

The data term (ϕ^v in Step 1) is based on the regional correlation coefficient similarity measure (the regional version of Φ_{CC}, described in Section 3.2.2). Recall that Φ_{CC} produces a value in range $[-1, 1]$ with better match indicated by larger score. The

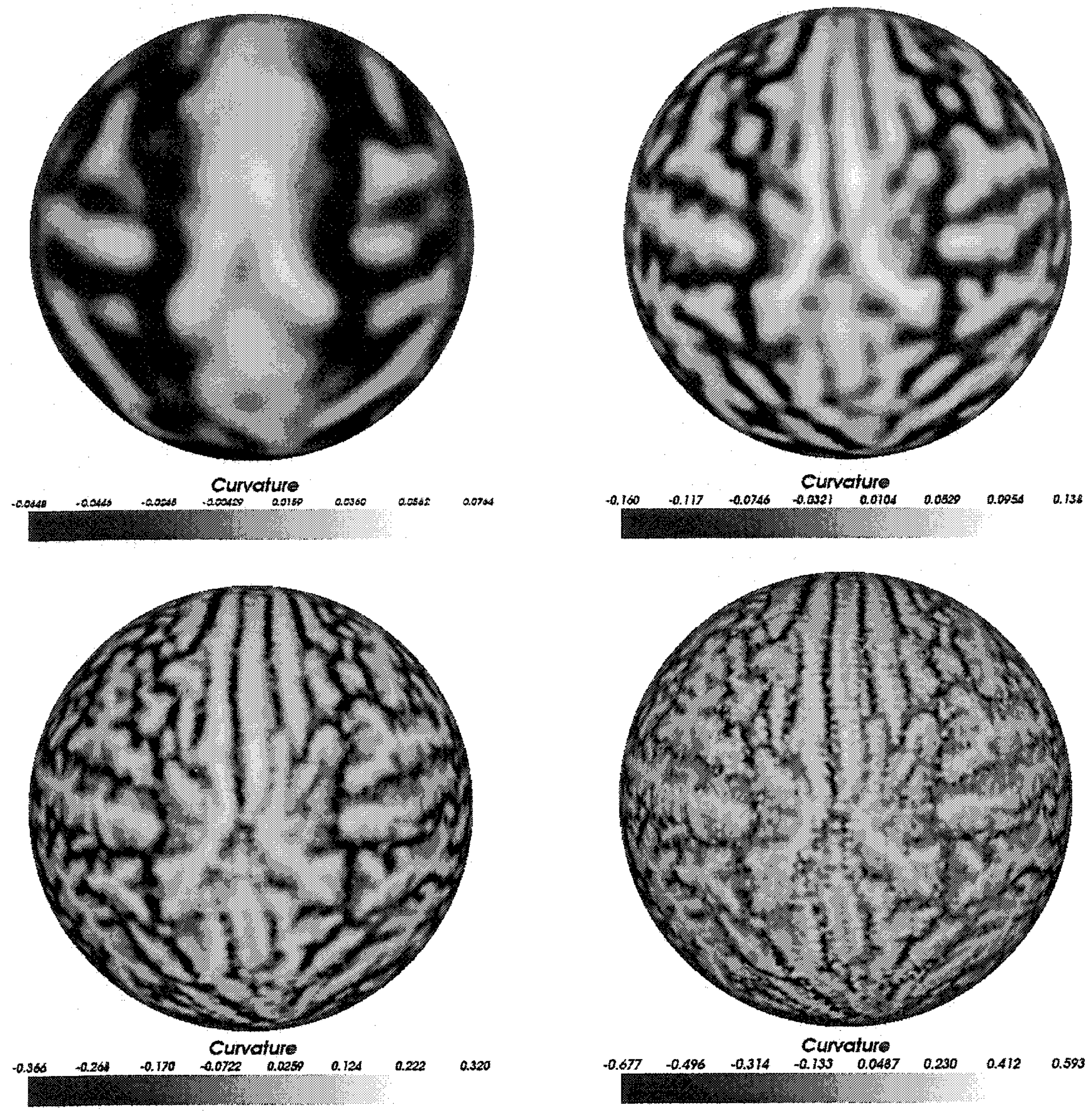


Figure 6.17: Smoothed mean curvature data for the four levels of coarse-to-fine outer loop.

Algorithm 5 Inner loop of Surface Registration.

```
1. Minimize $\Phi(U) = \sum_v (\phi^v(U(v)) + a\psi(||U(v) - T(v)||))$.
2. Let $C(v)$ be centroid of $\{U(u) : u$ is 1-ring neighbour of $v\}$.
   Set $T(v) = U(v) + wC(v)$, projected to $\mathbb{S}^2$.
3. Loop over Steps 1-2 a fixed number of times.
```

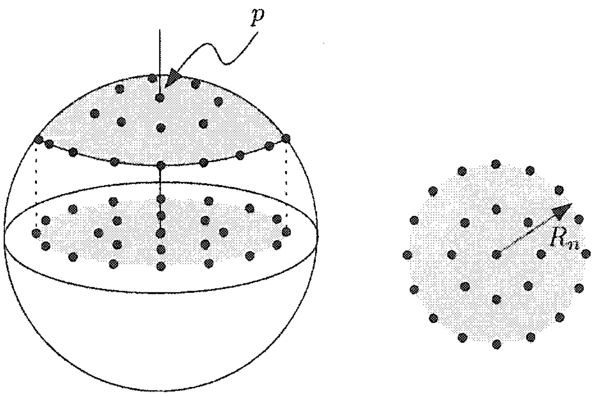


Figure 6.18: Spherical cap neighbourhood for evaluating similarity measure at p. Point p is a control mesh vertex, either v on the source or $U(v)$ on the target sphere. The neighbourhood is parameterized by a disc of radius R_n in a plane parallel to the tangent plane at p. The right figure shows how the sample points are arranged in the disc.

function to minimize is

$$\phi^v(U) = 1 - \phi^v_{CC}(U),$$

where ϕ^v_{CC} is the correlation coefficient evaluated between a neighbourhood of v on the source data and a neighbourhood of $U(v)$ on the target data.

Each neighbourhood is a spherical cap at the given centre point p, where p is either v or $U(v)$. The correlation coefficient is evaluated by sampling a number of points in each neighbourhood. The sample points are arranged on a disc of radius R_n parallel to the tangent plane at p, as illustrated in Figure 6.18. The set of sample points are: the centre of the disc, 8 points equally spaced around a circle of radius $R_n/2$, and 16 points equally spaced on a circle of radius R_n. The disc radius is

$$R_n = r_n d_C, \tag{6.7}$$

where r_n is a user-specified radius factor, and d_C sets the distance scale based on the coarseness of the control mesh. The value of d_C is the distance, projected to the disc, from the centre of the cap to a neighbouring control mesh vertex. In other words, d_C is the length of a typical control mesh edge after projecting to the disc. The value of R_n must be no greater than 1, so $r_n \leq 1/d_C$.

The $\psi(||U(v) - T(v)||)$ term in Line 1 of the algorithm is intended to limit the search for $U(v)$ to the hemisphere centred at $T(v)$ as discussed in Section 6.4.2. The search region is parameterized using a disc parallel to the tangent plane at $T(v)$. Points $T(v)$ and $U(v)$ are projected to this disc and r is defined as the distance

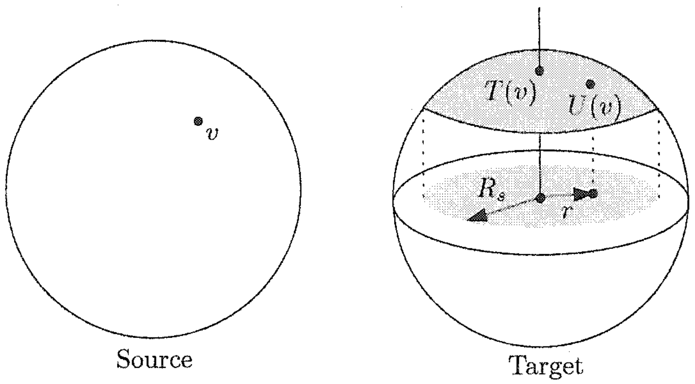


Figure 6.19: The penalty term ψ restricts the search for $U(v)$ to a spherical cap neighbourhood of $T(v)$, parameterized using a disc of radius R_s.

between the projected points as illustrated in Figure 6.19. The penalty is a logarithmic barrier function [NW99], $\psi = -\log(1 - r^2/R_s^2)$, which has the effect of constraining the search for $U(v)$ to points with $r < R_s$. The disc radius is

$$R_s = r_s d_T, \tag{6.8}$$

where r_s is a user-specified search radius factor, and d_T sets the distance scale based on the control mesh. The value of d_T is the distance, projected to the disc, between $T(v)$ and $T(u)$ where u is the control-mesh neighbour closest to v on the target, i.e. $||T(v) - T(u)||$ is minimum for all 1-ring neighbours u of v. The value of R_s must be no greater than 1, so $r_s \leq 1/d_T$.

The objective function is designed to be separable so that the minimization of

$$\phi^v(U(v)) + a\psi(||U(v) - T(v)||) \tag{6.9}$$

is performed independently for each control mesh vertex v. Each optimization is a 2-dimensional problem, parameterized by Cartesian coordinates in the tangent plane at $T(v)$. The initial iterate is $(0, 0)$ which corresponds to $T(v)$, and is a feasible point since it corresponds to $r = 0$ so the barrier function ψ is zero. The optimization is performed using the Nelder-Mead downhill simplex algorithm [PFTV88] as implemented in the GNU Scientific Library [GDT$^+$02]. The Nelder-Mead simplex algorithm is chosen because it does not require derivatives of the objective function 6.9, which are complicated due to the correlation coefficient data term in ϕ^v.

Step 2: Smoothing

The model term is given implicitly by specifying the smoothing operation (Line 2 of Algorithm 5). The smoothing, which is carried out in $\mathbb{R}^3$, is a simple weighted average of $U(v)$ and the centroid of its neighbourhood,

$$C(v) = \frac{1}{|N_v|} \sum_{u \in N_v} U(u),$$

where N_v is the set of neighbours of v. The smoothed transformation is given by

$$T(v) = \frac{U(v) + wC(v)}{||U(v) + wC(v)||},$$

where w is a user-specified smoothing weight.

Efficient Point Location

During the minimization of Expression 6.9, the location of $U(v) \in \mathbb{S}^2$ is updated many times. The location of $U(v)$ specifies the central point of the neighbourhood on the target sphere that is used to evaluate the correlation coefficient data term. The target feature value must be evaluated for each of the points shown in Figure 6.18. As discussed in Section 6.2.2 the feature value at point $x \in \mathbb{S}^2$ is obtained by triangulation interpolation, for which the triangle containing x must be identified. This would, in general, involve a linear-time search across the target data mesh, as discussed in connection with Figure 5.9 on page 128. The search time is reduced in the following manner.

The transformation at vertex v is a point $T(v) \in \mathbb{S}^2$. However, instead of storing the Euclidean coordinates for the point $T(v)$, the triple (h, α, β) is stored where h is a halfedge (internally represented as a pointer) specifying the spherical triangle of the target data mesh that contains $T(v)$ and (α, β) are the barycentric coordinates (along with $\gamma \equiv 1 - \alpha - \beta$) locating $T(v)$ on the spherical triangle. The situation is illustrated in Figure 6.20. Each time the data term needs to be evaluated during the optimization, the search for $U(v)$ always begins at the triangle indicated by h. Since $U(v)$ is known to be near $T(v)$, the search will traverse few triangles. Subsequently, the points in the similarity neighbourhood of Figure 6.18 are obtained starting the search at the triangle containing $U(v)$.

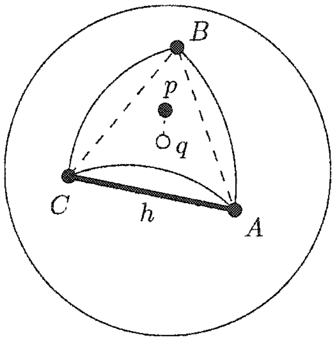


Figure 6.20: Point $q = \alpha A + \beta B + \gamma C$ is represented by triple (h, α, β), where h is the halfedge from C to A, and $\gamma = 1 - \alpha - \beta$. Point $p = q/||q||$.

Parameters

The user of this algorithm has four major parameters to specify: the search radius r_s, the neighbourhood radius r_n, the penalty ratio a, and the smoothing weight w. Note that the search radius and neighbourhood radius are dimensionless quantities that multiply a length set by the coarseness of the control mesh through Equations 6.7 and 6.8, respectively. Thus r_s and r_n can be set to a fixed value for all iterations of the outer loop, as are parameters a and w.

Experiments in the next chapter determine values for these parameters.

6.8 Summary

The chapter objectives are first to review the surface registration problem, mainly highlighting the differences from the image registration problem considered in Chapter 3, and second to propose a new algorithm for surface registration. Surface registration algorithms are put into a common optimization framework in order to compare them on the basis of the data terms and model terms, just as is done for image registration.

6.8.1 Data Terms

Landmark points can be used with distance based data terms, as is done with image registration. This approach is applicable in both the plane and the sphere auxiliary space [VEDD$^+$01]. To date, the method of identifying and matching the points has been manual [TT96, DECS99, VD99] which precludes using this approach for spatial normalization in large scale automated analyses. Automatic labelling of the cortical surface mesh, e.g. [RHXP02, RMPO$^+$02] could be an aid, but as is the case for image registration (Section 3.8.1), there is no system that generates enough labels for a high-dimensional warp that requires, say, two parameters at each of 40 000 control mesh nodes. Thus, this thesis concentrates on obtaining the best results possible without label data.

In the case of image registration (Chapter 3), the matching problem without label data is based on the image data itself, a real-valued function defined on the source and target domains. The surface registration problem, in contrast, does not have such a real-valued function given as part of the problem specification, so one must be created. In order to match the pattern of sulcal folding, a function is sought that reflects the geometry of the folding pattern.

The three features described in Section 6.2.2 all locate sulci and gyri in some manner. Mean surface curvature, given by Equation 6.3, can be used to identify the type of fold by the sign of the curvature value. Using the outward-normal convention, discussed following Equation 6.3 on page 140, points in the fundus of a sulcus generally have positive mean curvature while points on the gyral crown generally have negative mean curvature. This classification is not foolproof, since a outward bump at the bottom of the sulcus will have positive mean curvature, so some smoothing is required with this feature. The average convexity feature of Fischl et al., discussed on page 141 is one way of smoothing out these fluctuations [FSTD99]. Using either of these features will tend to match points along the crown of a gyrus on the source surface with points along the crown of a gyrus in the target surface. Similarly, the fundus of a sulcus will tend to match the fundus of a sulcus. Points along the bank of a sulcus, in between the crown and the fundus, will also tend to match to points on the target with similar curvature. The gyral crown distance transform described on page 142, in contrast, is designed to match these points on the banks of sulci according to their fractional distance towards the bottom of the sulcus, e.g. a vertex halfway down the sulcus in the source mesh should match a point halfway down the target sulcus.

The gyral crown distance transform feature values are generated in two steps. First, seed points representing the gyral crowns are identified using a method based on the α-shape of Edelsbrunner and Mücke [EM94]. The α-shape allows extreme vertices (the α-exposed vertices described on page 145) of the surface to be labelled as gyral, a discrete-geometry method similar in spirit to the deformable surface of Rettmann et al. [RHXP02]. The second step is to generate the geodesic distance transform from the gyral seed vertices. Approximate geodesic distances are obtained using Dijkstra's single source shortest-paths algorithm on the surface mesh augmented with $3m$ extra vertices and $O(m^2)$ edges on each triangular facet [LMS01]. Figure 6.13 shows that the approximate distance values become close to the true values after about $m = 4$ or 5, which is in agreement with the observations of Lanthier et al. [LMS01]. The experiments in this thesis use distance transforms generated with $m = 5$.

6.8.2 Model Terms

Here the options are generally similar to image registration. So far work has focused on enforcing smoothness either by expanding the transformation using a finite set of basis functions [TT96], using a regularization term based on derivatives of the transformation [JM00, BJM99, VD99], or a penalty based on the Jacobian determinant [FSD99, FSTD99]. The model terms for the surface registration algorithm proposed in Section 6.7 are implemented as smoothing of the transformation, as the algorithm is based on the two-step algorithm of Section 3.4.2.

6.8.3 Registration Algorithm

The approaches used for optimization depend on the auxiliary space chosen. When using the plane, the methods of image matching can be easily adapted. When the sphere is chosen as the auxiliary space, one option is to optimize $\{T(v)\}$ by working in $\mathbb{R}^3$, taking a small step, and projecting the result back to the sphere so that $||T(v)|| = 1$ for all control mesh vertices v [FSTD99, VD99].

The new algorithm proposed in Section 6.7 is based on the two-step algorithm. The transformation is parameterized using a sphere triangulation warp (Definition 5.4.8 on page 127) and a regional data term is chosen so that the optimization is separable. The optimization for each control mesh vertex, v, is therefore performed independently. Each such optimization is two-dimensional, which allows use of the (quadratic storage) Nelder-Mead downhill simplex algorithm [PFTV88]. This optimization method is chosen because it does not require derivatives of the objective function 6.9, which are complicated due to the correlation coefficient data term (Equation 3.13). The location of $T(v)$ in the source and target data meshes is stored in terms of a triangle and barycentric coordinates within the triangle as illustrated by Figure 6.20, rather than using the Cartesian coordinates of the point in $\mathbb{R}^3$. This speeds up the evaluation of the data term, which requires locating a number of points in the target data mesh that are near $T(v)$. The point location algorithm is, in general, a linear-time search across the target data mesh, as discussed in connection with Figure 5.9 on page 128. By storing the mesh location of $T(v)$, each search starts near to the point sought and thus only examines a few facets. The registration algorithm employs a typical coarse-to-fine strategy with four levels. The data values are blurred less at each level, while the control mesh is refined using quadrisection (Figure 6.15) of each facet at each level.

This surface registration algorithm is used for 2D spatial normalization in the following chapters.

Chapter 7

Quantifying Spatial Variability

This chapter has two goals. The first goal is to transfer the concepts and measures of variability developed for $\mathbb{R}^3$ in Chapter 4 to the sphere auxiliary space, $\mathbb{S}^2$. The second goal is to use these measures to demonstrate that the surface registration algorithm proposed in Section 6.7 does succeed in reducing spatial variability.

Spatial variability for 2D spatial normalization is quantified using two measures analogous to the two measures used to quantify variability after 3D spatial normalization in Chapter 4. The main difference being that the 3D measures are evaluated on a voxel grid whereas a triangular mesh is used in 2D. The first measure, segmentation variability, gives a global measure of spatial normalization based on the extent to which gyral crowns are matched to gyral crowns and sulci are matched to sulci. This measure is used to examine design choices of the algorithm and to choose the values for the numerical parameters. The second measure, structural variability, is the dispersion of a specific structure in the standard space and is used to form the localization measure.

7.1 Segmentation Variability

A segmentation of the surface is defined to be an assignment (manual or automatic) of a class label to each vertex. This is analogous to a segmentation of a 3D grid, where the labels were assigned to each voxel (Section 4.1).

As is done in 3D, labels assigned to an input surface that has been spatially normalized can be carried along with a spatial transformation to induce a segmentation on a mesh in the standard space. Each vertex of the standard space mesh is assigned a label by each input surface. The goal is to measure the consistency of the set of labels assigned to a given standard space vertex.

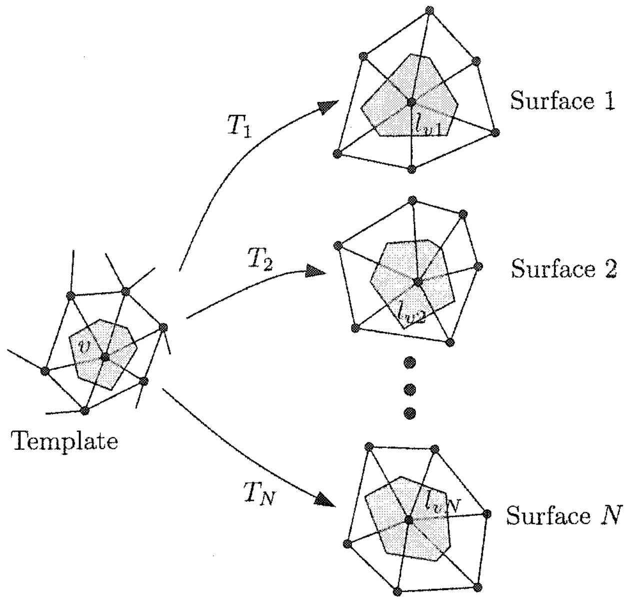


Figure 7.1: Induced labelling on standard mesh from spatially normalizing a population. For each subject i, transformation T_i maps standard space vertex v to the cell (shaded) on surface i with label l_{vi}. The resulting set of labels, $\{l_{v1}, \ldots, l_{vN}\}$, constitutes a sampling of the random label L_v of standard space vertex v. The vertex entropy, $H(L_v)$, is estimated from this sampling.

Consider vertex v of the standard space mesh. This vertex maps to a certain point in subject i, which has label l_{vi} as illustrated in Figure 7.1. A spatial normalization method that achieves its goal of matching homologous points of each input will result in identical labels across the subjects ($l_{v1} = l_{v2} = \ldots$) for each vertex v. As in 3D (Section 4.1), the label of standard vertex v is considered as a random variable L_v for which the entropy is computed as

$$H(L_v) = -\sum_l p_l \log_2 p_l, \tag{7.1}$$

where p_l is the probability that L_v is assigned label l. The sum is taken over the labels l for which $p_l \neq 0$. The probabilities p_l are estimated as the frequencies observed after spatially normalizing a sample population as follows. Let N_l be the number of subjects for which the vertex label is l, i.e. the size of the set $\{i : l_{vi} = l\}$. Then the probability p_l is estimated as $p_l = N_l/N$, where N is the total number of subjects. The entropy $H(L_v)$ measures the amount of uncertainty in label L_v (in bits, when using base-2 logarithms), which is regarded as the anatomical variability at vertex v. The total of the entropy values is used as a measure of variability remaining after

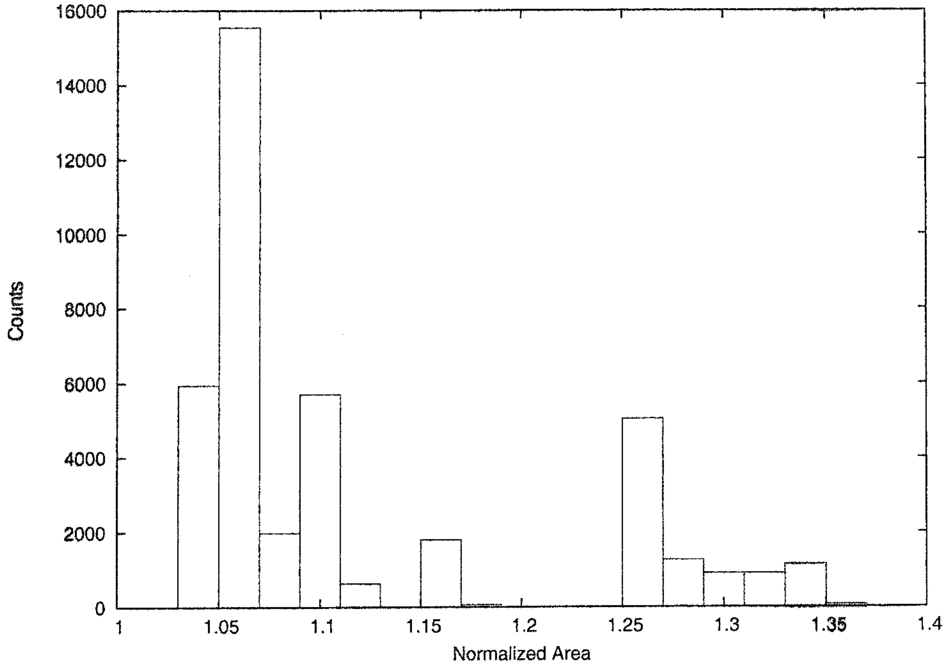


Figure 7.2: Normalized Voronoï cell areas on the standard mesh of 40962 vertices. About 2/3 of the vertices have normalized area in the narrow range of 1.05-1.10.

spatial normalization is applied.

On the 3D grids used in this thesis, each voxel is of equal size, so the total entropy used as a performance measure gives equal weight to the entropy of each voxel. On the 3D grid, each voxel is associated with a vertex located at the voxel centre. The voxel (except for boundary voxels) is actually the Voronoï cell of the vertex. The voxel label can thus be considered as the label for the Voronoï cell of the vertex, a concept that transfers directly to 2D. In contrast to the situation in 3D, the cells of the 2D mesh are not all of equal size. The largest cell has area about 1.36 times the area of the smallest, though most cell areas fall in a small band of areas about 5-10% larger than the minimal triangle area as the histogram of Figure 7.2 demonstrates.

To account for the variable cell size, the contribution of each cell is weighted according to the relative cell area with the smallest cells assigned unit weight. The total entropy in 2D is defined as

$$H = \sum_{v \in V} w_v H(L_v), \tag{7.2}$$

where w_v is the area of cell v divided by the area of a smallest cell. The quantity H, termed the total entropy, is used as a measure of variability remaining after spatial

Table 7.1: Major Parameters of surface registration algorithm.

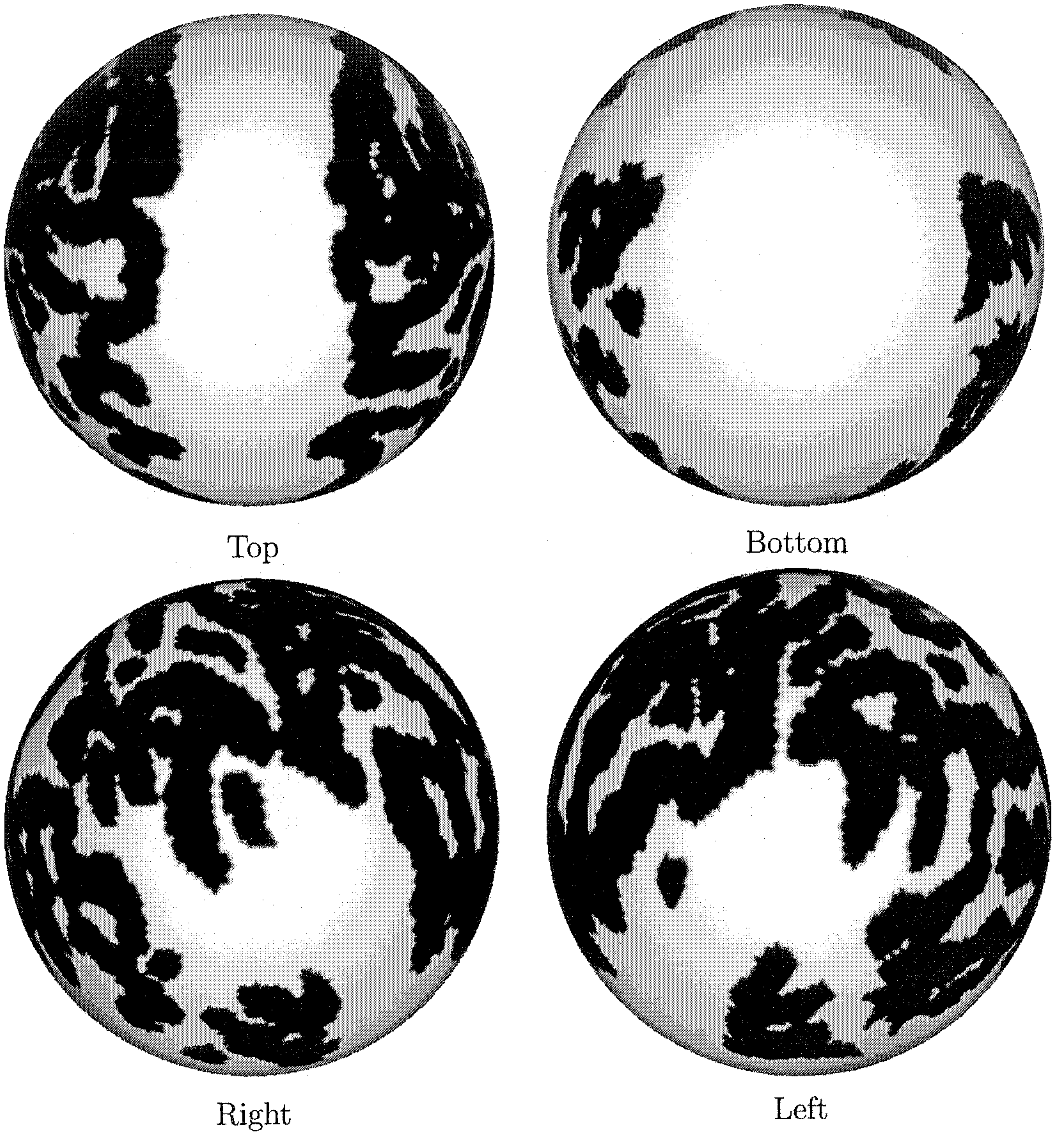
Symbol	Description
r_s	Search Radius
r_n	Neighbourhood Radius
a	Penalty Ratio
w	Smoothing Weight.

normalization is applied.

7.1.1 Application: Evaluating Algorithm Design

The inner loop of the surface registration, Algorithm 5, has four parameters to select, summarized in Table 7.1. An optimal value for each of these parameters is chosen using total entropy of a segmentation, as was done for ANIMAL in Section 4.1.1. The same template and ten test subjects that are used for the 3D work are used again to probe the choice of data term and to locate optimal parameter values.

In order to generate the segmentation of each test subject's surface, an automated vertex classification is required analogous to the tissue classification of voxels used in 3D. The cortical surface mesh is presumed to lie entirely on the boundary of two tissue types, specifically white matter and gray matter for the experiments presented here. Thus classification into tissue types is not an option. Instead, each vertex is classified as either gyral (lying on the gyral crown) or non-gyral. The classification is achieved by simply thresholding the crown distance transform values at distance 10, a value empirically chosen. Figure 7.3 shows a typical surface segmentation. Recall from Section 6.2.2 on page 145 that the crown distance transform is designed to remove from the transformation seed set, the points on the surface that lie on the cap through the brain stem. Thus no such points are labelled as gyral by the segmentation. Indeed, Figure 7.3 shows that there are no vertices labelled as gyral anywhere near this cap. This means that the vertices associated with this part of the surface will generate a consistent set of sample labels resulting in a vertex entropy, $H(L_v)$, of zero. The total entropy is therefore not sensitive to the mapping on the unphysical parts of the surface.



Top

Bottom

Right

Left

Figure 7.3: Views of a typical gyral surface segmentation, with gyral points shown in black. Note in the bottom view that the unphysical cap through the brain stem is all the same label (non-gyral). These vertices will generally contribute zero to the total entropy, since they tend to map to non-gyral vertices on every input.

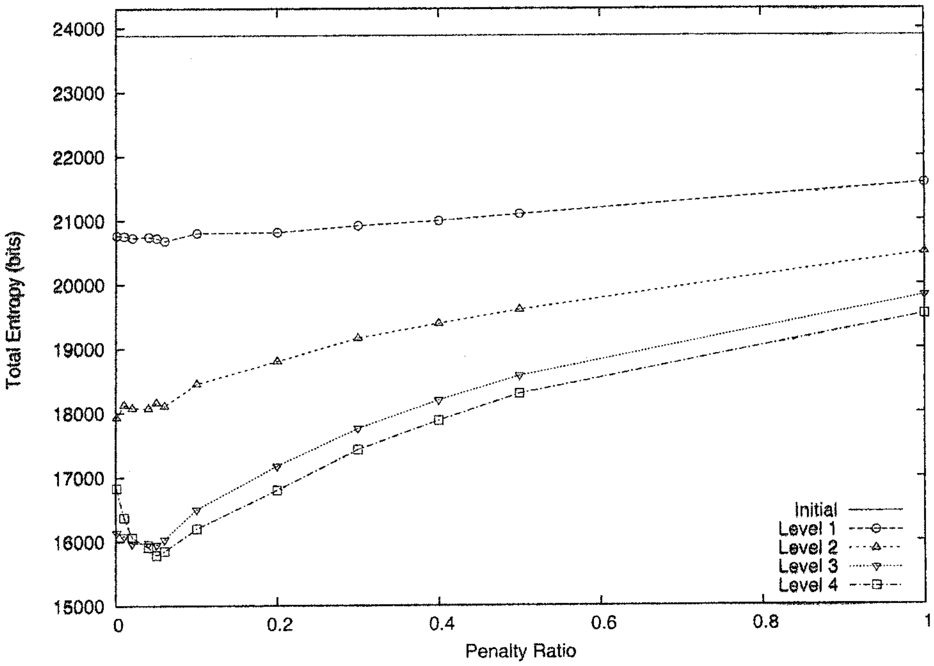


Figure 7.4: Residual anatomical variability as measured by total entropy, H, on a sample of 10 individuals after surface registration, along with the initial (unregistered) value for comparison. The control mesh of each level is the quadrisection of the previous level; level 1 is the coarsest control mesh. Note the reduction of total entropy value with each mesh refinement level.

Outer Loop

The test data is normalized using several choices for the numerical parameters and the total entropy after each of the four iterations of the outer loop is computed. Figure 7.4 shows representative results using the crown distance transform data term with $r_s = 0.5$, $r_n = 2.8$, $w = 1$, and a range of penalty ratio values, a. The plot demonstrates that alignment improves at each finer resolution of control mesh.

Data Term

Both the mean curvature and the crown distance transform presented in Chapter 6 are possible data terms. A search over several parameter sets yields a parameter set ($r_s = 0.5$, $r_n = 2.8$, $w = 1$) that provides good performance, i.e. low total entropy, with each data term. Since the entropy is much lower when using the crown distance transform, as shown in Figure 7.5, all further experiments in this thesis use the crown distance transform.

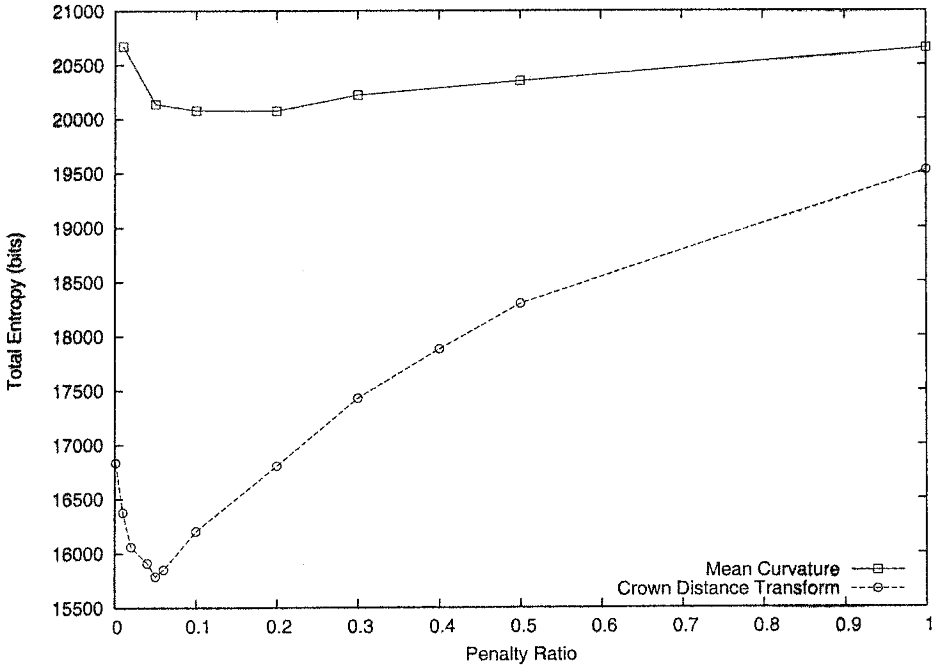
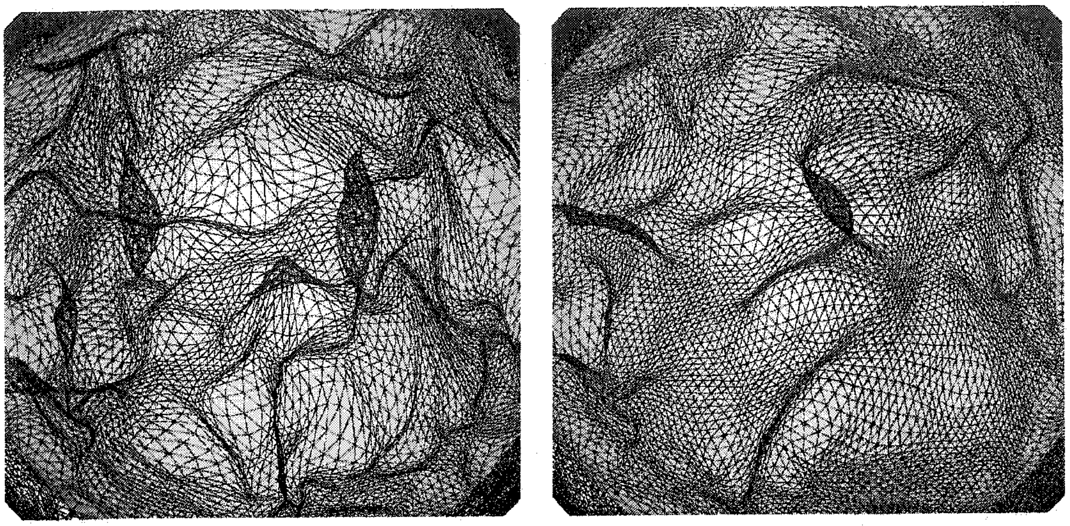


Figure 7.5: Residual anatomical variability as measured by total entropy, H, on a sample of 10 individuals after surface registration. The plot compares the results using mean curvature as the feature value for the data term, and using crown distance transform as the feature value. The crown distance transform produces a normalization with much less variability.



Penalty Ratio 0.001

Penalty Ratio 0.05

Figure 7.6: Control mesh vertices shown in their displaced positions. The result shown on the left, using a very low penalty ratio parameter, allows more folding of the mesh.

Regularization

The trade-off between the data match and the amount of displacement allowed during one step of the inner loop is controlled by the penalty ratio, a. This trade-off occurs in Line 1 of Algorithm 5 which minimizes the following expression for each control mesh vertex v,

$$\phi^v(U(v)) + a\psi(||U(v) - T(v)||),$$

where ϕ^v is the data term and ψ is the penalty term. Figure 7.5 shows that $a = 0.05$ is optimal for the crown distance transform data feature.

Setting the value of a to zero eliminates the regularization and so, as with 3D (Figure 4.6), the triangulation warping is expected to be less smooth. This is illustrated in Figure 7.6, where the mesh warped using the lower penalty ratio shows larger regions where the mesh has folded (i.e. the transformation is noninjective) and a larger variation in size of the warped triangles, both signs of a less smooth warping.

Smoothing

The smoothing step of Algorithm 5 sets the location of vertex v to a weighted average of the location obtained by the data matching step, $U(v)$, and the centroid $C(v)$

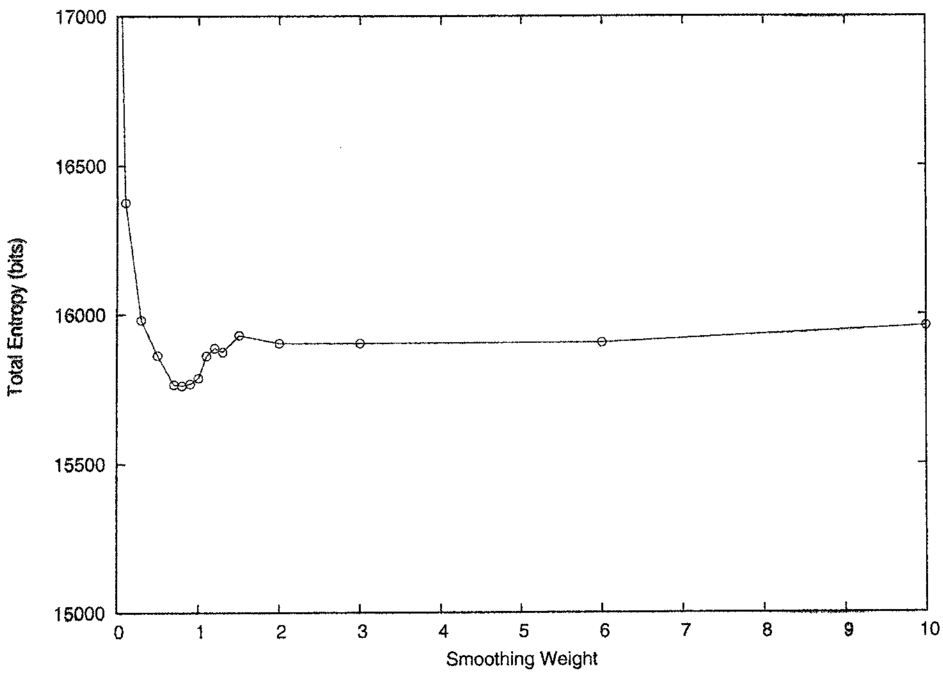


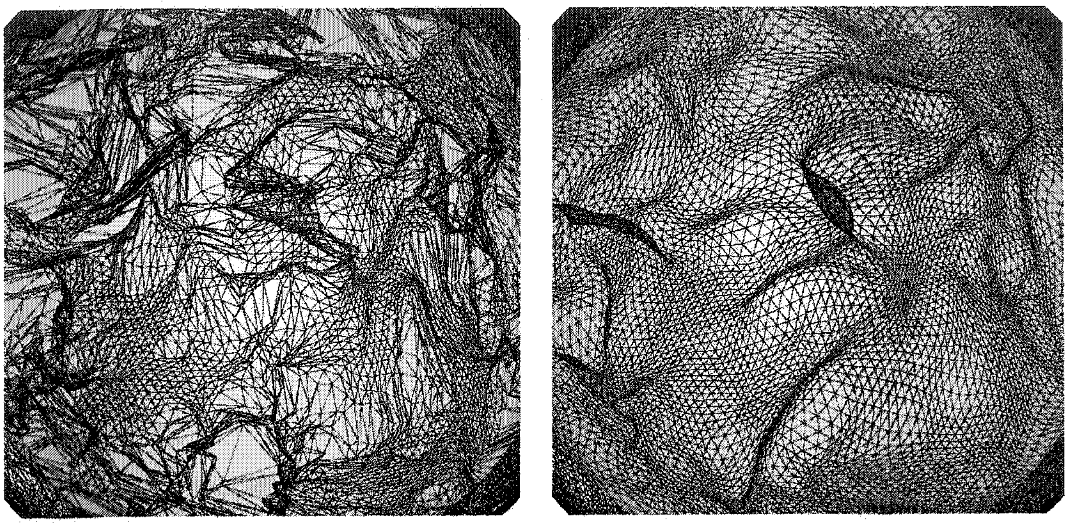
Figure 7.7: Residual anatomical variability as measured by total entropy, H, on a sample of 10 individuals after surface registration. Smoothing sets the position of control mesh vertex v to a weighted sum of the position $U(v)$ obtained by the matching step and the centroid computed from the neighbours of v. With large smoothing weight values, the sum asymptotically becomes equal to the centroid value. A weight value near 1 is best.

computed from the neighbours of $U(v)$. The smoothed transformation is given by

$$\frac{U(v) + wC(v)}{||U(v) + wC(v)||},$$

where w is the smoothing weight value. Figure 7.7 plots the entropy as a function of smoothing weight, showing weights near $w = 1$ provide the best performance.

Note that large values for w will asymptotically set the smoothed mesh equal to the centroid value, $C(v)/||C(v)||$, and the performance does indeed level off in Figure 7.7. For low values of smoothing weight, the plot shows a sharp increase in entropy. This happens because the lack of smoothing allows more folding of the mesh, as illustrated in Figure 7.8, which produces poor performance.



Smoothing weight 0.01 Smoothing weight 1

Figure 7.8: Control mesh vertices shown in their displaced positions. The result shown on the left, using a very low smoothing weight, allows more folding of the mesh.

Miscellaneous Parameters

The other two major parameters for the algorithm are the search radius and the neighbourhood radius. Figure 7.9 illustrates the optimal values $r_s = 0.5$ and $r_n = 2.8$, respectively.

Recall that the search radius is measured in units of the distance to the nearest control mesh vertex neighbour on the target sphere (see Equation 6.8). Using a small search radius prevents the optimization from finding a good data match and results in a warping that is near the initial transformation and thus produces high total entropy. The poor performance produced by a large search radius, on the other hand, is because triangles are too easily able to reverse orientation. The value $r_s = 0.5$ restricts the search radius to half the distance to the nearest control mesh vertex neighbour.

The neighbourhood radius is measured in units of the distance to the nearest control mesh vertex neighbour on the source sphere (see Equation 6.7). As discussed in Section 6.2.2, the feature value (i.e. the crown distance transform) is a function defined on the sphere using sphere triangulation interpolation, i.e. it is interpolated from values on a given data mesh. A small radius limits the number of data mesh vertices that participate in computing the correlation coefficient data term, thus limiting the information available for matching. In addition, there is less overlap between the neighbourhood caps for adjacent control mesh vertices with small r_n, so

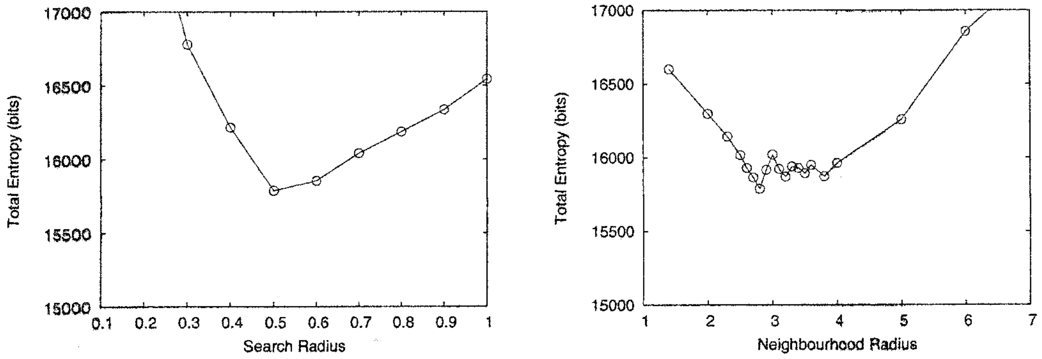


Figure 7.9: Residual anatomical variability as measured by total entropy, H, on a sample of 10 individuals after surface registration. The plot on the left uses optimal neighbourhood radius of 2.8, while the plot on the right uses optimal search radius of 0.5; both plots use penalty ratio = 0.05 and smoothing weight = 1.

the transformation obtained after the matching step may be less smooth. At large values of r_n the neighbourhood can sample points from more than one sulcus or gyrus, rendering the matching step less reliable. In addition, the data term may be undersampled as it is evaluated on a fixed number of sample points in the neighbourhood, so the density of the sampling decreases with increasing r_n. The behaviour for r_n near 3 is likely due to an aliasing effect caused by the changing sample density.

At each iteration of the outer loop, the surface registration runs the inner loop for a fixed number of iterations. For the experiments in this thesis, the registration runs the inner loop for 20 iterations. Figure 7.10 indicates that the optimization has adequately converged for each level before the next level of the outer loop begins.

Injectivity

In order for the sphere triangulation warping be injective, it is necessary that both the cyclic ordering of neighbours around each vertex as well as the orientation of each triangle be preserved, as stated by Theorem 5.5.5. The surface registration algorithm (Section 6.7) used for experiments in this thesis does not ensure an injective mapping. To gauge the extent of noninjectivity encountered in practice, the area of the target sphere covered by triangles that have their orientation reversed is computed.

The template is registered to 151 other ICBM subjects using parameters $r_s = 0.5$, $r_n = 2.8$, $a = 0.05$, and $w = 1$. The largest area observed is 0.13, which is about 1% of the 4π surface area of the unit sphere ($4\pi \approx 12.57$). As shown in Figure 7.11, most of the subjects were about 2/3 of this number. Figure 7.12 illustrates the distribution of reversed orientation triangles. While such triangles appear in many

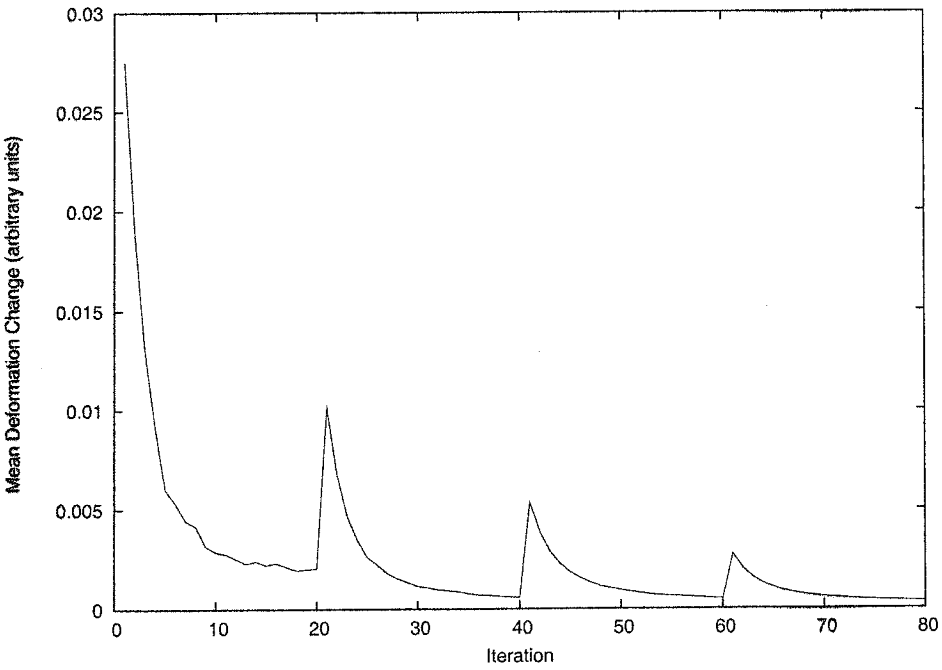


Figure 7.10: Mean magnitude of change in vertex position, plotted for each iteration of the inner loop. The registration is done using parameters $r_s = 0.5$, $r_n = 2.8$, $a = 0.05$, and $w = 1$. The increases at iterations 20, 40, and 60 result from starting the second, third, and fourth iteration of the outer loop. Convergence of inner loop is demonstrated by the mean change falling off before the next iteration of the outer loop.

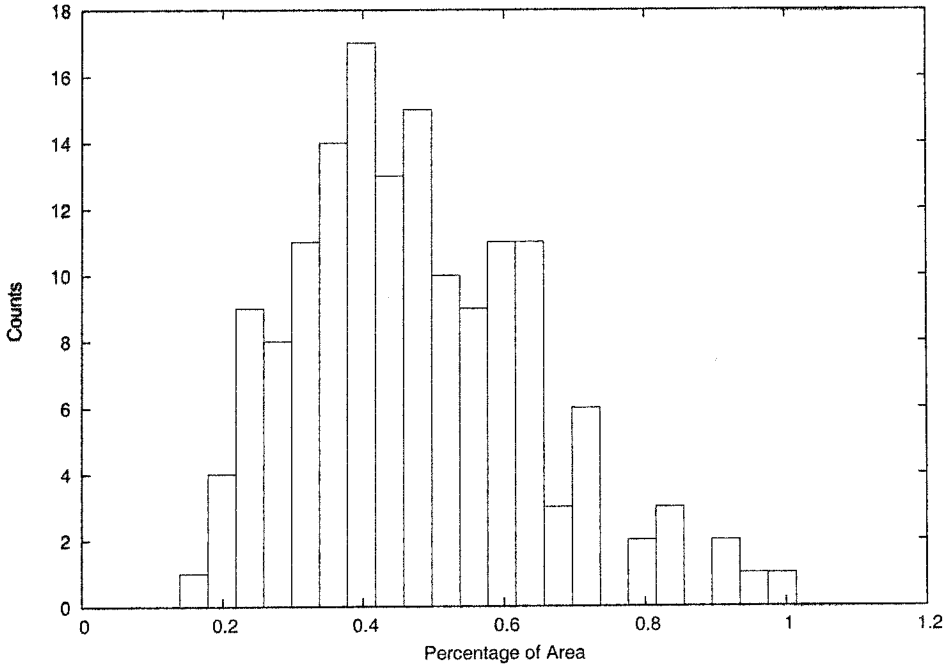


Figure 7.11: Histogram showing total area of triangles with reversed orientation on target sphere as a fraction of the sphere area, computed on a dataset of 151 subjects. The majority of the surfaces had less than 0.7% of the sphere area covered by such triangles, indicating that noninjectivity is at a tolerable level.

regions of the template, the single region of largest area is actually the unphysical cap through brain stem shown in the bottom view which is not normally considered in data analysis. Moreover, as noted above, the total region of noninjective mapping for any one individual is on the order of a few percent and is thus considered small enough to neglect.

Transferability

The experiments presented so far all use the same template and the same set of 10 test subjects. The variability as a function of penalty ratio is computed for a second set of 10 subjects. The results in Figure 7.13 show the same qualitative behaviour and the same optimal value for the penalty ratio. The optimal parameters values $r_s = 0.5$, $r_n = 2.8$, $a = 0.05$, and $w = 1$ are used for all experimental work.

Summary of Experiments

Figure 7.14 provides a visual illustration of the reduced anatomical variability in the full set of 151 individuals. The variability is reduced in all areas of the cortex, indicating that the gyral patterns are better aligned. The variability that remains is concentrated on the edges of the gyral regions. As was the case in 3D, some of this variability is the result of imperfections in the vertex classification while some is due to misalignment. At a boundary, even a misalignment on the order of the spacing between control mesh vertices is enough to change the labelling from an input surface and hence the entropy value.

Figure 7.15 shows intensity-averaged images which become sharper with tuning, a qualitative display of the improvement in aligning fine detail. Note also that the initial average images show a lot of structure: the interhemispheric and sylvian fissures are obvious, and even the central sulcus is visible. This supports the observation that there is a "loose consistency" [MKAE00] for the location of a given vertex of ASP's deformable mesh across a population. Thus it is reasonable to use the identity as the initial mapping of the auxiliary sphere, as discussed in Section 6.7.1.

7.1.2 Frontal Sulci

The structure labels described in Section 4.1.2 are also used to examine the variability of warping in 2D, after transferring the labels to the surface. The labels, delineated in 3D, are assigned to the voxels of the CSF spaces of the sulci. The surface used for registration is the inner cortical surface, which does not intersect CSF voxels,

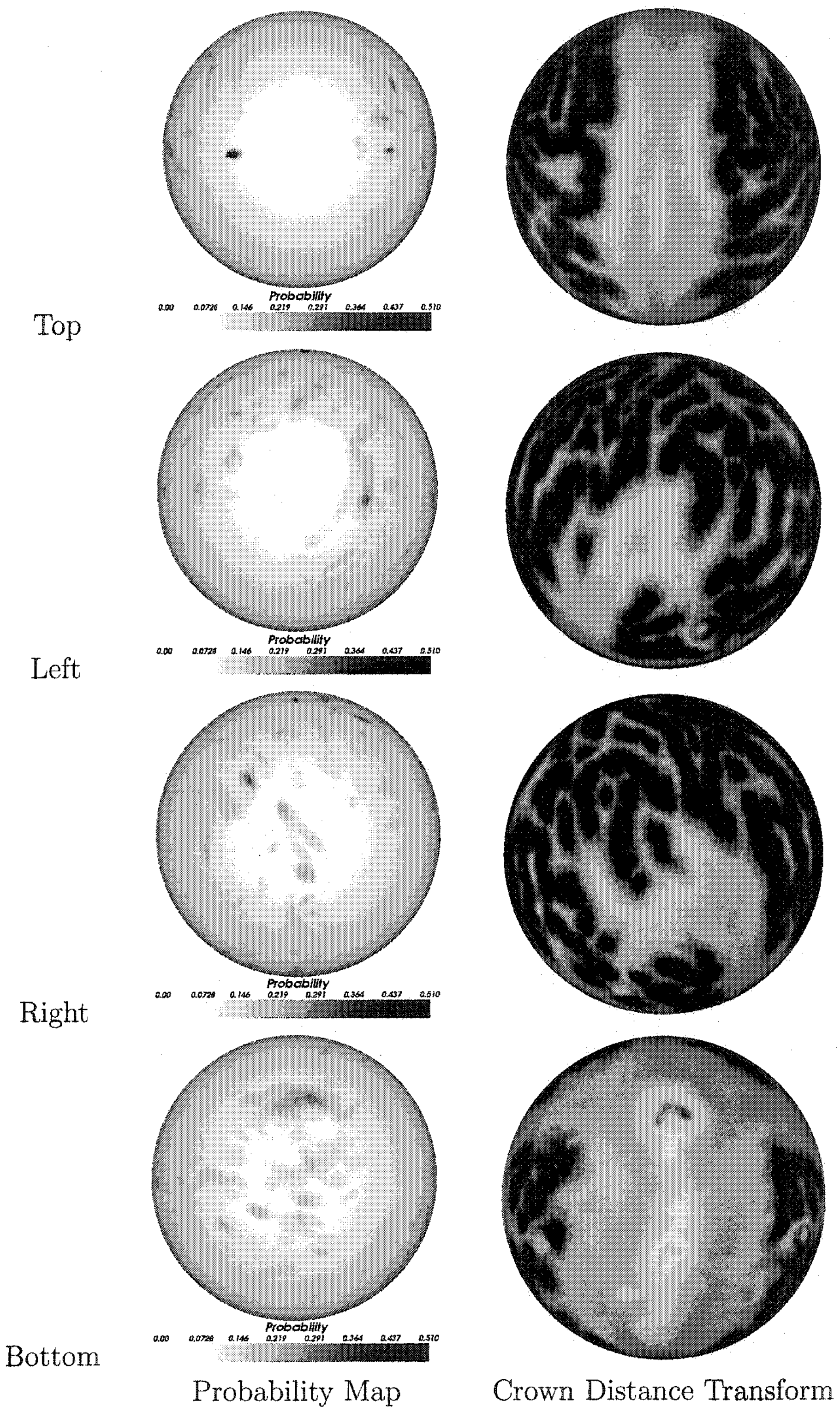


Figure 7.12: Map showing frequency of reverse orientation triangles, a measure of noninjectivity, computed from spatially normalizing a set of 151 surfaces to the template. The template gyral crown distance transform is given for reference. Notice that a large number of the reversed triangles appear on the unphysical brain stem cap shown on the bottom view, and thus can be ignored.

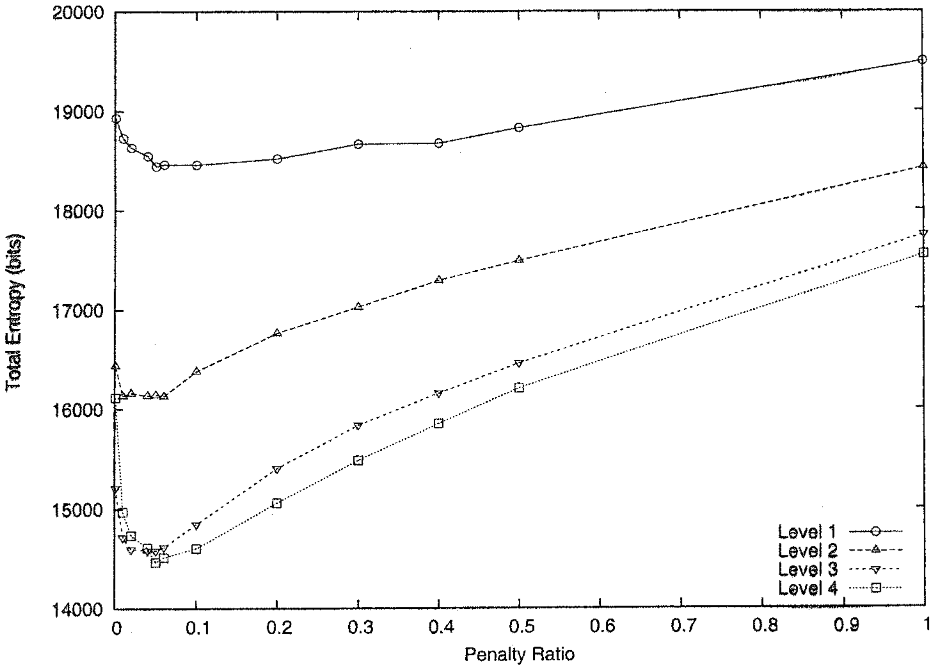


Figure 7.13: A second set of 10 subjects show similar behaviour with respect to penalty ratio. Note that the optimal value of 0.05 is the same as in Figure 7.5.

as illustrated in Figure 7.16. Thus, the labels cannot be transferred to the surface by simply sampling the voxel label information at the surface vertex location. The surface extraction algorithm used is ASP [MKAE00], which extracts both the inner and outer cortical surfaces. Moreover, the inner surface (white matter / gray matter interface) is obtained first, and then used as the initial surface for the deformation process that extracts the outer surface. Thus, there is a correspondence between vertices of the inner and outer surfaces, which is employed to transfer the labels from 3D to 2D as follows. Let v be a vertex on the inner surface for which a label is sought and w be the corresponding vertex on the outer surface. Draw a line segment from v to w and extend it by 50% past w so that it includes CSF spaces. If any voxels intersected by that segment have a sulcus label, assign that label to v. If not, repeat the same procedure for v and each neighbour of w. The idea is to sample a small cone with apex at v that points toward the CSF space. The procedure is illustrated in Figure 7.16, with the resulting labelling of the central sulcus vertices shown in the first row of Figure 7.17.

The coverage function, p_X, for a random set $X \subset \mathbb{S}^2$ is defined as $p_X(x) = \mathsf{P}(x \in X)$, analogous to the situation in 3D. The coverage function resulting from spatial normalization of 39 subjects is shown for the central sulcus in Figure 7.17.

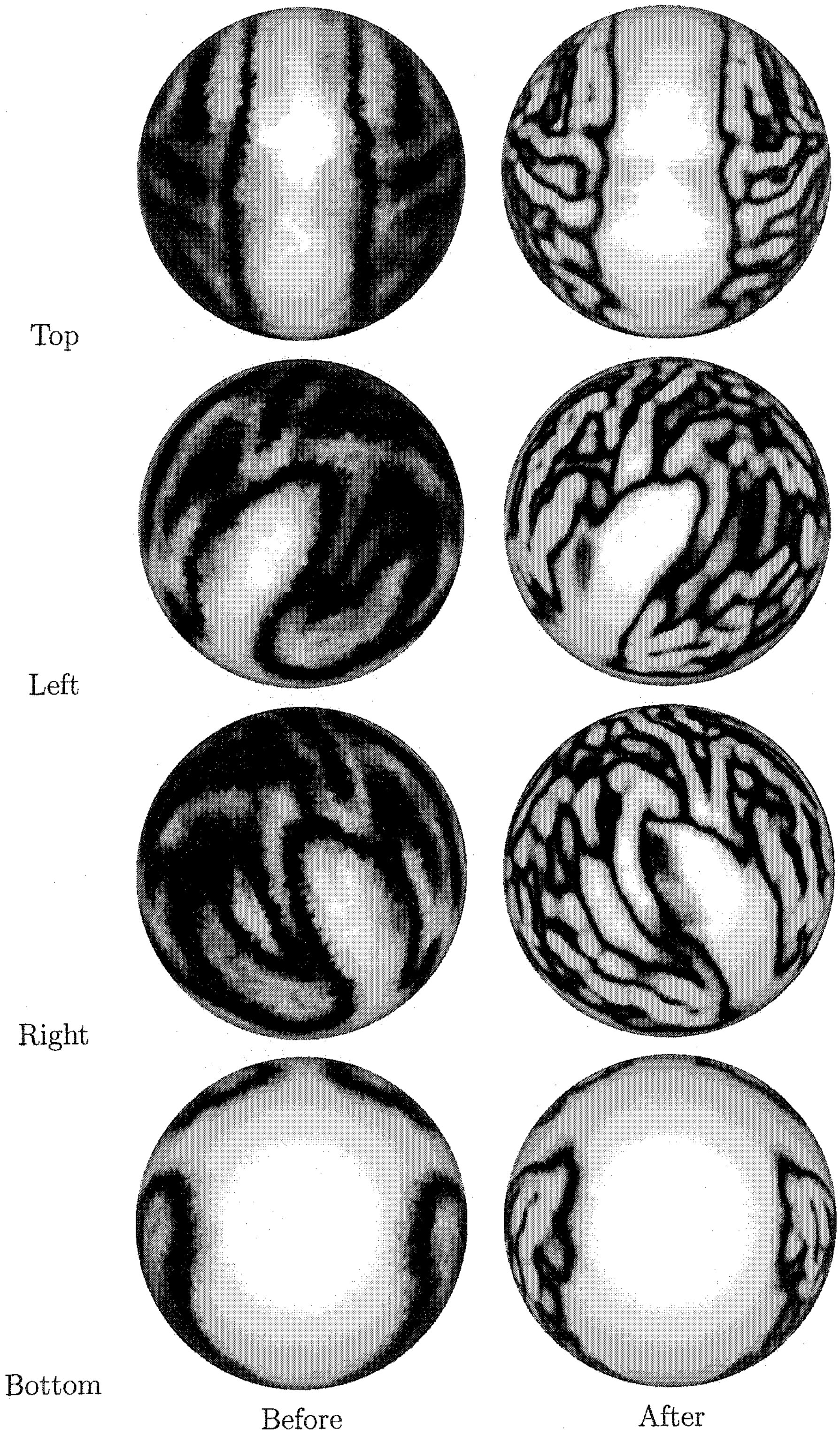


Figure 7.14: Entropy maps before and after surface registration of 151 subjects. Black indicates a vertex with high entropy (high variability). Notice that the variability is reduced in all areas of the cortex after surface registration.

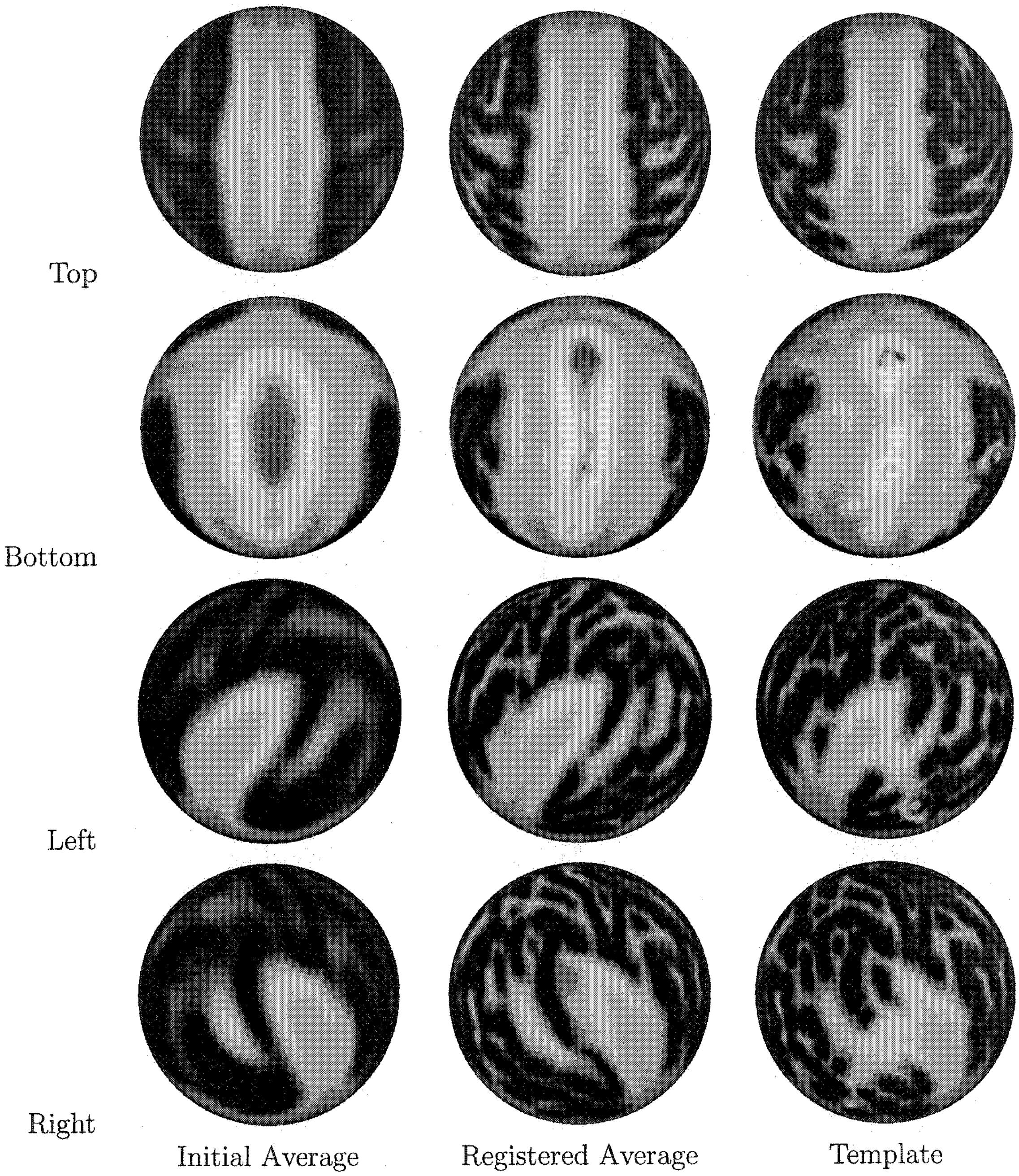


Figure 7.15: Average of crown distance transform feature data of 151 subjects shown before and after surface registration with the template data shown for comparison. Note the appearance of smaller sulci after registration, and the agreement with the template data pattern.

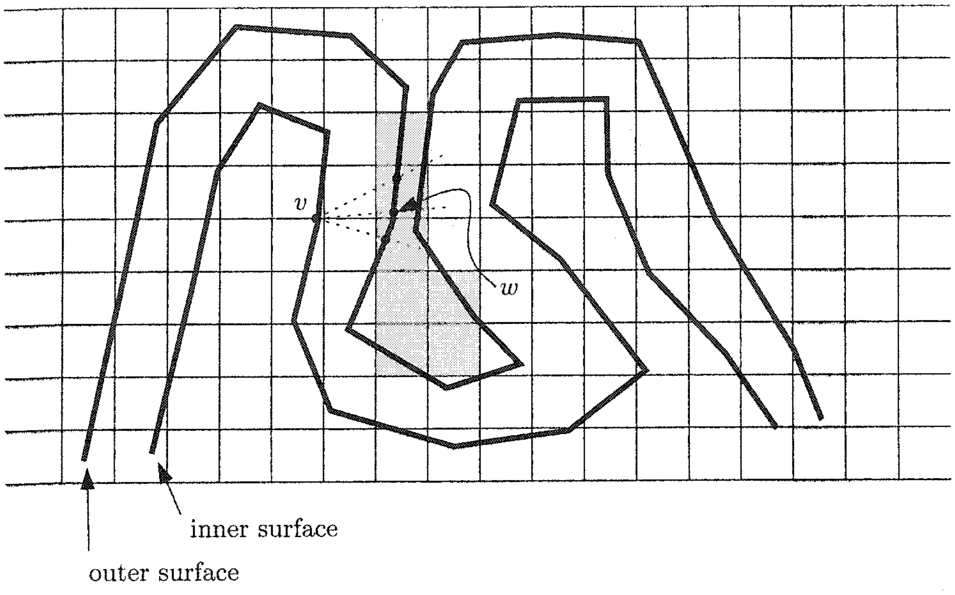


Figure 7.16: Method used to map labels from 3D voxels (shaded) onto the inner cortical surface mesh. Each voxel that intersects a line segment (dotted) with endpoint v and passing through w is checked. If such a voxel has a label, it is assigned to v. Otherwise, the procedure is repeated with v and each neighbour of w. The pairing of v and w is obtained during the surface extraction process using ASP.

7.2 Structure Variability

As discussed in Chapter 4, while the total entropy is a useful variability measure for investigating design choices and parameter values of the algorithm, it is not the best choice for measuring variability of a structure. In particular, as discussed at the end of Section 4.1.2 on page 83, the entropy of a set of sulcus labels does not capture the spatial spread of the distribution of resampled sulci. The total entropy can decrease if the input sulci are well matched to, say, two folds on the template, regardless of how widely separated in space are the two folds, as illustrated in Figure 4.15. The analogous problem can occur with surface matching.

Thus, a measure of dispersion is computed for surface structures, analogous to the measure in 3D.

7.2.1 Distance Measure

The definitions and notation of Section 4.2.1 are taken over into the unit sphere auxiliary space, $\mathbb{S}^2$. Let $d(x,y)$ denote the distance between $x, y \in \mathbb{S}^2$. A closed

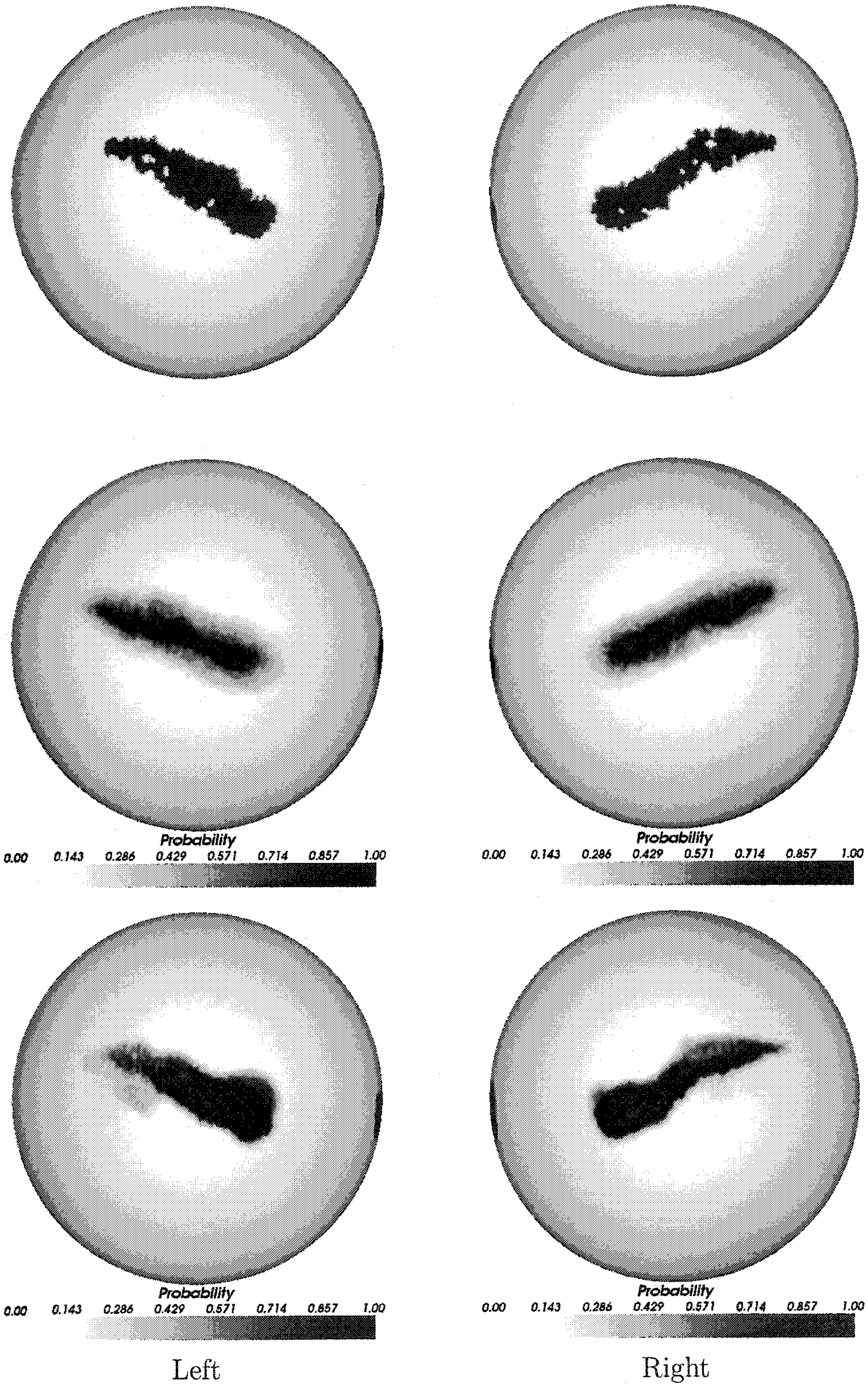


Figure 7.17: Central sulcus mapped to the surface. The top row shows the template central sulcus label set. Holes and cracks are due to imperfections in the label transfer from 3D. The second row shows the coverage function computed with 39 subjects before any surface normalization. The third row shows the coverage function after surface normalization. Notice that after registration, the region of high probability is more sharply defined. This indicates that surface normalization reduces dispersion of the central sulcus.

point set $A \subset \mathbb{S}^2$ has a distance transform, denoted d_A, defined as

$$d_A(x) = \inf_{a \in A} d(x, a),$$

computed as discussed in Section 6.2.2 on page 148. The notation A_r denotes the point set located at distance $\leq r$ from A, i.e.

$$A_r = \{x \in \mathbb{S}^2 : d_A(x) \leq r\}. \tag{7.3}$$

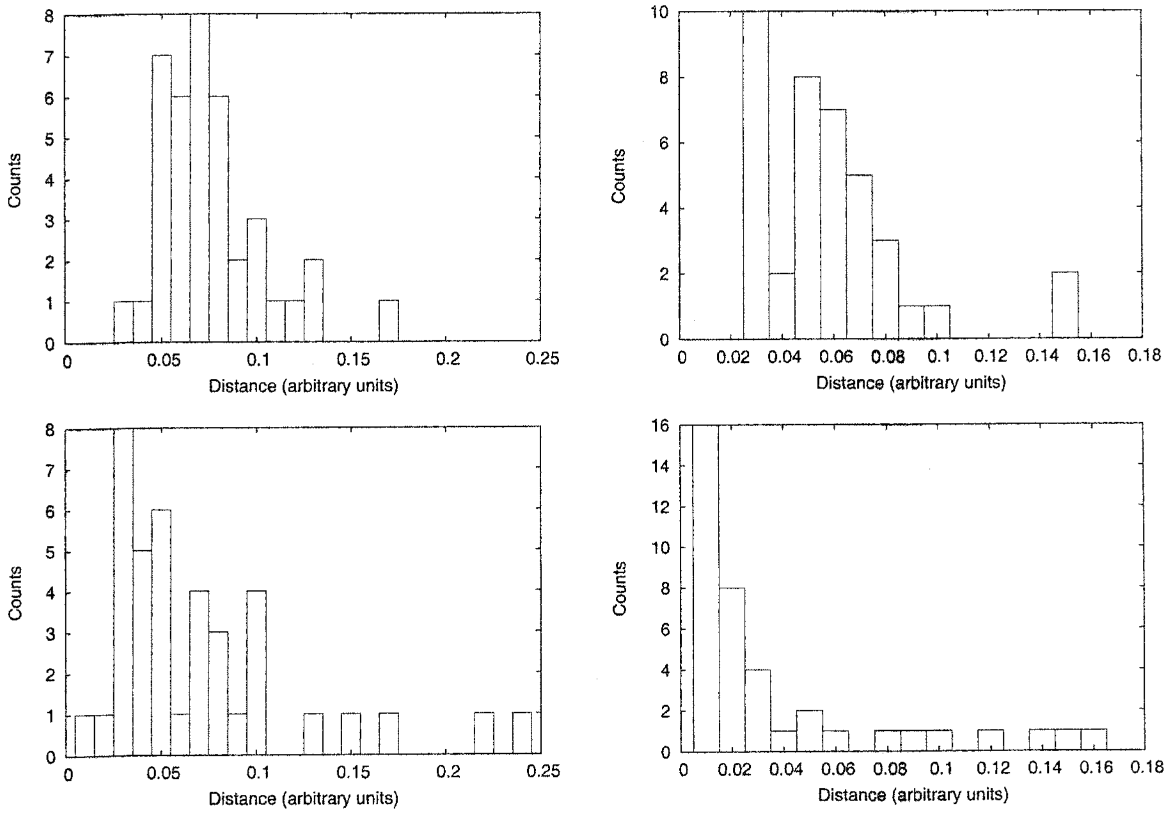
A distance between two point sets is computed using the same 90%-trimmed Hausdorff measure as described in Section 4.2.1.

7.2.2 Dispersion about the Template

The template is one of the 40 subjects for which the frontal sulcus labelling is available (the template is in fact the same as used in Chapter 4), so the distance between the template structure and each input structure can be measured. For the left central sulcus, the coverage function map of Figure 7.17 suggests that some of the inputs might end up further away from the template after normalization

The dispersion about the template sulci is measured before and after surface registration, analogous to measuring the dispersion in 3D after 9-parameter affine normalization and after nonaffine normalization using ANIMAL. As is the case in 3D, the distance to the template for some individuals decreases while others increase. As an example, consider the distribution of distance measures for the central sulcus, shown in Figure 7.18. This figure shows that after spatial normalization, while there are some exceptions, the central sulcus of most subjects becomes closer to the template central sulcus. The effect is most dramatic for the right hemisphere. Compare the left hemisphere histograms to the analogous result in 3D, shown in Figure 4.19 on page 90. The distribution in 3D is much more evenly split into two clusters than in 2D, meaning that more individuals are being matched to the wrong fold when the normalization is carried out in 3D.

The dispersion values computed for each of the four sulci are shown in Figure 7.19, both before and after surface normalization. The nonparametric paired-sample sign test [Fre92] shows that four of the eight structures show a significant decrease in median dispersion after spatial normalization, demonstrating that the surface registration succeeds in its task of aligning the inputs to the template.



Left Central Sulcus

Right Central Sulcus

Figure 7.18: Distributions of the 90%-trimmed Hausdorff distance to the template central sulcus for 39 subjects. The distances in the top row are measured before surface normalization while the bottom row shows the distances after surface normalization. Note that after normalization, while the distributions for both the left and the right sulcus are mainly shifted towards smaller values, some of the individual distances have increased.

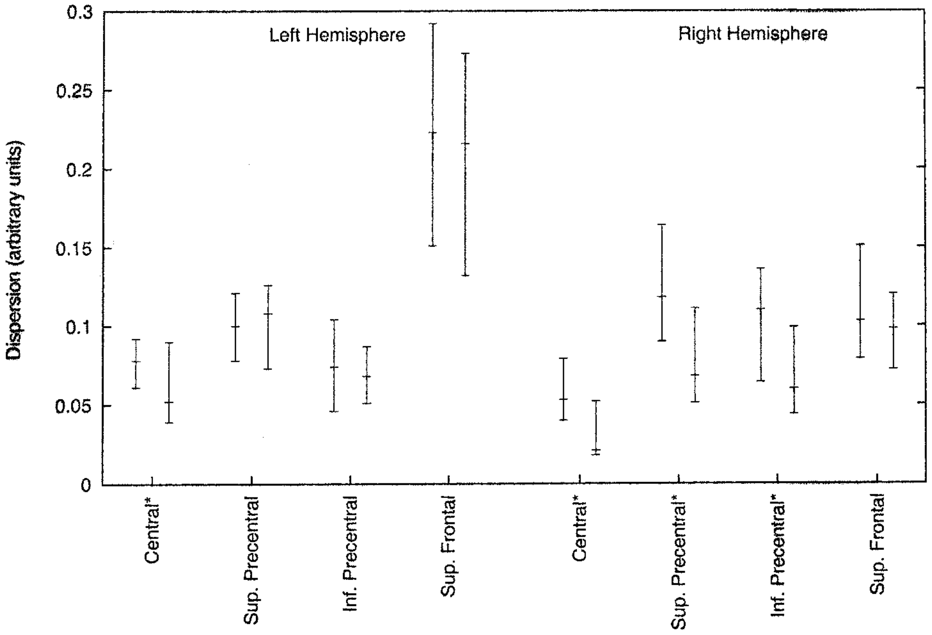


Figure 7.19: For each sulcus, the dispersion about the template sulcus is displayed before normalization (left bar of each pair) and after normalization (right bar of each pair). The central mark indicates the dispersion, which is the median of the set of distances from the template, while the error bars display the 25% and 75% quantile values, computed with 39 subjects. The sulci for which the medians differ significantly (sign test at $p \leq 0.05$) are marked by an asterisk.

Automated 2D Sulcal Labels

Another demonstration of successful spatial normalization of surfaces is shown using a second data set. Cachia et al. [CMR$^+$03] describe a method to automatically identify vertices of sulcal fundi. The sulci are automatically generated from the 3D MR image as a set of voxels obtained by skeletonization of gray and CSF voxels of the sulcus [RMPO$^+$02], the result of which is a sheet of voxels on or near the medial axis of the sulcus qualitatively similar to the manual labelling shown in Figure 4.13. The lower edge of the sheet is then projected down on to surface along the fundus of the sulcus. The result is a set of connected curves on the cortical surface, shown in the top row of Figure 7.20. The central and superior temporal sulcal fundi are obtained for 150 ICBM brains. The second and third rows of the figure display the coverage function of the two fundi before and after spatial normalization. Spatial normalization produces a much more spatially concentrated distribution that has higher probability values. The dispersion values computed for each of the sulci are shown in Figure 7.21, both before and after surface normalization. Three of the four sulci show significant reduction in dispersion, indicating that the improved alignment visible in Figure 7.20 is statistically significant.

7.2.3 Mean of a Random Closed Set

The distance mean for a surface structure is computed just as is done in 3D (Section 4.2.3). As an example, the distance mean is computed for the central sulcus after spatial normalization of the 39 subjects used to generate Figure 7.17 is shown in Figure 7.22. This mean structure is quite similar to an individual structure, e.g. the template structure in the top row of Figure 7.17.

7.2.4 Dispersion about the Distance Mean

The dispersion values about the distance mean are shown in Figure 7.23 for the 8 structures before and after surface registration. The dispersion value decreases for 6 of 8 structures after using surface normalization.

7.3 Localization Measure of Spatial Normalization

The localization measure described in Section 4.3 is used also for 2D spatial normalization. Let M be the distance mean and σ be the dispersion about the distance

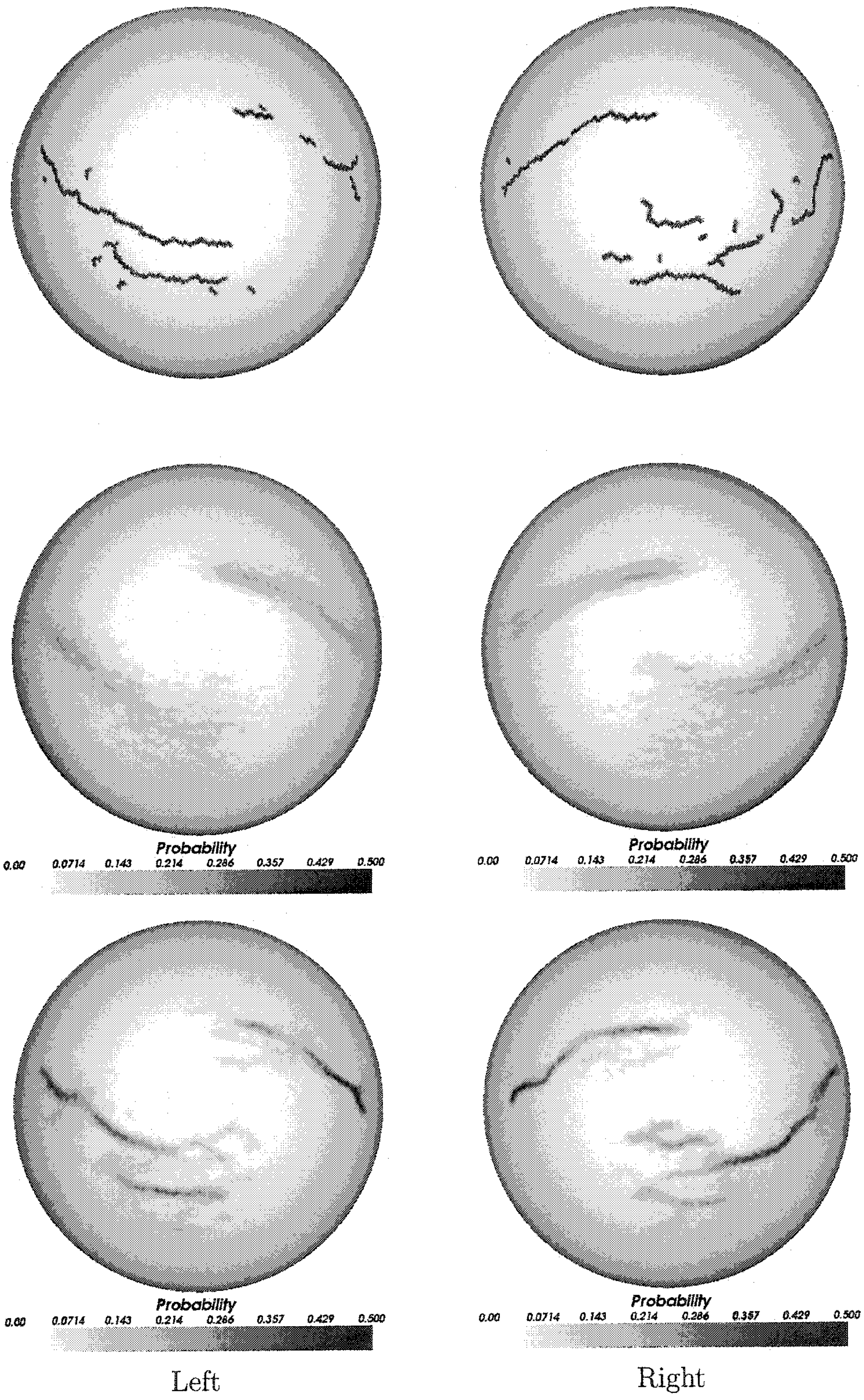


Figure 7.20: Automatically generated traces of the central and superior temporal sulcal fundi. The top row shows the traces generated for the template surface. The second row shows the coverage function computed with 149 subjects before any surface normalization. The third row shows the coverage function after surface normalization. Note that the distribution is much more spatially concentrated and the probabilities are much greater.

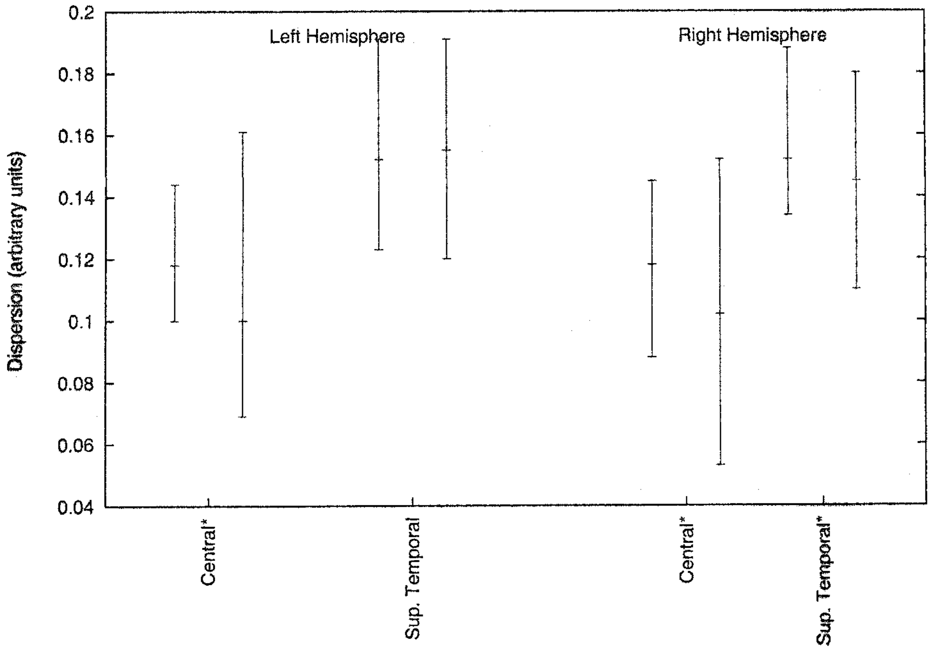


Figure 7.21: For each sulcus, the dispersion about the template sulcus is displayed before normalization (left bar of each pair) and after normalization (right bar of each pair). The central mark indicates the dispersion, which is the median of the set of distances from the template, while the error bars display the 25% and 75% quantile values, computed with 149 subjects. The sulci for which the medians differ significantly (sign test at $p \leq 0.05$) are marked by an asterisk.

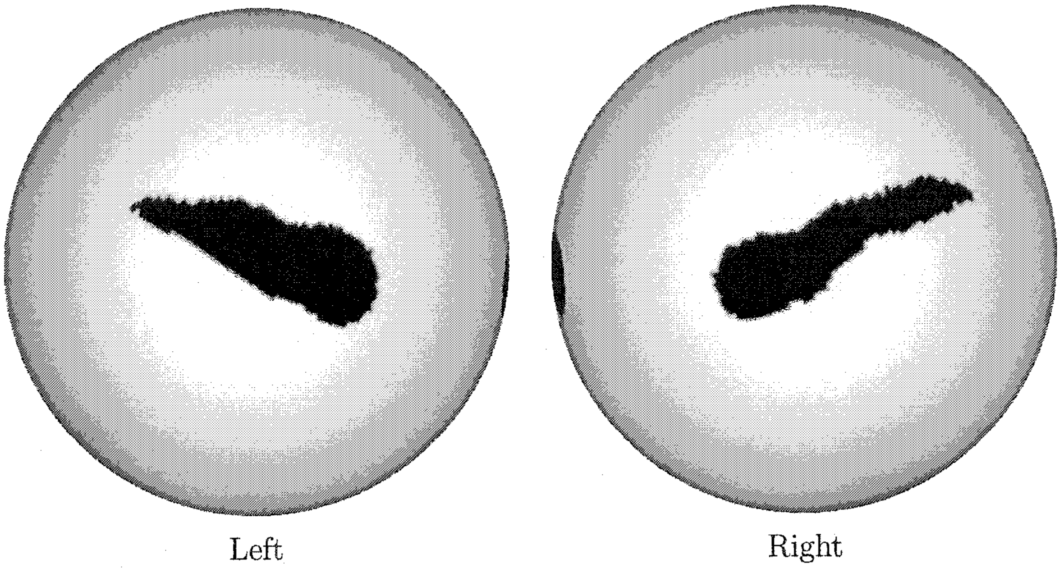


Figure 7.22: Distance mean of the central sulcus after spatial normalization of 39 individual surfaces.

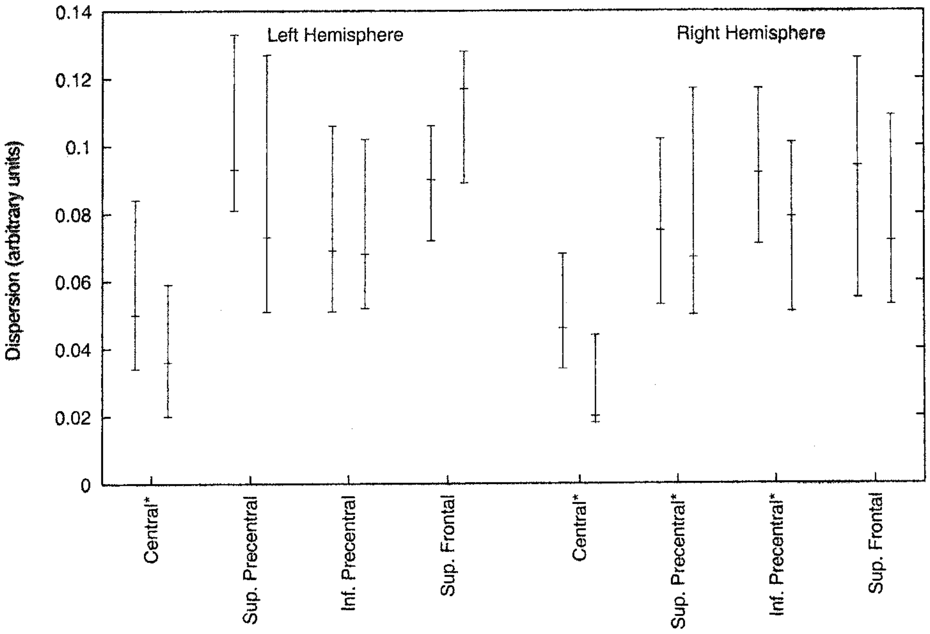


Figure 7.23: For each sulcus, the dispersion about the distance mean is displayed before normalization (left bar of each pair) and after normalization (right bar of each pair). The central mark indicates the dispersion, which is the median of the set of distances from the distance mean, while the error bars display the 25% and 75% quantile values, computed with 39 subjects. The sulci for which the medians differ significantly (sign test at $p \leq 0.05$) are marked by an asterisk.

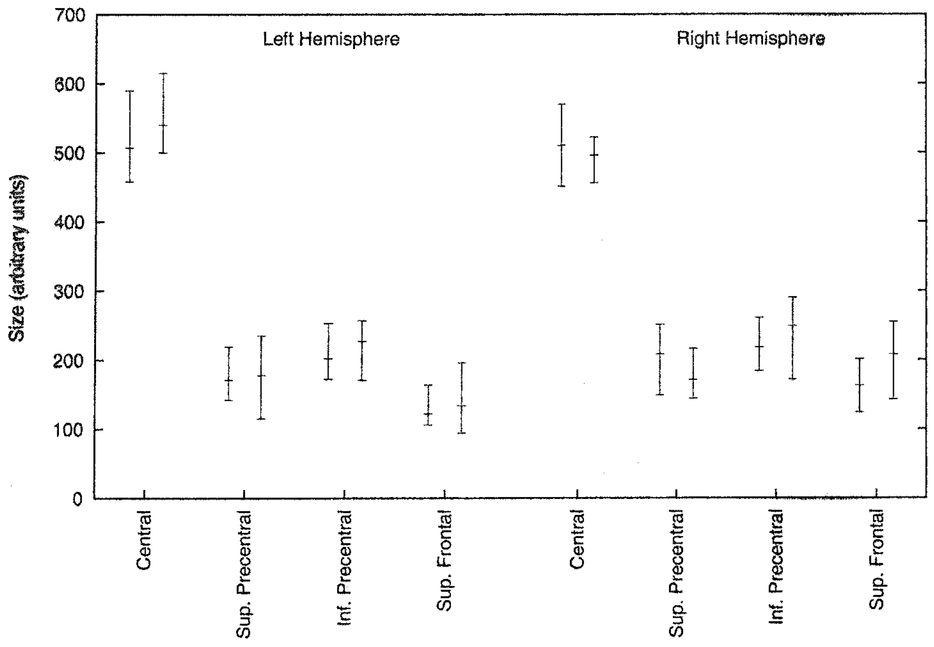


Figure 7.24: For each sulcus, the input size is displayed before normalization (left bar of each pair) and after normalization (right bar of each pair). The central mark indicates the median while the error bars display the 25% and 75% quantile values, computed with 39 subjects.

mean. The localization measure is defined to be

$$L = |M_\sigma|,$$

where M_σ is the dilation of the mean, defined by Equation 7.3, and $|A|$ denotes the size (area) of set A.

Localization, as discussed in Section 4.3, is sensitive to both the size of the mean structure and the dispersion. The size of the distance mean is given by the median size of the input structures (see the discussion around Equation 4.12 on page 95). In 3D, it is found (see Figure 4.26) that this size can be changed by spatial normalization. The equivalent data computed in 2D, shown in Figure 7.24, does not exhibit large changes in size after spatial normalization. The localization measures shown in Figure 7.25 therefore follow the same pattern of improvement (for 6 of 8 structures) as the dispersion about the distance mean shown in Figure 7.23.

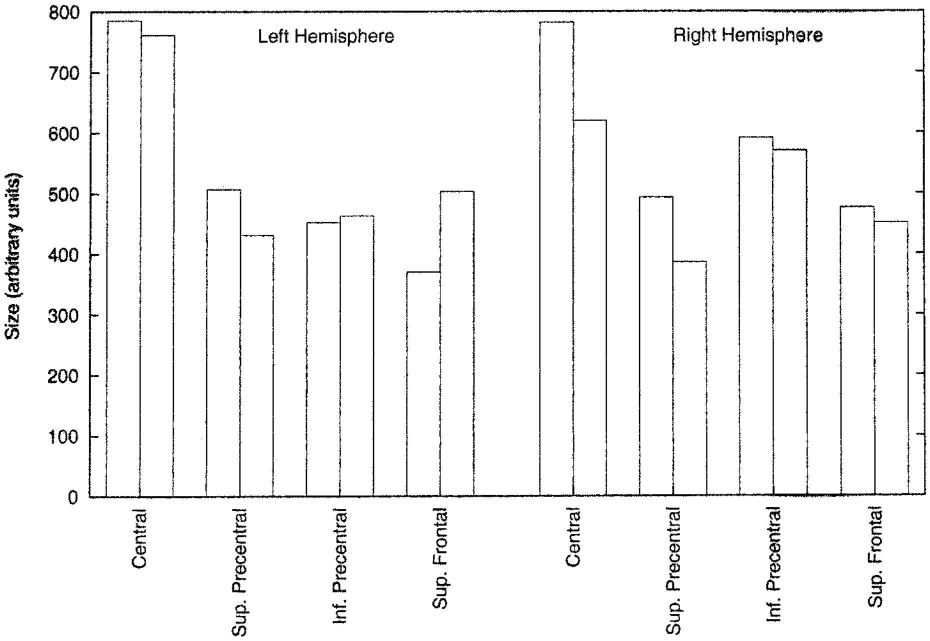


Figure 7.25: For each sulcus, the localization value is displayed before normalization (left bar of each pair) and after normalization (right bar of each pair). The central mark indicates the median while the error bars display the 25% and 75% quantile values, computed with 39 subjects.

7.4 Previous Work

Fischl et al. [FSTD99] measure localization of a structure as the area of the union of all input structures, in direct analogy to their measure of localization in 3D discussed in Section 4.3. As discussed in that section, their measure is not sensitive to dispersion. For example, if the union consists of two disjoint regions, their measure remains the same no matter how widely separated in space are the two regions. Figure 4.28 is another manifestation of the same problem.

Comparison of surface warping by Van Essen et al. [VEDJM98, VELD+01] is limited to displaying contours of the deformed template on the auxiliary surface of one subject.

7.5 Summary

The goals of this chapter are twofold. First, to reformulate the concepts and measures of variability in $\mathbb{R}^3$ presented in Chapter 4 for use on the sphere $\mathbb{S}^2$. Second, to demonstrate that the surface registration algorithm proposed in Section 6.7 does succeed in reducing spatial variability.

The concepts of segmentation and segmentation entropy transfer in a straightforward manner from 3D to the sphere. The main change is to weight the total entropy by cell area in Equation 7.2, to account for the variation in cell size. As in 3D, the segmentation entropy is demonstrably useful to probe design choices of the surface registration algorithm. In particular, a better performance is obtained using crown distance transform feature data rather than mean curvature, as shown in Figure 7.5. In addition, total entropy is able to demonstrate optimal values for the algorithm parameters: penalty ratio $a = 0.05$ (Figure 7.5), smoothing weight $w = 1$ (Figure 7.7), search radius $r_s = 0.5$ (Figure 7.9 left), and neighbourhood radius $r_n = 2.8$ (Figure 7.9 right). With these parameters, the transformation is injective except for a few percent of the target sphere area as shown in Figure 7.11. This small amount of noninjectivity is tolerated for this work.

With the aid of the second outer surface extracted with ASP, the label data used in 3D is transferred to the surface to provide similar structure labels for both 2D and 3D normalization. The concepts of distance mean, trimmed Hausdorff distance, and dispersion are straightforwardly reformulated for $\mathbb{S}^2$.

The surface registration algorithm proposed in Section 6.7 is shown to be effective in aligning anatomical structures, as evidenced by the following. The total entropy is reduced after surface normalization and the entropy maps of Figure 7.14 show that the reduction occurs all across the sphere. The sharper images of average feature data shown in Figure 7.15 is further qualitative evidence of the global feature alignment. The spatial distribution of automatically-extracted sulcal fundi shown in Figure 7.20 is more compact after surface normalization. Finally, the localization measures are lower for 6 of 8 frontal lobe sulci after normalization, as shown in Figure 7.25.

Part III

Further Experiments

Chapter 8

Template Selection

The experiments presented so far in this thesis have all used an arbitrarily-chosen individual image as the template. For the purpose of spatial normalization, the best template to choose is one that best localizes the structures of interest in the population under study. This chapter applies the localization measure of Chapter 4 to quantify the effect of template choice for 3D spatial normalization, enabling a principled template selection.

8.1 Template Generation Strategies

The registration of the template to an individual uses local optimization starting from the transformation obtained by an initial 9-parameter affine registration. This initial transformation is obtained independently of the template subsequently used for nonaffine spatial normalization. A particular structure of the template will vary in location depending on the template chosen, and thus will vary in the ease with which it can be matched to the corresponding structure in the population images. More successful matching is expected if the template's structure lies near the spatial centre of the population distribution than if it lies on an extreme of the distribution.

If an individual from the population is selected to be the template, it should be checked that it is representative of the population. One method for such a check is proposed below. Alternatively, a synthetic template that better represents the population or average anatomy can be used. A synthetic template can be generated, for example, by averaging the intensity of a number of inputs, by applying a spatial transformation to an image, or both.

8.1.1 Intensity Average

Intensity-normalized images from a number of different individuals, affinely co-registered, can be simply averaged voxel-by-voxel to form an intensity-average template [CZPE01]. Due to the normal anatomical variability in the images being averaged, fine-scale features with variable position tend to be averaged away, and only the large-scale, consistently located features remain. The left column of 8.1 shows the icbm152 image, an average of 152 brains from the ICBM database. Due to loss of fine detail, such an average image is normally considered a suitable template only for affine registration.

An intensity average can also be formed using different images of the same individual, the benefit of which is a greatly enhanced signal-to-noise ratio. An average of 27 scans of one individual, published by Holmes et al. [HHC+98], is distributed widely to laboratories around the world. Despite being a single individual, the colin27 image (middle column of Figure 8.1) might be used as a template due to the fine detail visible on it. The 1996 release of the widely used Statistical Parameter Mapping (SPM) software [SPM03] provides the colin27 image as the standard template for nonaffine spatial normalization using sine and cosine global basis functions (see Section 2.4.1). Starting in the 1999 release, the icbm152 image is used as the standard SPM template. The surface registration software from the Van Essen group [VEDD+01] uses the surface of colin27 as the standard template for surface registration [VE02].

8.1.2 Representative Individual

The second strategy examined here is to use an individual brain as the template, but choose an individual that is closest to "average" in some sense [RGW+94].

One notion of average uses the spatial transformation function itself to measure distance between images [GM98]. Specifically, if T is the transformation that registers image I to image J, then some measure of the distance between T and the identity transformation can be interpreted as the amount of effort required to warp I to J, and also be interpreted as a distance between images I and J. The distance measure on T can be as simple as the sum of squared displacements on the set of control mesh vertices V, $\sum_{v \in V} ||\Delta(v)||^2$, or it can involve derivatives of Δ [GM98].

An implementation of this distance measure, e.g. [KLT+01], requires a number of nonaffine registrations be carried out between images in the population in order to measure these distances and estimate the best template. In principle, with a sample of N images, $\{I_1, \ldots, I_N\}$, all N^2 transformations $\{T_{kl} : k, l \in 1, \ldots, N\}$,

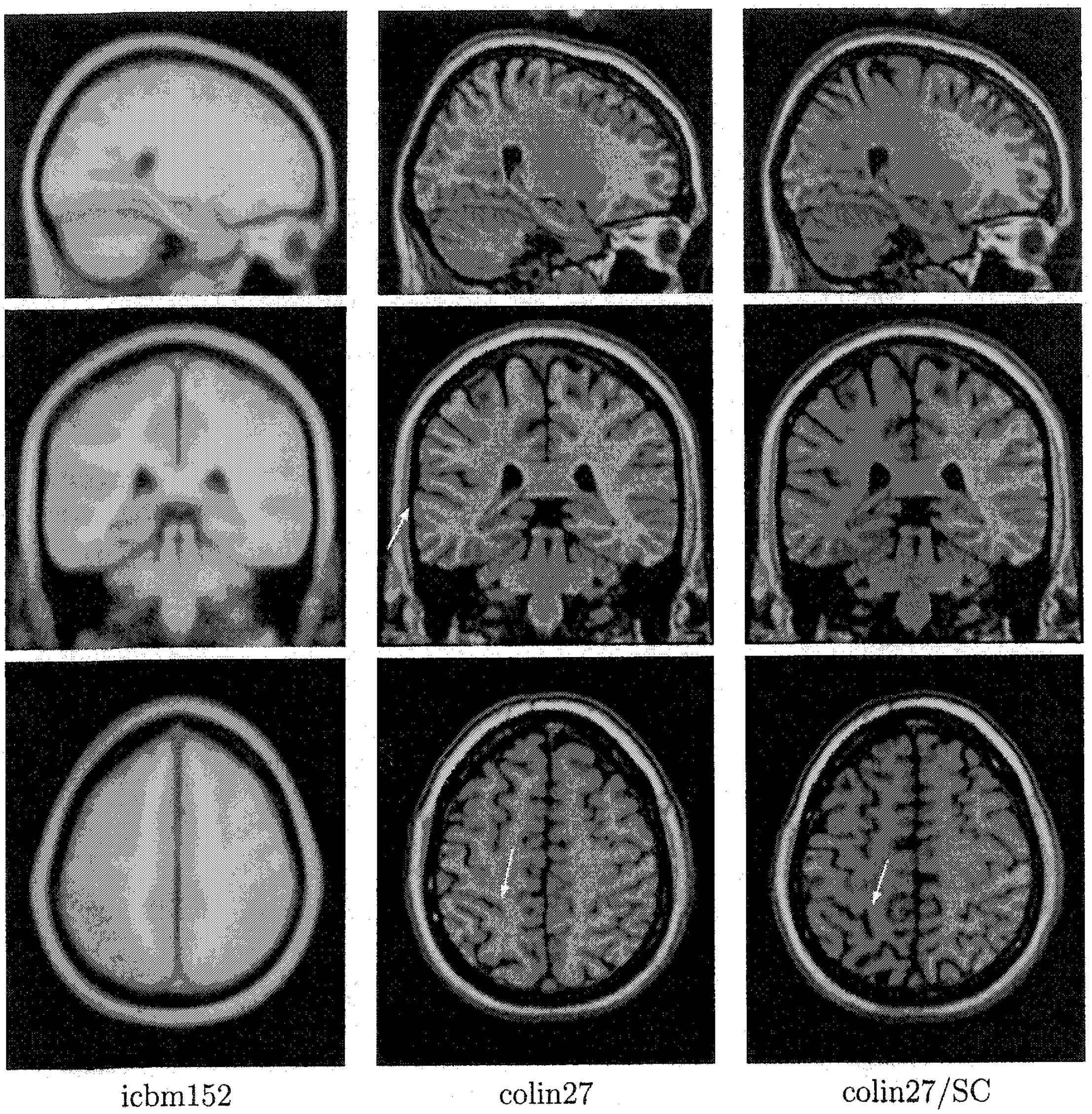


Figure 8.1: Possible templates images. The icbm152 image is the intensity average of 152 images co-registered using a 9-parameter transformation, capturing the average shape of the brain envelope. The colin27 image is a single individual with its own idiosyncratic shape. One example of the shape difference from the average is a slight bulge indicated by the arrow in the centre picture. The colin27/SC image is a slightly warped version of colin27 that corrects the brain shape. In the axial view on the second row, for example, the bulge is slightly less prominent. The warping also shifts the position of sulci; the sulcus indicated by an arrow in the third row has shifted to the posterior and a second branch is now visible on this particular slice.

where T_{kl} is the spatial transformation that matches image I_k to image I_l, need to be evaluated. Computing N^2 nonaffine registrations is an expensive proposition with even a moderate sample size, say $N = 40$. Kochunov et al. [KLT+01] suggest a reduction to just $2(N-1)$ registrations, $\{T_{1k} : k = 1, \ldots, N\}$ and $\{T_{k1} : k = 1, \ldots, N\}$. The transformation T_{kl} is estimated as the concatenation of T_{k1} followed by T_{1l}. This strategy relies on the registration method being able to successfully match I_1 to all $N-1$ other images. Otherwise, the concatenation of T_{k1} and T_{1l} is not a good estimate of T_{kl}. The template used in Chapter 4, for example, is not a good candidate for I_1, as it was successfully matched to the left central sulcus of less than half the sample images (see Figures 4.14 and 4.19). The image selected to be the intermediary I_1 should therefore be carefully chosen to be easily matched to the others in the sample set. This renders the problem circular.

The trimmed Hausdorff distance presented in Section 4.2.1 enables an estimation of proximity to a mean based on the affinely-registered image characteristics only, and is thus independent of the particular nonaffine registration algorithm chosen. For spatial normalization of the cortex, matching the gray matter is of primary importance so the distance is measured using gray matter voxels. The idea is that if the template gray matter is spatially close to the average gray matter, then the template should be easy to match to most of the images of the sample.

For each of the 152 images in the ICBM data set, the gray matter voxels are identified using a tissue classifier [Kol96]. Each set of gray matter voxels is treated as a single structure and the Vorob'ev mean of the 152 structures is computed, as described in Section 4.2.3. Finally, the 90%-trimmed Hausdorff distance between each individual and the mean gray structure is computed. The distribution of these distances is shown in Figure 8.2. ICBM subject 00244 has the minimum distance to the Vorob'ev gray mean and is used as a template in the experiments of this chapter.

8.1.3 Shape Correction

Prior to nonaffine spatial normalization, individual images are initially registered using a low-dimensional warping. In the experiments presented in this thesis, for example, the images are first registered using a 9-parameter affine transformation to the MNI305 average target [ECM+93], an intensity average of 305 images which is similar to the icbm152 in appearance. The initial transformation includes scaling along the coordinate axes intended to make all the brains similar to the average target in size. By using a slightly more flexible transformation, it is possible to improve the matching of the overall shape with the shape of the intensity average

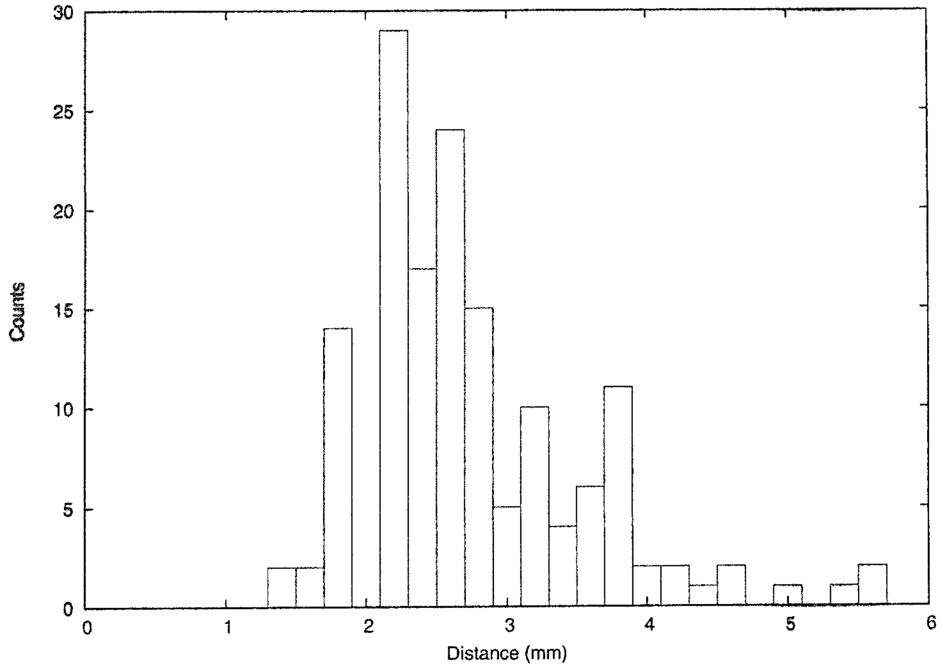


Figure 8.2: Histogram of 90%-trimmed Hausdorff distances between Vorob'ev mean gray matter and the individuals of the ICBM data set.

template.

For this thesis, a mesh warping obtained using ANIMAL at the 8 mm grid resolution (see Section 3.7) is used for shape correction. The shape correction warp is obtained using the input images smoothed using a FWHM=8 mm Gaussian kernel, so the match is mainly determined by overall shape, rather than the details of smaller sulci as the latter get blurred away. The resulting warp is very smooth.

The right column of Figure 8.1 shows slices from colin27/SC, the result of deforming the colin27 image to match the icbm152 average shape. In the axial view (second row), there is a bulge (indicated by an arrow) compared to the shape of the icbm152 image. The bulge is slightly reduced by shape correction, shown in the third column. As a result of correcting overall shape, the locations of sulci do change as illustrated in the third row. It is hoped that this shift puts the sulci in a more average position, improving the quality of the spatial normalization when using this template.

8.1.4 Warp Average

Another method proposed to generate a representative template starts with an arbitrary individual image, I_0, and then warps it into a "more average" configuration. In contrast to the shape correction method just described, this method uses a high-dimensional warping. The warp is generated using a representative sample of the population, images $\{I_1, \ldots, I_N\}$. Image I_0 is registered to each of the others, producing a set of transformations $\{T_i\}$, where transformation T_i registers I_0 to I_i. Then the average transformation is computed,

$$\overline{T}(x) = \frac{1}{N}\sum_{i=1}^{N} T_i(x), \tag{8.1}$$

and applied to I_0 to generate the template image, $I_T(x) \equiv I_0(\overline{T}^{-1}(x))$ [MBC+97, KLT+01]. Other groups, e.g. [GMT00, RFS01] take the procedure one step further, by first forming the intensity average of the resampled input images,

$$\overline{I}(x) = \frac{1}{N}\sum_{i=1}^{N} I_i(T_i(x)), \tag{8.2}$$

and then applying $\overline{T}$ to the intensity average image rather than to I_0. The template image in this case is given by $I_T(x) \equiv \overline{I}(\overline{T}^{-1}(x))$.

Figure 8.3 shows an example warp average template, both with and without intensity averaging.

8.2 Experiments

Each experiment consists of normalizing the same set of 40 ICBM subjects as used in Chapter 4 with a particular template, and computing the localization measure described in Section 4.3 for the four frontal sulci described in Section 4.1.2. Only the template varies between experiments; all other registration parameters (similarity=0.3, weight=1, stiffness=1) are identical for each experiment.

The templates examined include two individual templates. One is the ICBM subject 00244, identified as nearest to the mean gray matter in Section 8.1.2. The other is the individual template used for the initial experiments in Chapter 4, which happens to be one of the scans that make up the colin27 average image. This individual template is denoted colin1.

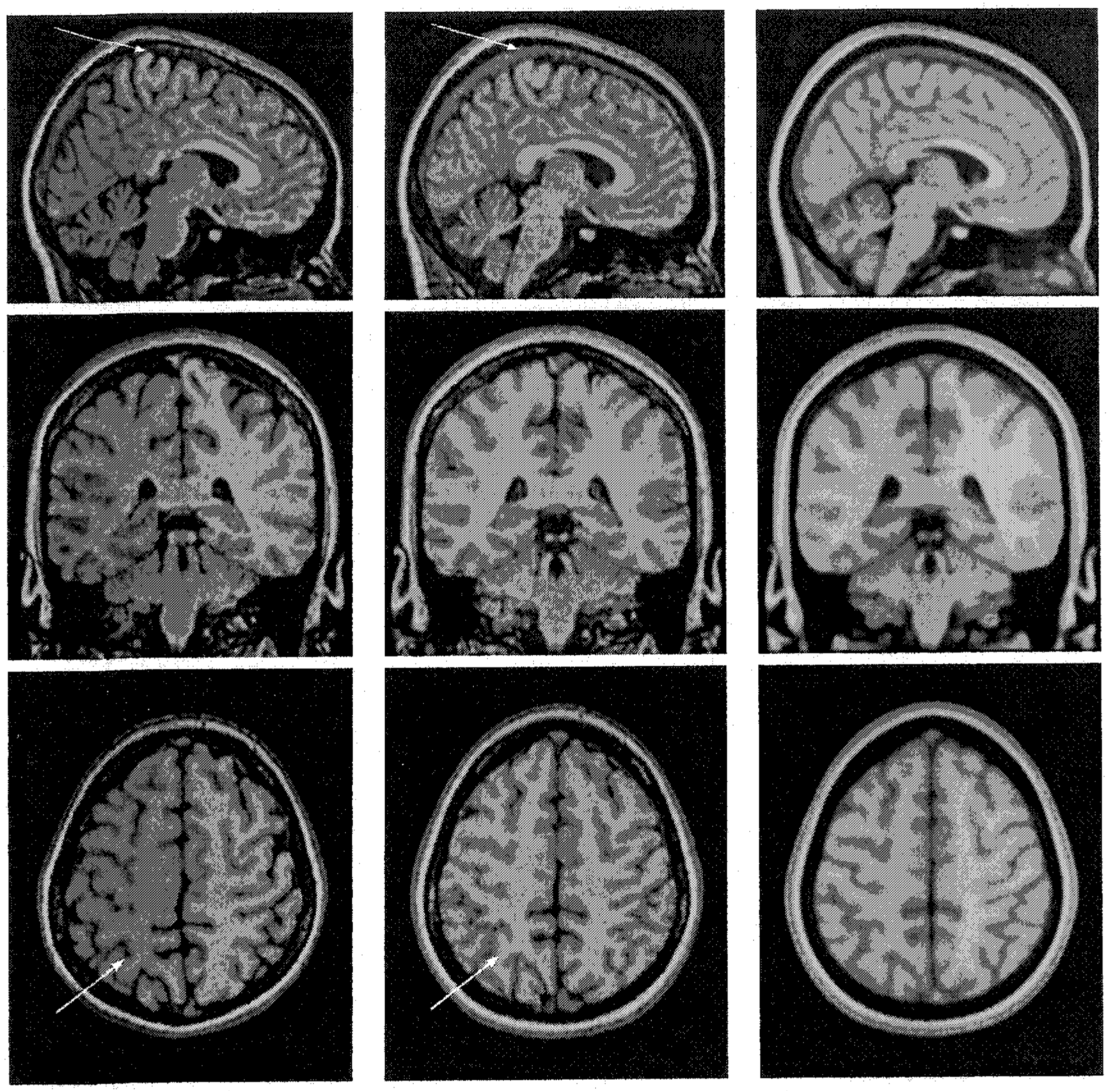


Figure 8.3: Possible template images. The ICBM subject 00244 is a single subject image. Template 00244/WA is a warp average applied to 00244. Template 00244/WA/IA is the same warp average applied to an intensity average image. See text for details. The arrows in the first row are at the same spatial position, illustrating a shift of the brain downwards at the top of the head. Similarly, the arrows in the third row are at the same position and demonstrate the shift of a sulcus towards the front of the head.

Table 8.1: Summary of the templates evaluated.

Template	Base Image	Processing
T1	00244	-
T2	00244	Shape Correction
T3	00244	Warp Average
T4	00244	Warp Average & Intensity Average
T5	colin1	-
T6	colin1	Warp Average
T7	colin27	-
T8	colin27	Shape Correction

The shape correction strategy described in Section 8.1.3 is applied to subject 00244, deforming it to match the icbm152 average shape. The warp average strategy described in Section 8.1.4 is used with subject 00244 to generate two more templates. A second sample of 40 ICBM images, different from the 40 used to measure variability, is used to generate the average warp from 00244 using Equation 8.1. This warp is applied to 00244 itself to generate one template, and the warp is applied to the intensity average of the 40 images (Equation 8.2) to generate the second template. Figure 8.3 shows example slices of these two templates.

The warp average strategy is also applied to colin1, using the same set of 40 subjects to generate the average warp as used to measure localization. The intensity average image colin27 is also tested, to measure the effect of the improved signal to noise ratio vis-a-vis colin1. The final experiment uses the shape correction strategy of Section 8.1.3 applied to colin27.

A summary of the templates appears in Table 8.1.

8.2.1 Results

The localization measures for each of the four sulci, measured after nonaffine spatial normalization using each of the eight templates are plotted in Figures 8.4 to 8.7. The localization measures using just the 9-parameter affine normalization are also given, for comparison.

The large variability for the left central sulcus when using colin1 (Template T5) is shown in Chapter 4 (e.g. Figure 4.14) to be due to the population being matched to two different folds on the template, as illustrated in Figure 8.8. The same phenomenon is observed for templates T6 (warp average of colin1) and T7 (intensity

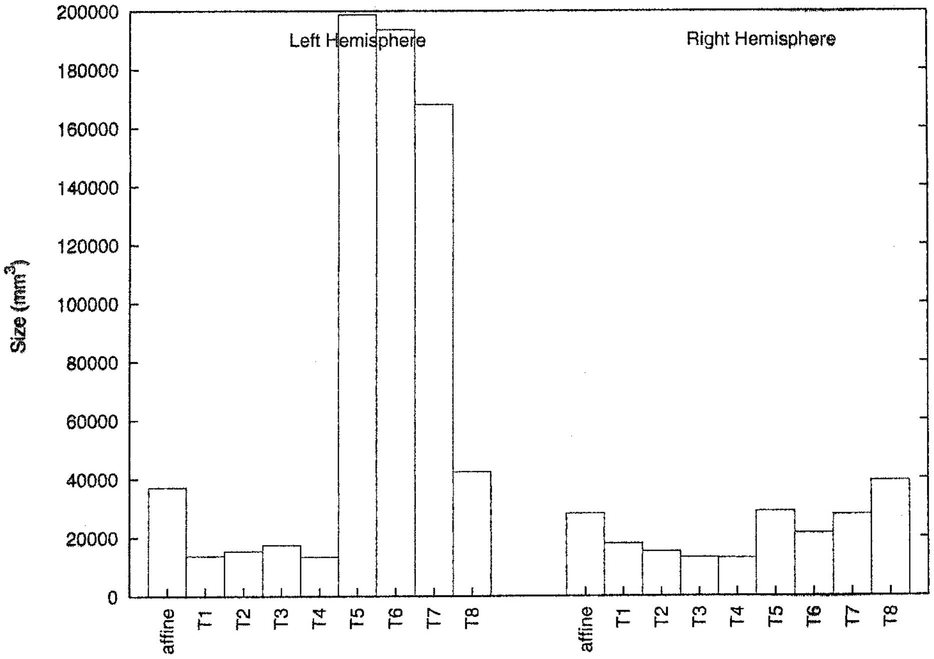


Figure 8.4: Central sulcus localization.

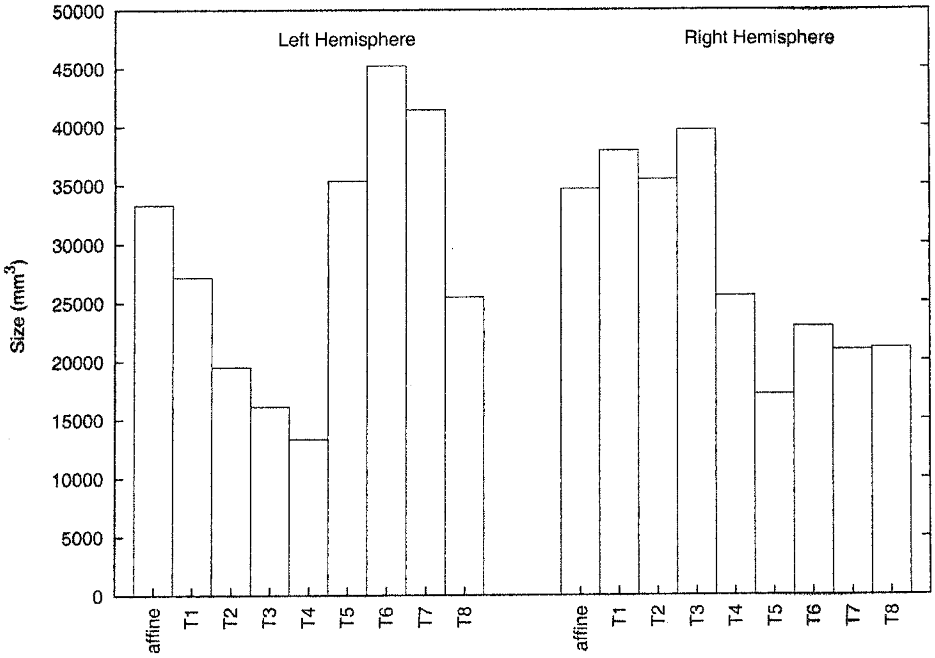


Figure 8.5: Superior precentral sulcus localization.

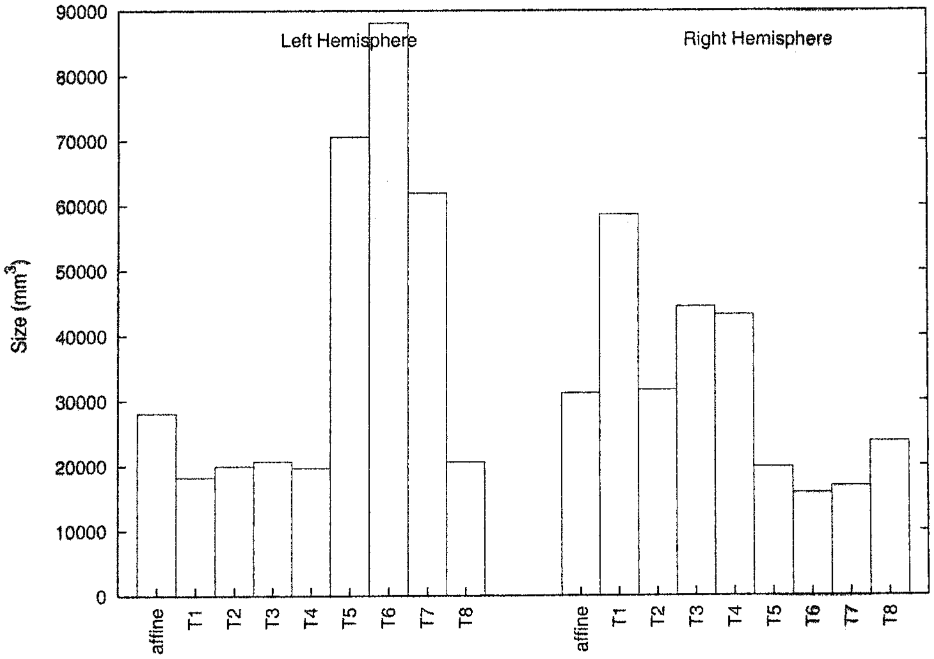


Figure 8.6: Inferior precentral sulcus localization.

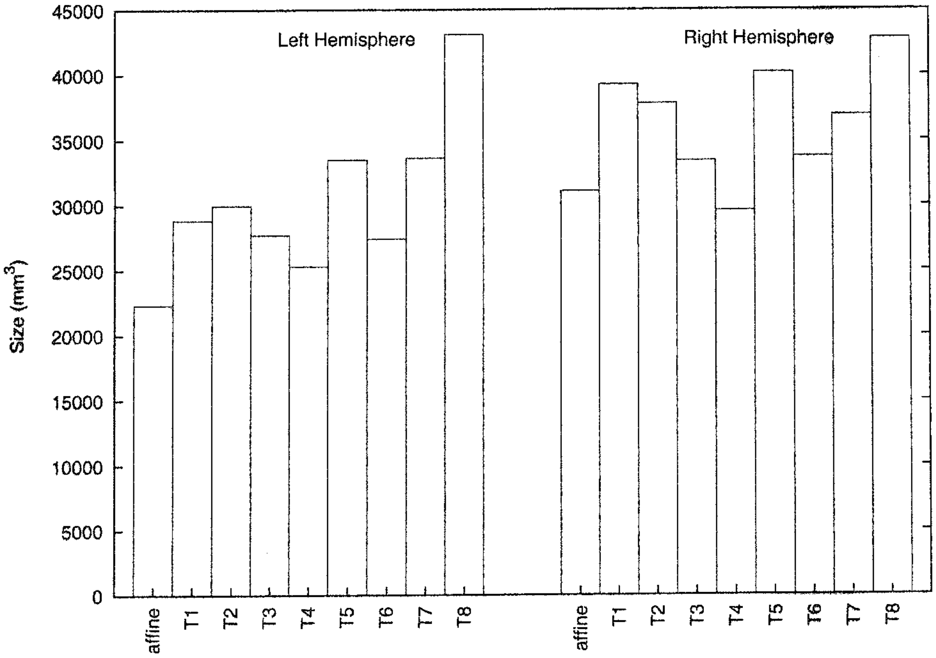


Figure 8.7: Superior Frontal sulcus localization.

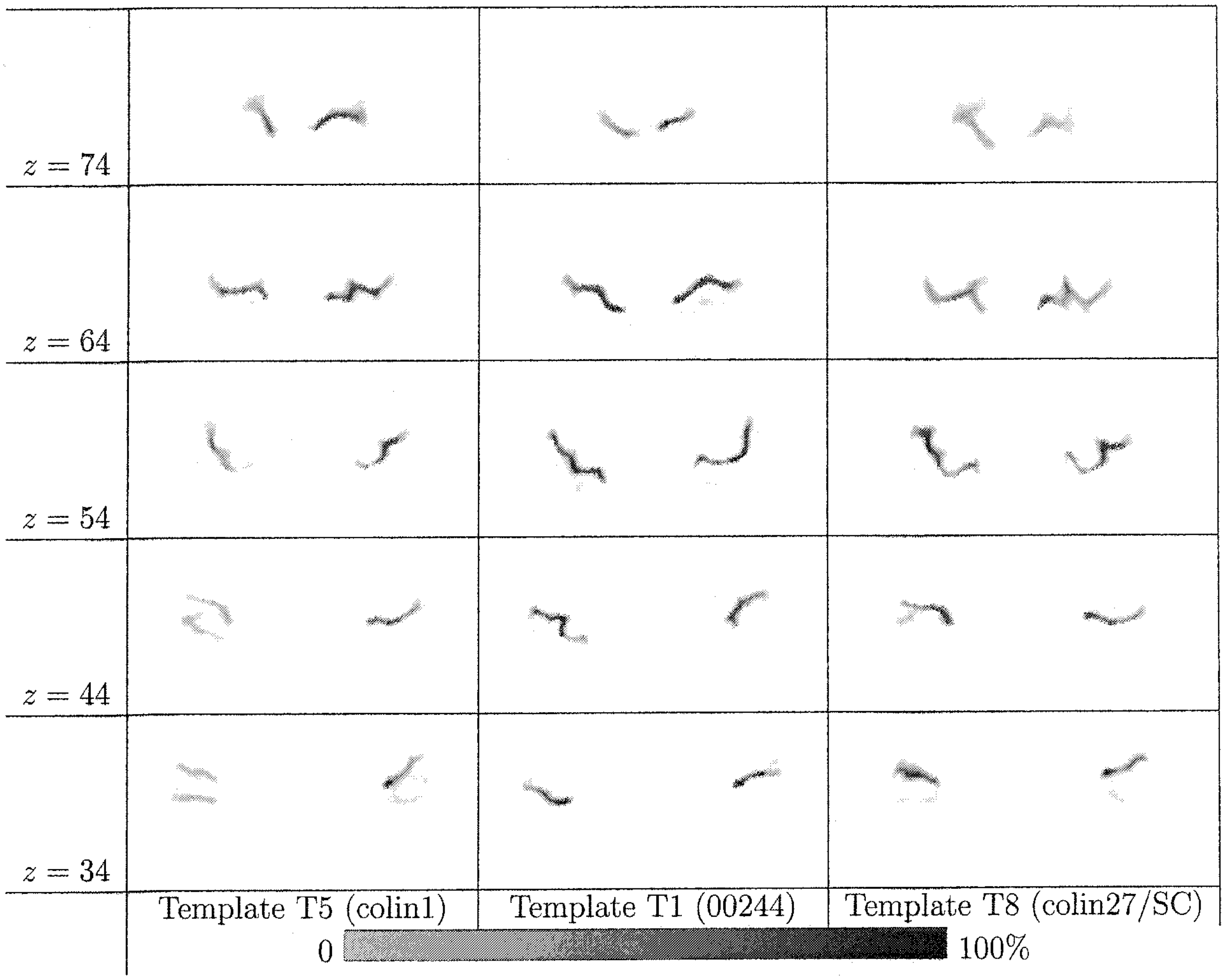


Figure 8.8: Coverage function for the central sulcus, after nonaffine normalization with three templates. The tendency of matching to two different folds is most obvious with template T5; it is reduced with T8 and is almost nonexistent with template T1.

average, colin27). The effect is much reduced using template T8 (shape correction of colin27), though not completely eliminated.

Each of the four strategies discussed in Section 8.1 is considered in turn. The comparisons are based on the localization measures of the 8 structures shown in Figures 8.4 to 8.7, where a lower score indicates a better localization.

Intensity Average

This strategy does not change at all the location of sulci in the image. Nevertheless, the expected improvement in signal-to-noise ratio [HHC+98] does have an effect on the registration.

Consider the localization values measured using the colin27 template (T7) compared to using template colin1 (T5). These templates differ only in that colin27 is an intensity average. The colin27 shows better localization for 5 of 8 structures, while colin1 is better for just 2 of the structures and 1 structure is equally-well localized with either template. This shows that the improved signal of colin27 does aid the registration. On the other hand, the sulcus pattern of colin1 and colin27 is the same so the left central sulcus continues to be matched to two folds of template colin27.

The effect of intensity averaging is also seen in the template pair T3 and T4. Template T3 is the warp average applied to image 00244, while template T4 is the same warp average applied to an intensity average image. In contrast to the colin1/colin27 pair considered above, this intensity average uses scans of different individuals. However, each individual has first been warped to image 00244. In principle, if each individual is exactly homologous to 00244 down to the smallest furrow and the warping was perfect, then each warped image would look exactly like 00244. In that case, the intensity averaging would serve the same purposes as in the case of colin27, namely to improve the signal to noise ratio [HHC+98]. However, neither is each individual exactly homologous to 00244 nor is the warping perfect. Nevertheless, the large folds do align very well and the average image has much less noise; compare the white matter tracts in the second and third columns of Figure 8.3. In addition to improving the signal-to-noise ratio, it is likely that the averaging also has the effect of blurring away some of the smaller, more variable folds. The localization values of the average (template T4) are better than the warped individual (template T3) for 6 of 8 structures while the other two are about equally well localized.

The results from both pairs of templates indicate that intensity averaging aids the registration.

Representative Individual

The effect of choosing an individual image for use as a template is compared using the arbitrarily chosen colin1 image (template T5) and ICBM subject 00244 (template T1). The latter is nearest to the mean gray matter as discussed in Section 8.1.2, with distance 1.6 mm from the mean gray matter. For comparison the colin1 scan is at distance 3.9 mm, which is on the high side of the distribution (Figure 8.2).

The localization values are better using image 00244 for 6 of 8 structures, so the 00244 image is certainly a better template than colin1. Figure 8.8 illustrates the improved matching of the central sulcus. In particular almost all of the input images succeed in matching to the correct fold.

Shape Correction

The shape correction strategy is examined twice, once using image 00244 (templates T1 and T2), and once with image colin27 (templates T7 and T8).

In the results for image 00244 (templates T1 and T2), template T2 produces better localization for 5 of 8 structures. The results for colin27, however, are improved by shape correction only for 3 structures with one more being about the same whether using shape correction or not. The left central sulcus is more likely to be matched to the template sulcus, but some inputs are still matched to the wrong fold of the template.

Warp Average

This strategy is examined twice, once using image 0244 (templates T1 and T3) and once using image colin1 (templates T5 and T6). In each case, the warp average template produces the better result for 5 of 8 structures. This suggests that the warp average is generally better as a template than the underlying image.

However, using the warp averaged colin1 template (T6) does not localize the left central sulcus very well. The bimodal distribution shown in the first column of Figure 8.8 still persists when template T6 is used. The reason for this is as follows. The transformations used to compute the average warp are computed from the domain of colin1 to each of the subjects. Consider the transformation $\overline{T}(x)$ for locations x in the left central sulcus of colin1. The template sulcus is located on one edge of the population distribution of left central sulci. Only those subjects that near to that edge of the distribution will correctly match to the template sulcus. The subjects on the other side of the distribution will not, so the warp average does not generate a displacement towards the centre of the distribution as one would hope. For the warp average to be truly representative of the population, the initial image must itself be able to match the population sulci, e.g. as 00244 does.

Overall Best Template

Normalization using template T4 shows the best overall localization for 4 of 8 structures.

8.3 Summary

Four strategies are presented for modifying an image with the goal of producing a better image for use as a template.

The first strategy is to form an average of the intensity of several inputs. This doesn't change the location of any structures in the image, but it can improve the signal-to-noise ratio of an image [HHC$^+$98] which may aid in registration. Template T7 is an average of 27 different images of the same individual and is a better template than T5, a single image of that individual. Intensity averaging is also used in conjunction with warp averaging. In this context, the average not only improves the signal-to-noise ratio, but may also remove small, variable folds thus simplifying the template image. Template T4 is the average of 40 different individuals that are nonaffinely spatially normalized and then transformed by a warp average, while template T3 is an individual with the equivalent anatomy also transformed by a warp average. The intensity-averaged T4 performs better than T3. Thus, the results from both pairs of templates indicate that intensity averaging aids the registrations.

The second strategy is to identify a representative individual image. This is implemented for these experiments by measuring the distance from a mean gray matter structure to the gray matter of 152 brains from the ICBM data set. The distribution of these distances is shown in Figure 8.2. The individual of that set closest to the mean gray matter, template T1, is a better template than an arbitrarily chosen individual (template T5).

The third strategy applies a medium-dimensional warping to an image in order to match the overall shape of the cortex somewhat better than by a low-dimensional affine matching. This strategy is of uncertain value, showing improvement in localization with image 00244 (templates T1 and T2) but not with image colin27 (templates T7 and T8).

The fourth strategy registers an initial template to the population, computes the average of these warps and applies that warp to the initial template. This strategy successfully improves the performance with image 00244 (templates T1 and T3) and also with image colin1 (templates T5 and T6).

Three of the four strategies show clear improvement in localization: intensity averaging, warp averaging, and choosing a representative individual. Indeed, the best overall performance is obtained with template T4, which was generated by applying all three of these strategies simultaneously. Thus, a good template for spatial normalization can be obtained by applying the warp average and the intensity average strategies, starting with a well-chosen individual image.

Chapter 9

Comparison of 2D and 3D Spatial Normalization

This chapter compares the localization obtained on the cortical surface after a 3D spatial normalization with the localization obtained after a 2D normalization.

In order to compare spatial normalization using a 3D method with a 2D spatial normalization, the results of one method must be transferred into the space of the other. The 2D results can be very easily transferred into $\mathbb{R}^3$, as each location of the standard auxiliary space corresponds to a location on the native surface mesh of the template which, in turn, is assigned a point in $\mathbb{R}^3$. Doing the comparison in $\mathbb{R}^3$, however, implies that the distances (e.g. dispersion) are measured without regard to the surface geometry. Since the main aim of using a 2D spatial normalization is to take the surface geometry into account, the comparison is instead performed in the 2D auxiliary space.

9.1 Mapping Functions

Let I be the template MR image, and J be the MR image of a particular subject. The labels for the subject are given in the native space of J and will ultimately be mapped to the standard space which is the auxiliary space associated with the template. Recall from Chapter 6 that the mapping from an MR image to the corresponding auxiliary 2D space (the sphere in this thesis) comprises two steps. The first step is to extract a triangulated surface mesh that follows the desired cortical boundary. For the work in this thesis, this is the inner cortical surface. The mapping from the domain of image I to surface mesh M_I is denoted E_I. The second mapping is from the mesh M_I to the sphere, and is denoted Π_I. The corresponding pair of mappings for image J are denoted E_J and Π_J, respectively. See Figure 9.1.

The 3D approach to spatial normalization computes a mapping from $\mathcal{D}_I \subset \mathbb{R}^3$ to $\mathbb{R}^3$, shown as mapping T_{3D} in Figure 9.1. Using T_{3D}, label information in $\mathcal{D}_J$ can be transferred to $\mathcal{D}_I$ as described in Section 4.1 and illustrated in Figure 4.1. From $\mathcal{D}_I$, it is a simple matter to map this label data using E_I and Π_I to the standard space, namely the sphere S_I.

The 2D approach to spatial normalization computes the mapping $T_{2D} : S_I \rightarrow S_J$. In this case, the labels in $\mathcal{D}_J$ are preprocessed to bring them onto the surface M_J, as discussed in Section 7.1.2; see Figure 7.16. From mesh M_J, the labels are mapped onto the sphere using Π_J and then resampled after spatial normalization using the approach discussed in Section 7.1 and illustrated in Figure 7.1.

The two routes for mapping from the domain of image J, $\mathcal{D}_J$, to the standard space, sphere S_I, to be compared are as follows. The 3D approach is to resample the labels onto the space of I using T_{3D}, then project the label data using E_I followed by Π_I. The 2D approach is to project the label data using E_J followed by Π_J, then resample the labels using T_{2D}.

The mapping T_{3D} is a 3D spatial transformation as discussed in Chapter 2 and computed using ANIMAL as discussed in Section 3.7. The mapping denoted T_{2D} is the 2D spatial transformation discussed in Chapter 5 and computed using the surface registration described in Section 6.7. Mappings Π_I and Π_J from the surface mesh to the auxiliary space $\mathbb{S}^2$ are each a reverse iterative deformation mapping described in Section 5.2, and used for all the 2D work in this thesis. The mappings E_I and E_J of Figure 9.1 are simply projection to the nearest surface point.

9.2 Input Images

In order to keep the comparison as fair as possible, the same template, the same population of subjects and the same set of labels are used for both normalization strategies. Based on the results of Chapter 8, the template is selected to be ICBM subject 00244, the best individual template tested. The subject population is the set of 40 ICBM subjects for which the manual frontal sulcus labels described in Section 4.1.2 are available.

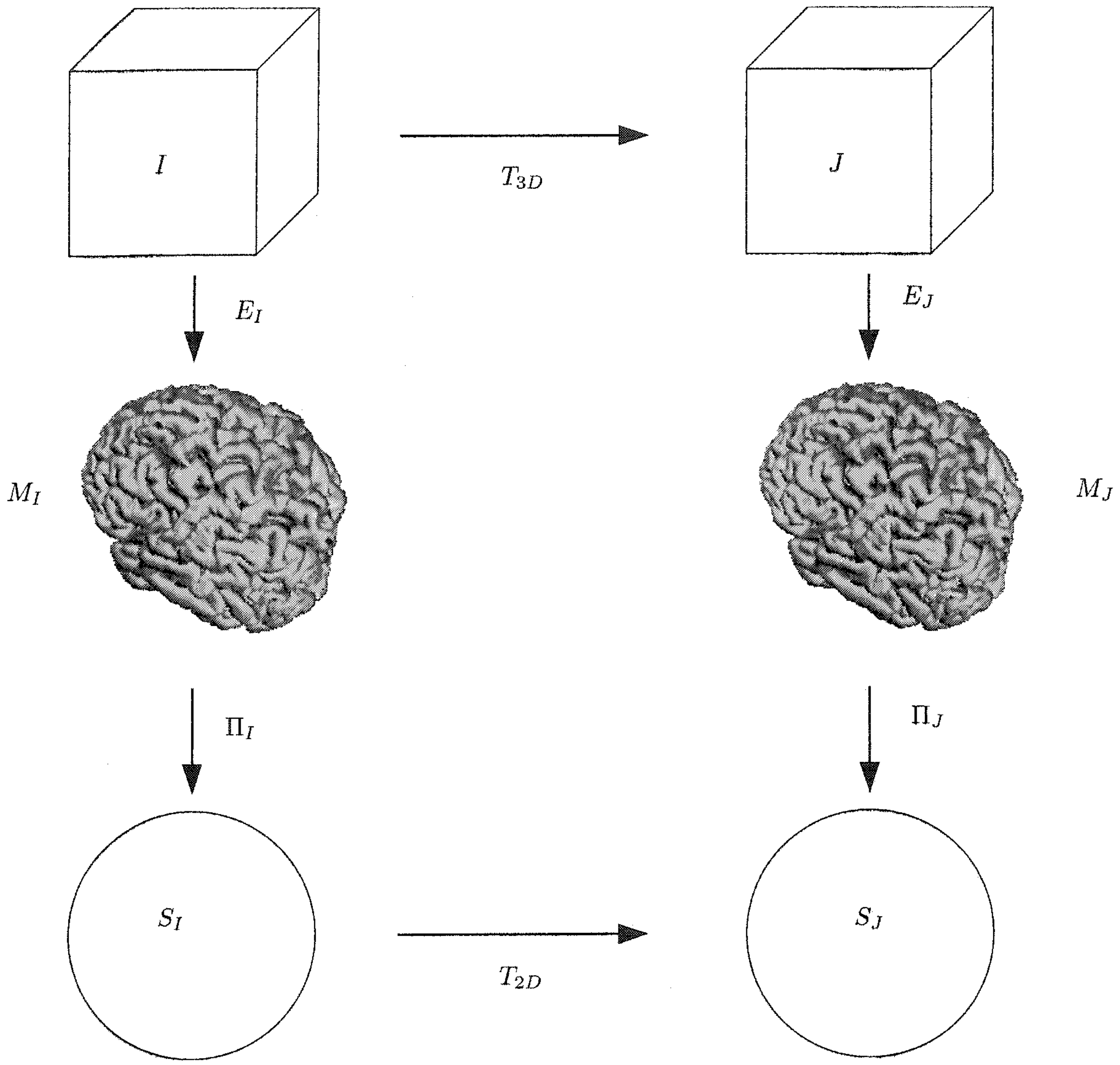


Figure 9.1: There are two routes to get label data from the domain of image J to the standard space, sphere S_I: (3D) resample the labels onto the space of template image I using T_{3D}, then project the data using E_I followed by Π_I; (2D) project the labels using E_J, followed by Π_J, then resample the label data using T_{2D}.

9.3 Label Data

The comparison is done using the same four frontal sulci as used in Chapters 4 and 7. For the experiments in the latter chapter, the labels of each subject were projected onto the corresponding surface mesh, as described in Section 7.1.2. This projection operation effectively changes the region of space that is labelled, from the original CSF space of a sulcus (as shown in Figure 4.13) to the gray matter / white matter interface bounding the sulcus.

In order that the 2D and 3D spatial normalizations compared both use the same set of initial labels, a new set of 3D labels is produced by assigning the label associated with surface vertex v to a voxel if the latter intersects any surface triangle incident to v. Figure 9.2 shows the 3D labels generated using this procedure. The same labels as they appear on the auxiliary spherical surface are shown in Figure 9.3.

9.4 Results

The results are all analyzed on the sphere using the localization measure described in Section 7.3. For all 8 structures, the localization was better when using the 2D spatial normalization, as shown in Figure 9.4.

The main reason for the superiority of 2D normalization is that the population sulci tend to be mapped onto the corresponding template sulcus. In contrast, the 3D normalization results show a large number of subjects being mapped to regions outside the template sulcus. This is illustrated by the coverage functions displayed in Figure 9.5. The result is that the 3D normalization produces a larger mean structure, which increases the localization measure.

9.5 Prior Work

Drury et al. [DECS99] examine the distribution of landmark points after a 3D or a 2D mapping. There are a number of differences between their study and the comparison done in this chapter: the auxiliary space is the plane, the labels are functional data (voxels at the location of the local maxima of a functional MR image), and the surface warping is based on anatomical landmark features [JM00]. The largest difference, however, is that the 3D spatial normalization uses a 9-parameter affine mapping, rather than a high-dimensional warping as is used here. The comparison is limited to a qualitative display of the functional landmark locations from a single subject

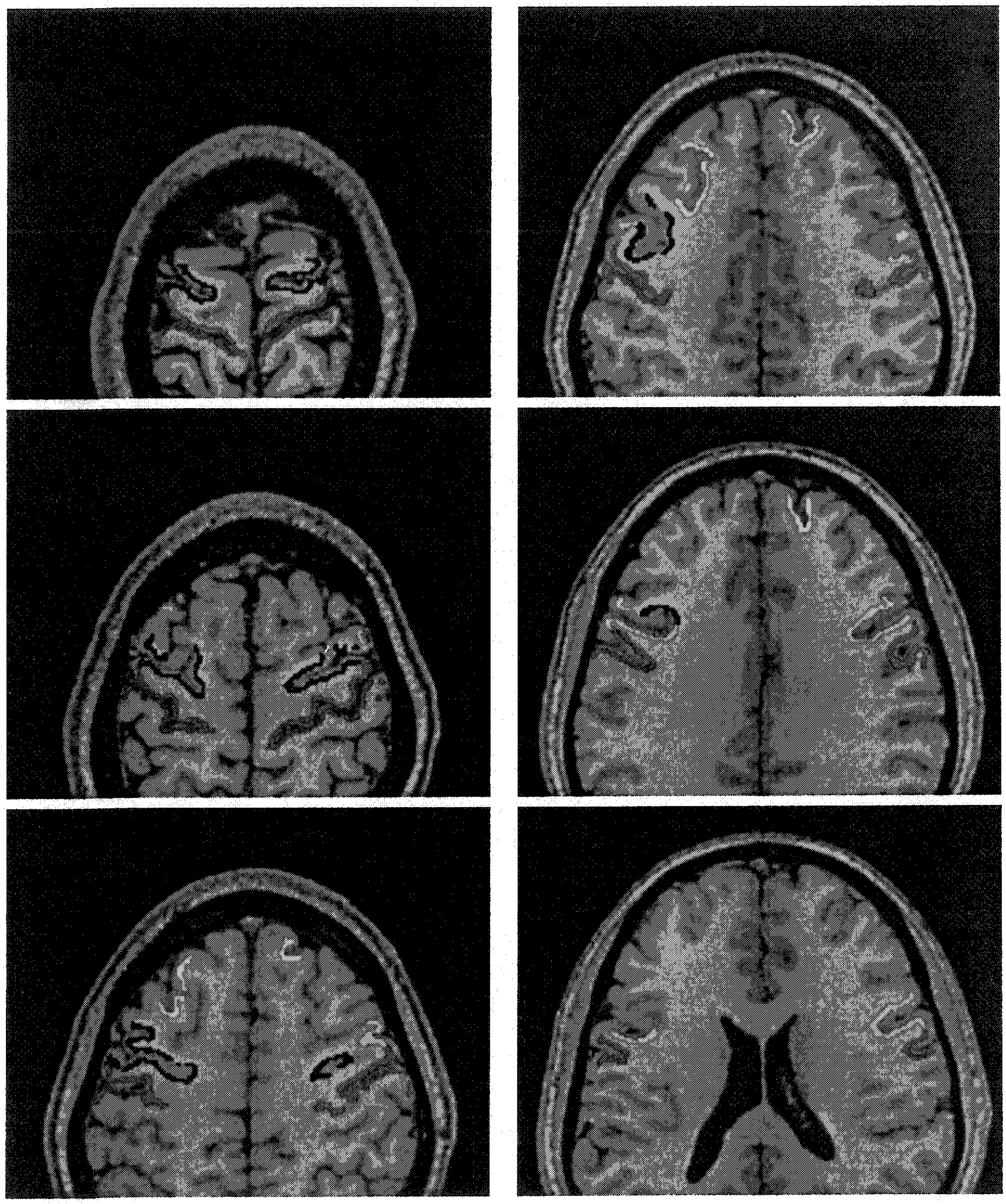


Figure 9.2: 3D labels for 3D / 2D comparison. Axial slices of one individual are displayed.

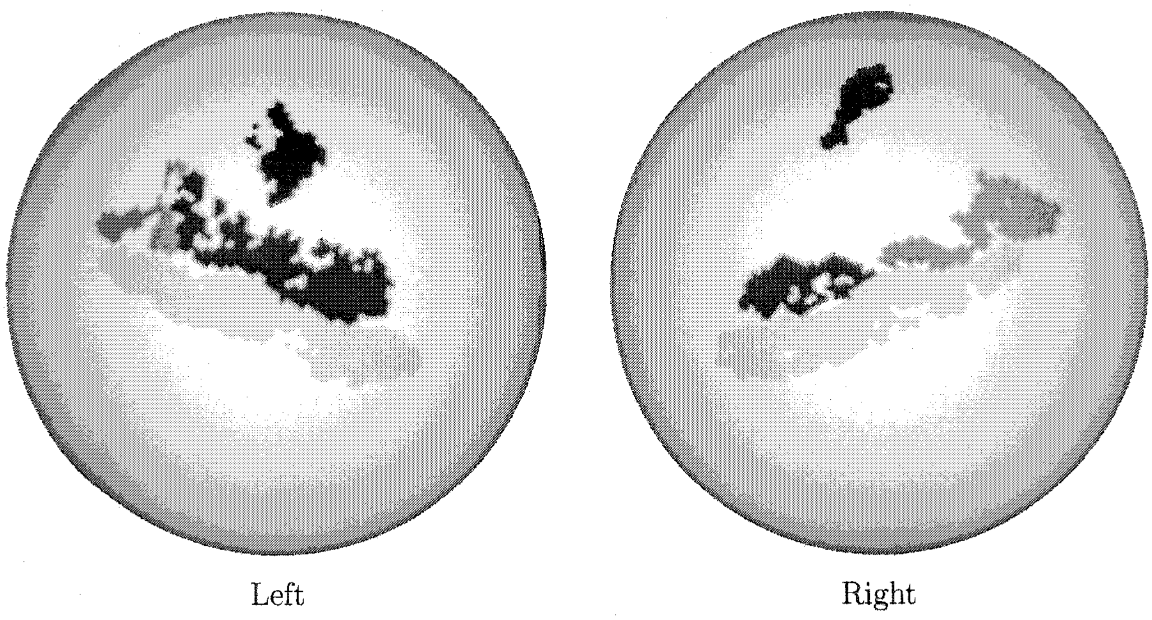


Figure 9.3: 2D labels for 3D / 2D comparison. The same individual is shown as in Figure 9.2.

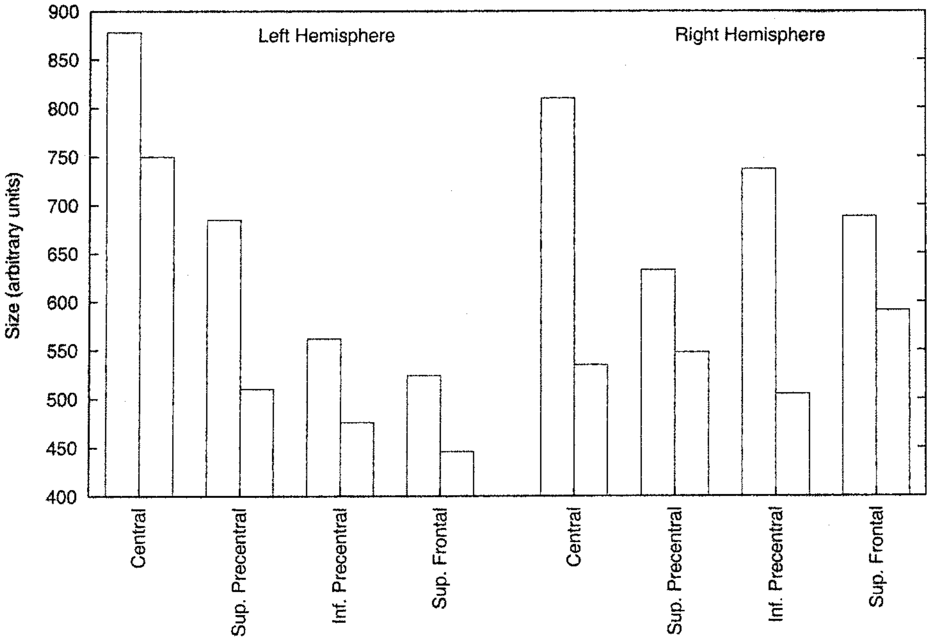
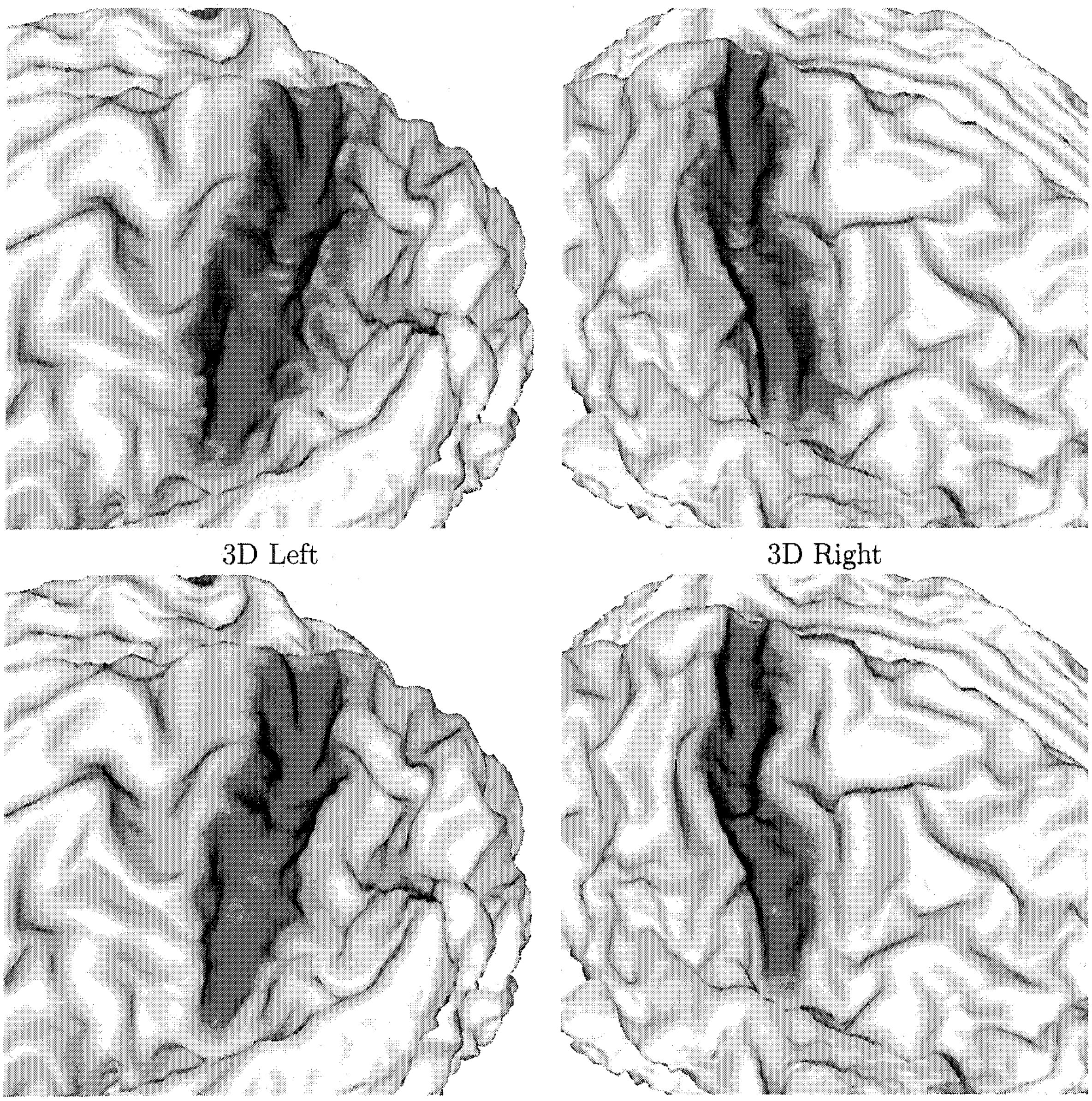


Figure 9.4: For each sulcus, the localization value on the surface is displayed for spatial normalization in 3D (left bar of each pair) and for spatial normalization in 2D (right bar of each pair). A set of 40 subjects is used for the measurements.



3D Left

3D Right

2D Left

2D Right

Figure 9.5: Coverage function shown in shades of grey (white is 0, dark grey is 1) for the central sulcus. The views are illustrated on a slightly-opened version of the template cortical surface in order to better view inside the sulcus. Note that the shading away from the central sulcus is due to the illumination in the 3D rendering process used to generate these images. After warping in 3D (shown in the top row), many vertices on the bank of the neighbouring sulci have nonzero probability of being mapped to the central sulcus. This is particularly evident for the left hemisphere. In contrast, the warping in 2D localizes the central sulcus nearly completely inside the template central sulcus.

displayed on the flat map after using a 2D or a 3D warping. The 2D pattern is quite different from the 3D pattern, which largely reflects their observation that the location of sulci "was not adequately compensated by the transformation to Talairach space" [DECS99]. In other words, the low-dimensional 3D spatial normalization failed to adequately localize the sulci.

Fischl et al. [FSTD99] also compare their own high-dimensional 2D normalization to a 9-parameter affine 3D spatial normalization [CNPE94]. They do not compare the two spatial normalization procedures in a common space, however. Rather, the localization measure after 3D normalization is computed in the standard 3D space as $V/\overline{V}$, where V is the volume of the union of the 3D structures, and $\overline{V}$ is the mean volume of the input structures. The localization measure after 2D normalization is computed in the standard 2D space as $A/\overline{A}$, where A is the area of the union of the 2D structures, and $\overline{A}$ is the mean area of the input structures. Since the ratios are dimensionless these two measures are comparable, though they are measuring localization in two different spaces. Also, recall from Section 4.3 that this measure doesn't account for dispersion. Nevertheless, they show that warping in 2D produces smaller localization scores for three sulci and five functionally-defined regions.

9.6 Summary

This chapter compares the localization on the surface obtained for four sulci using either a 3D spatial normalization or a 2D spatial normalization. On a data set of 40 subjects, the 2D normalization produces a better localization for all four sulci in each hemisphere as shown in Figures 9.4 and 9.5.

In contrast to the previous work of Drury et al. [DECS99] and of Fischl et al. [FSTD99], the 3D normalization is carried out using a performance-tuned (see Chapter 4) high-dimensional warping algorithm (ANIMAL, described in Section 3.7) rather than a 9-parameter affine transformation. Moreover, the template chosen is demonstrated in Chapter 8 to localize the sulci very well. For example, the central column of Figure 8.8 shows that virtually all the subjects map the central sulcus onto the template central sulcus. Compare this to the variability remaining after a 9-parameter affine normalization, shown in the left column of Figure 4.14. The comparison performed in this chapter is therefore a much more sensitive test as to the relative performance for spatial normalization in 3D versus 2D, than previous work.

Chapter 10

Conclusions and Future Work

10.1 Summary

A 3D spatial normalization method is the combination of a template image and a registration algorithm that matches the template to an input image. Similarly, a 2D spatial normalization method is the combination of a template surface and a registration algorithm that matches the template to an input surface. The registration algorithms are put into a common framework of optimization in order that they may be compared on the basis of the data and model terms employed in the objective function, $\Phi(I, J, T)$.

10.1.1 3D Normalization

Chapters 2 and 3 summarize and synthesize the literature on 3D medical image registration, concentrating on high-dimensional nonaffine transformations. Chapter 2 argues that a successful spatial normalization of the cortex requires a high-dimensional warping, such as the standard mesh warping (Definition 2.4.1). Chapter 3 uses the framework of optimization in order to compare different registration algorithms on the basis of the data and model terms. At the present time the data term best suited for automated spatial normalization is one that performs matching based on image intensity alone or in combination with label data (Section 3.8.1). The experiments in this thesis concentrate on obtaining the best results using intensity alone. In the maximum likelihood framework, the form of the data term corresponds to an assumption about the intensity relationship between registered images. The intensity transformation between MR images could plausibly be a linear or an affine intensity transformation, though the latter may better match two tissue classes (Section 3.2.2).

Though the principle of coordinate invariance (Section 3.1) does place some limits on the allowed differential regularization terms (Section 3.3.2), the model terms suitable for registering brain images do not have as well-developed a theory as the data terms do. Moreover, the objective function is often expressed as a sum of data terms Φ_D and regularization terms Φ_R,

$$\Phi(I, J, T) = \Phi_D(I, J, T) + \lambda\Phi_R(T),$$

and no theory exists to guide the selection of the parameter λ. Thus the segmentation variability measure described in Section 4.1 is proposed in order to evaluate registration algorithm performance in the context of spatial normalization. This enables an objective selection to be made between competing design choices, for regularization terms, data terms, and other features of the algorithm (e.g. the node thinning heuristic of ANIMAL, Section 3.7.2), as well as for selecting an optimal value for numerical parameters such as λ.

The 3D experiments in this thesis use the ANIMAL algorithm (Section 3.7), which has the following characteristics:

- the transformation is parameterized as a mesh warping with cubic interpolation (Definition 2.4.1),
- the matching is driven by smoothed intensity and smoothed intensity gradient features,
- the data term is a regional version of cross-correlation (Section 3.2.2),
- the model term is implemented as an elastic smoothing of the transformation in a two-step algorithm (Section 3.4.2),
- control mesh vertices do not participate in the data term if the intensity value is below a threshold (the node thinning heuristic, Section 3.7.2), and
- a four level coarse-to-fine hierarchy is used:

    1. control vertex spacing of 8 mm, FWHM=8 mm smoothed image data,
    2. control vertex spacing of 4 mm, FWHM=8 mm smoothed image data,
    3. control vertex spacing of 2 mm, FWHM=4 mm smoothed image data,
    4. control vertex spacing of 2 mm, FWHM=4 mm smoothed image gradient data.

In addition to the design elements listed above, there are several numerical parameters used by the algorithm, as discussed in Section 3.7 and in Section 4.1.1 on page 72. Using the total entropy as a performance measure for ANIMAL in the context of spatial normalization, the following changes are shown to produce improved performance (Section 4.1.1): omit node thinning, omit gradient fit, use correlation coefficient as the data term, and use parameters similarity=0.3, stiffness=1, weight=1. Figure 4.10 shows the quantitative improvement of these changes in terms of reduced entropy throughout the brain while Figure 4.11 shows the improvement qualitatively in terms of increased sharpness in the intensity average image.

In order to gauge the generalizability of conclusions drawn using the ANIMAL algorithm, consider the transformation class, data term, and regularization used. The transformation obtained is a mesh warping, a very flexible and commonly-used parameterization [BK89, KPH$^+$96, RSH$^+$99, KLF99, RCRM01, SD02, Dav96, CRM96], and thus not a limiting factor. The data term is based on image intensity assuming an affine relationship between the registered images, a very reasonable assumption for registering images of the same modality, as discussed in Section 3.2.2. Finally, the regularization method is a very simple elastic method. As discussed in Section 1.2, there is no theory to suggest an optimal regularization method, so the empirical performance remains the best guide. While localization results as measured in this thesis are not currently available for other registration methods, comparing the performance of two registration algorithms can be done qualitatively by comparing the mean intensity images. The sharpness of Figure 4.11 compares well with similar images obtained using other methods [SD03, RFS01], suggesting that the tuned ANIMAL algorithm performs as well as any in the field. Thus there is every reason to believe that the other elastic methods will produce results similar to those described in this thesis.

10.1.2 2D Normalization

Chapters 5 and 6 summarize and synthesize the literature on cortical surface registration. The 2D problem shares many characteristics of the 3D problem: spatial normalization in 2D is set up as registration with a template, and the registration algorithms are described in a framework of optimization. Thus, these chapters mainly highlight the features unique to the 2D problem.

One difference is that there are two additional steps required, illustrated in Figure 5.1. First, the cortical surface mesh must be identified from the MR image; here there are several competing definitions of cortical surface mesh from which to choose

(Section 5.1). Second, the mapping from the surface situated in 3D to an auxiliary 2-manifold must be defined and computed (Section 5.2). From the point of view of surface registration, the auxiliary space is used for computational convenience. The main choice for cortical surface and auxiliary space is to choose between a disc or sphere topology. In the latter case, one can choose to use one sphere per cortical hemisphere or a single sphere for both hemispheres. For this work the sphere topology is chosen, so as to avoid having to identify the hemisphere boundaries in the the image [DECS99]. The cortical surfaces are extracted using ASP [Mac98], which results in a single sphere comprising both cortical hemispheres. Thus, a single unit sphere is used as the auxiliary space for experiments in this thesis. The spatial normalization mapping is from the sphere to itself, parameterized as a sphere triangulation warp (Definition 5.4.8).

In contrast to 3D image registration where the matching problem is based on the image data itself, the surface registration problem, does not have such a real-valued function given as part of the problem specification, so one must be created (Section 6.2.2). In order to match the pattern of sulcal folding, a function is defined that reflects the geometry of the folding pattern. In Section 7.1.1, two possible functions are tested: the mean surface curvature, and the crown distance transform. The mean surface curvature requires smoothing and, for the smoothing scheme used (Section 6.7.1), the crown distance transform produced better results as shown in Figure 7.5. Though the smoothing scheme could use more investigation, the experiments in this thesis use the crown distance transform feature data as it currently produces better results.

The data terms, model terms, and registration algorithms suitable for surface registration are adaptations of existing methods for image registration. The main challenge is to accommodate the spherical topology. The 2D experiments in this thesis use the sphere to sphere warping algorithm proposed in Section 6.7, which has the following characteristics:

- the transformation is parameterized as a triangulation warp (Definition 5.4.8),
- the matching is driven by the crown distance transform feature (Section 6.2.2),
- the data term is a regional version of cross-correlation (Section 3.2.2),
- the model term is implemented as an elastic smoothing of the transformation in a two-step algorithm (Section 3.4.2), and

- four levels of coarse-to-fine hierarchy are used, refining the mesh by quadrisection (Section 6.5) between levels.

In addition to the design elements listed above, there are several numerical parameters used by the algorithm, as discussed in Section 6.7. Using the total entropy as a performance measure in the context of spatial normalization, the optimal values for the algorithm parameters are shown (Section 7.1.1) to be: penalty ratio $a = 0.05$, smoothing weight $w = 1$, search radius $r_s = 0.5$, and neighbourhood radius $r_n = 2.8$. The anatomical variability, as measured by the total entropy, is reduced after surface normalization and the entropy maps of Figure 7.14 show that the reduction occurs all across the sphere. The sharper images of average feature data shown in Figure 7.15 is further qualitative evidence of the global feature alignment.

As discussed above for the ANIMAL algorithm for 3D, general conclusions may be drawn from the experiments with the surface registration proposed in this thesis. The transformation is parameterized using a triangulation warp that can be made as fine as needed, and so captures a very wide class of transformations. The data matching is based on the folding pattern, which is the common method for matching surface anatomy [VEDD+01, FSTD99, VD99]. And finally, the regularization method, while simple, is demonstrated in Chapter 7 to be effective in producing good matches of the gyral and sulcal anatomy. Thus there is every reason to believe that the other elastic methods for matching surface anatomy will produce results similar to those described in this thesis.

10.1.3 Injectivity

As discussed in Chapter 1, due to the smoothing used the data analysis done after spatial normalization is effectively comparing small regions about $T_i(x)$ at each point x of the standard space rather than a point-to-point comparison. Thus, a small amount of noninjectivity in the spatial mappings can be tolerated. Using the optimal parameters, a set of 39 individuals is spatially normalized to check whether the generated transformations are commonly non-injective. For 3D, the results in Figure 4.8 show that while there are pockets of non-injectivity, overall the Jacobian determinant is negative at only 0.2% of the control mesh vertices.

In 2D, the situation is similar in that a triangle mesh warping is not necessarily injective and the algorithm of Section 6.7 does not enforce injectivity. A set of 151 surfaces are spatially normalized using the optimal parameters and the transformation is found to be injective except for 1% of the target sphere area as shown in

Figure 7.11.

For both 3D and 2D, the small amount of noninjectivity found is tolerated for this work.

10.1.4 Variability Comparison

Two competing spatial normalization methods, each having been tuned, can be compared by using each to spatially normalize a population and examining the residual variability in the two cases.

If homologous structures are being aligned by the spatial normalization, then gray matter is expected to align more often with gray matter, and similarly white matter with white matter. This will generate more consistent labellings at each voxel, so the anatomical variability, as measured by total entropy, should decrease. However, the converse is not true: a decrease in entropy at a voxel because it is consistently assigned the label gray matter does not mean that the gray matter comes from the same sulcus in each individual. The right column of Figure 4.14 demonstrates that this can happen. Thus it is of interest to quantify the variability of structures smaller than the totality of white matter or of gray matter. For the experiments of this thesis, a manual segmentation of frontal sulci, shown in Figure 4.13, is used. Since the gray matter of different sulci have different labels, the total entropy of the sulcus labelling *is* sensitive to mismatching of sulci, in contrast to the situation described for tissue class labels.

However, as discussed at the end of Section 4.1.2 on page 83, the entropy of the sulcus labelling does not capture the spatial spread of the distribution of normalized sulci. The total entropy can decrease if the input sulci are well matched to, say, two folds on the template, regardless of how widely separated in space are the two folds, as illustrated in Figure 4.15. To capture the spatial aspect of variability, a dispersion measure akin to median absolute deviation of univariate statistics is proposed in Section 4.2. Dispersion is defined as the median distance from a central structure to each of the input structures. Each structure is a closed point set and the distance between point sets is measured using a trimmed Hausdorff distance (Equation 4.7) to reduce sensitivity to outliers. This distance measure is chosen because it is applicable to any closed point set, e.g. the gray matter object, and the sulcus structures given as a set of voxels that are used in this thesis. It does not require a point-to-point matching of the structures nor does it require any manual intervention (apart from perhaps initially delimiting the structures).

Three methods for obtaining a central object about which dispersion can be measured are discussed in Section 4.2. One option is to measure dispersion about the template structure as in Section 4.2.2. This could be useful to gauge suitability of that template for the purpose of automated labelling, since labelling requires that the template structure be accurately matched to the corresponding structure in the population. The second option is to use the Vorob'ev mean structure, which is possibly useful when the input structures contain a lot of overlap, as is the case for larger structures such as gray matter. The Vorob'ev mean of the gray matter is used successfully in Chapter 8 to select a more representative individual from the population of brains, for use as a template. The third option is the distance mean, which is useful for small structures that do not necessarily overlap greatly such as the frontal sulcus structures.

A measurement of the localization of a given structure after applying spatial normalization should capture the notion that the normalized structures should be neither too disperse, nor cover too large a volume, as illustrated in Figure 4.25. From the notion of confidence intervals on the real line, which locate the region in which a given fraction of the population is concentrated, an analogous measure for extended objects is proposed. The localization measure is the size of the distance mean after dilating by the dispersion measure. This measure takes into account both the size of the mean as well as the dispersion about it. All of the definitions can be taken over to the sphere and used in 2D as well (Chapter 7). When comparing two competing spatial normalization methods, the smaller localization value is preferred.

The localization measure is used in Chapter 8 to show that a good template for spatial normalization can be obtained by applying the warp average (Section 8.1.4) and the intensity average (Section 8.1.1) strategies, starting with a well-chosen individual image (Section 8.1.2).

The localization measure is used in Chapter 9 on a data set of 40 subjects, to show that the 2D normalization produces a better localization than 3D normalization. In contrast to the previous work, which compared high-dimensional 2D warping with a low-dimensional warping in 3D (a 9-parameter affine transformation) [DECS99, FSTD99], the 3D normalization is carried out using a performance-tuned, high-dimensional warping algorithm (ANIMAL).

10.2 Conclusions

The impetus of this work was to investigate the question of whether it is better to perform spatial normalization in 2D or in 3D, for the purposes of brain mapping.

There are clearly some situations where 3D normalization is required, e.g. whenever non-cortical structures are of interest. It is equally clear that the analysis of functional brain mapping data may suffer when performed in 3D since the smoothing and averaging of data can easily mix together signals from opposite banks of a sulcus when they lie close together in 3D, whereas the locations may be quite distant as measured along the surface. This is the main motivation behind identifying the cortical surface and doing the data analysis on it. While there remains some challenges in identifying the cortical surface and mapping functional data to it, especially data from fast imaging methods which do not have the resolution of the anatomical images used in this thesis, we may imagine that these problems will be solved. It is then of considerable interest to know whether homologous anatomy of the cortex can be better aligned when modelled as surfaces, or whether a 3D registration algorithm can match the cortex sufficiently that data no longer has to be averaged over such a large region so as to mix together signals from the opposite banks of a sulcus.

This question can be answered empirically by performing a spatial normalization with a 3D algorithm and with a 2D algorithm and comparing the results. It is clear that the 3D algorithm must be a high-dimensional warping, rather than simply affine as the previous comparisons have done [DECS99, FSTD99]. In order to ensure that the results are generalizable, each algorithm should be representative of a whole class of algorithms. Moreover, it became clear that the template used has a large impact on the quality of the resulting spatial normalization, so a good method for choosing a template was also required.

3D Spatial Normalization

For 3D experiments, the ANIMAL algorithm was readily available in the lab. It was developed to match sub-cortical structures and had not been completely successful in matching sulci, even incorporating explicit sulcal constraints [CLGE98]. Therefore, I employed the segmentation variability (total entropy) described in Chapter 4 in order to explore the performance of ANIMAL over a range of parameters. This proved to be a very effective tool for probing any design aspect of a registration algorithm. This procedure can be applied equally well to any registration algorithm.

After this tuning, the performance of ANIMAL is believed to be comparable to other elastic registration methods, as discussed above. Localization results as measured in this thesis are currently not available for other registration methods, so comparing the performance of two registration algorithms can only be done qualitatively. Comparing the intensity average in Figure 4.11 with similar images obtained using other methods [SD03, RFS01] indicates that the tuned ANIMAL produces a similarly-sharp result. Thus, conclusions drawn from experiments using ANIMAL are expected to apply to other elastic 3D registration methods.

To compare ANIMAL with fluid registration methods, note that the transformation for the latter is generally parameterized in the same manner, i.e. using a mesh warping [CRM96, BNG96]. While the "re-gridding" technique used by Christensen means that the final transformation is actually a concatenation of several mesh warpings with variable control vertex spacings, this technique could equally well be used in an elastic transformation if required. The data term used in fluid methods is generally based on image intensity using the sum-of-squared difference criterion, Φ_{SSD} (Equation 3.8 on page 32), which is a less general assumption than the affine relationship implied by Φ_{CC} used in ANIMAL. The data matching behaviour of ANIMAL should therefore be no worse than that of an algorithm using Φ_{SSD}. The main difference between the methods, therefore, is the regularization. The claimed advantage of fluid registration over elastic is that the former allows injective matching with larger deformations, under certain conditions [Ami94, DGM98, JM00]. It is true that the transformations generated using ANIMAL are not injective but, as discussed above, the amount of non-injectivity is tolerable in this situation. As for large deformations, the results of this thesis suggest that the transformations produced using elastic model terms are sufficient to capture cortical variability of a normal population. For example, Figure 8.8 indicates that ANIMAL is able to match successfully the central sulcus, as long as a good template is chosen, e.g. template T1 in the middle column. Note, however, that this conclusion is based on a population of normal anatomies. It is likely that the large deformation properties make a fluid registration more appropriate than an elastic registration in a situation for which there truly is a large difference in the images, e.g. one has a tumour or other large structural change and the other image does not.

Template Selection

While the entropy measure is a useful global indication of alignment, it does not capture spatial variability of individual structures, as discussed at the end of Section 4.1.2 on page 83. Thus, I developed a measure of dispersion using distance between point sets and the distance mean of a random closed set. It became clear that the template that was selected (arbitrarily) for the experiments of Chapter 4 was not able to be matched well to the population. For example the left central sulcus of half the input brains was matched to the wrong fold of the template. In order to find a more representative individual to use as a template, I used the distance measure from the mean gray surface. This proved to be a much more effective template, and was further improved by using warp averaging and intensity averaging, as shown in Chapter 8.

2D Spatial Normalization

As no corresponding 2D registration algorithm was available, I wrote one as discussed in Chapter 6, modelled on the successful pattern of ANIMAL. The numerical parameters were obtained by a 2D segmentation variability measure, as described in Section 7.1, ultimately producing a surface registration that is very effective at aligning anatomy, as evidenced by the reduction in localization measures shown in Figure 7.25, and the sharper images of average feature data shown in Figure 7.15.

As argued above, the design of this algorithm is representative of the class of elastic warping based on matching sulcal and gyral features. Thus the results of experiments using this algorithm are expected to be representative.

3D versus 2D

With a representative algorithm for 3D image registration and for 2D surface registration, as well as a demonstrably good-quality template, we can now approach the motivating question with some confidence that the finding will hold generally. To answer this question, I obtained the frontal sulcus labelling of Figure 4.13 and then used the localization measure described in Section 7.2 to gauge the amount of variability remaining on the surface after spatial normalization. As shown in Chapter 9, the localization was smaller after 2D normalization, for all 8 structures examined. The main reason for this is that the 2D normalization generally managed to map the individual sulci into the template sulcus, whereas the 3D normalization mapped many individuals to neighbouring sulci, as illustrated in Figure 9.5. Thus the 2D normalization should enhance the analysis of surface-based data better than a 3D normalization.

10.3 Future Work

Using the tuning methodology of Section 4.1.1, there are many aspects of both ANIMAL (Section 3.7) and the surface warping algorithm (Section 6.7) that remain to be investigated and improved upon. Testing for convergence in terms of iterate change of the inner loop may yield improved execution time by eliminating useless iterations. The link between the amount of data smoothing and the control grid vertex spacing could be investigated. Non-isotropic smoothing [PM90, KS96], which preserves edge location, might be a better choice. The surface warping algorithm should have the data term retain a constant sampling density, independently of the chosen neighbourhood radius value.

Further work on localization should be done, such as repeating the 3D/2D comparison of Chapter 9 using the warp average and intensity average template T4, the best template tested in Chapter 8.

The improvements to ANIMAL should enhance its ability to label gyral structures, given a suitable template image. Finally, the new surface matching algorithm is certain to be useful in spatially normalizing functional or structural data on the surface.

Appendix A

Glossary of Terms

1-ring neighbourhood Let (V, E) be a graph, and $v \in V$. The 1-ring neighbourhood of v is the set of neighbours of v, i.e. the set $\{u : (u, v) \in E\}$.

26 neighbours On a 3D mesh, a 26-neighbours of v are the vertices of the $3 \times 3 \times 3$ block centered on v, other than v itself.

ANIMAL Automated Nonlinear Image Matching and Anatomical Labeling, registration software by D. L. Collins [CHPE95], summarized in Section 3.7.

antipodal Let $p \in \mathbb{S}^2$. The point $-p$ is antipodal to p.

ASP Anatomic Segmentation using Proximities, surface extraction software by D. MacDonald [MKAE00].

axial Imaging plane. For an upright person, the axial plane is parallel to the floor.

bijective A mapping that is both injective and surjective.

bounded set A subset A of a metric space is bounded if it is contained in an open ball of finite radius.

brain stem A bundle of nerve fibres attached to the underside of the cortex and descending into the spinal column.

closed set A subset A of a metric space is closed if for each convergent sequences in A, the limit is also in A.

compact set A set $A \subset \mathbb{R}^3$ is compact if, and only if, it is bounded and closed. In a general metric space X, a set A is compact if, and only if, each infinite sequence of points $\{p_n\}$ in A contains an infinite subsequence, say $\{q_n\}$ which converges to a point in A.

complete metric space A metric space (X, d) for which each Cauchy sequence converges to a point in X. A sequence is Cauchy if for every $\epsilon > 0$, $\exists N$ such that $p, q > N \implies d(x_p, x_q) < \epsilon$.

convex A point set is said to be convex if, for any pair of points x and y in the set, the line segment xy also lies in the set. A real-valued function of a vector space is convex if its domain is a convex set and the function satisfies $f(tx+(1-t)y) \leq tf(x) + (1-t)f(y)$ for all x, y in the space and all $t \in [0, 1]$.

convex hull Given a discrete set of points $P = \{p_i \in \mathbb{R}^d : i = 1, \ldots, N\}$, the convex hull of P is the smallest convex set containing P.

coronal Imaging plane. For an upright person, the coronal plane is perpendicular to the floor and contains the left-right direction.

corpus callosum Large bundle of nerve fibres connecting the two cerebral hemispheres.

cortex The superficial coat of gray matter of the brain.

crown See gyral crown.

cryosection Refers to techniques in which the specimen is frozen and sliced into sections.

CSF Cerebral Spinal Fluid.

decimate Reduce the complexity (the number of vertices, edges, and facets) of a polyhedral surface mesh, generally with the aim that the new surface remains an approximation of the old surface.

distance transform For a point set A, the distance transform $d_A(x)$ is the length of the shortest path from x to a point in A.

eigenfunction Let $\mathcal{L}$ be a linear differential operator. A function f is an eigenfunction of $\mathcal{L}$ if it satisfies $\mathcal{L}f = \lambda f$ for some $\lambda \in \mathbb{R}$.

embedded graph A planar graph with a specified circular ordering of the neighbours at each vertex. The embedding defines the graph faces.

Euclidean space The metric space $(\mathbb{R}^n, d)$, where $d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$.

filtering Transformation of a signal, whether scalar (e.g. image intensity I) or vector (e.g. transformation function T), by convolution with some kernel. Sometimes described as smoothing.

function Any mapping between sets. The domain set could be a set of functions; some authors refer to this as a functional, but that distinction is not used in this thesis.

fundus See sulcal fundus.

FWHM Full Width at Half Maximum, an expression of the extent of a function $f(x)$ given by the distance between two extreme values of x at which the function value is half its maximum value.

genus The genus of a surface is equal to the maximum number of nonintersecting closed curves that can be drawn on the surface without separating it. A sphere has genus zero, the torus has genus one.

geodesic A locally shortest curve. A geodesic in $\mathbb{R}^n$ is a straight line. On a 2-sphere, a geodesic is a great circle. See shortest path.

graph A pair (V, E) where V is a finite set of vertices, and $E = \{(u,v) : u, v \in V\}$ is a set of edges. All graphs in this thesis are undirected, i.e. edge (u,v) is considered the same as edge (v,u).

gyral crown The ridge running along the top of a gyrus.

gyrus Fold of brain tissue; see Figure 1.2.

halfedge data structure Data structure for storing an orientable 2-manifold polyhedral surface. If vertices u and v are neighbours, the data structure stores two directed halfedges, (u,v) and (v,u) [Ket99].

Hausdorff space A space in which any two points have disjoint neighbourhoods.

Hilbert space A vector space with an inner product $< x, y >$ for which the metric defined by $d(x,y) = ||x - y||$, where $||x|| = \sqrt{< x, x >}$, turns it into a complete metric space.

homeomorphism A continuous, injective mapping with a continuous inverse.

homogeneous coordinates A coordinate system for which $(x_1, x_2, \ldots)$ and $(\lambda x_1, \lambda x_2, \ldots)$ for any $\lambda \neq 0$ represent the same point. Homogeneous coordinates for a point in $\mathbb{R}^n$ are given by a set of $n+1$ numbers, $(x_1, \ldots, x_n, w)$ that correspond to the Cartesian coordinates $(x_1/w, \ldots, x_n/w)$.

homologue A biologically corresponding component.

homology A biological correspondence; similar in position and structure but not necessarily in function.

ICBM International Consortium for Brain Mapping, a collaboration of several neuroimaging labs [MTE+95].

infimum The infimum of a nonempty set $A \subset \mathbb{R}$ is the largest number that is less than or equal to all $a \in A$. If A is not bounded from below, then the infimum is defined to be $-\infty$.

injective A mapping $f : A \to B$ is injective if $f(x) = f(y) \implies x = y$. Synonym for "one to one" mapping.

interpolation The computation of values between ones that are known using values at surrounding points.

magnetic resonance imaging Imaging technique based on the interaction of nuclear magnetic fields of the sample with an externally-imposed field.

Mahalanobis distance Let X be a random variable drawn from an n-dimensional normal density with mean $\mu = \mathsf{E}X$ and covariance matrix $\Sigma = \mathsf{E}(X-\mu)(X-\mu)^T$. The Mahalanobis distance from x to μ is r where $r^2 = (x-\mu)^T\Sigma^{-1}(x-\mu)$.

manifold An n-manifold is a connected Hausdorff space each point of which possesses a neighbourhood that is homeomorphic to an open subset of Euclidean n-space, $\mathbb{R}^n$. A polyhedral surface (e.g. representing the cortex) and the sphere $\mathbb{S}^2$ are examples of 2-manifolds.

mesh A graph that has a point in $\mathbb{R}^3$ associated to each vertex. In this thesis, two kinds of mesh are used. The first, used for 3D warping (see Definition 2.4.1), has a vertex set isomorphic to the set of $N_1N_2N_3$ triples $\{(i,j,k) : i \in [1,2,\ldots,N_1]; j \in [1,2,\ldots,N_2]; k \in [1,2,\ldots,N_3]\}$ with an edge set $\{((i,j,k),(u,v,w)) : |i-u| + |j-v| + |k-w| = 1\}$. This kind of mesh is sometimes referred to as a grid. The second kind of mesh is a an embedded graph where each face is a triangle, sometimes referred to as a triangulated mesh. This kind of mesh is used to represent the surface of the cortex, both in its native configuration and after being projected to the unit sphere. A triangulated mesh is also used for sphere triangulation warping (Definition 5.4.8).

metric Let X be a set. A mapping $d : X \times X \to \mathbb{R}$ is a metric if it satisfies the following conditions for all $x, y, z \in X$: (i) $d(x,y) \geq 0$; (ii) $d(x,y) = 0 \Leftrightarrow x = y$; (iii) $d(x,y) = d(y,x)$; (iv) $d(x,z) \leq d(x,y) + d(y,z)$. The standard Euclidean distance function is an example of a metric for points in $\mathbb{R}^n$.

metric space The pair (X, d) where X is a set and d is a metric on X.

modality An image generation technology or technique; examples are: PET, X-ray, and T_1-weighted MR.

MR Magnetic Resonance.

MRI Magnetic Resonance Imaging, or Magnetic Resonance Image.

Newton methods Class of optimization methods that determine the step direction and length by minimizing a quadratic approximation to the objective function, i.e. $\Phi(x + p) \approx \Phi(x) + p^T \nabla\Phi(x) + \frac{1}{2}p^T Hp$, at the current iterate, where the Hessian, H, is an $n \times n$ matrix of second derivatives of Φ evaluated at x. This approximation is minimized by setting $p = -H^{-1}\nabla\Phi$, an expensive operation when n is large. See quasi-Newton methods.

open ball Given a real $r > 0$, and a point p in metric space (X, d), the open ball $B(p, r)$ is defined as $\{x \in X : d(p, x) < r\}$.

open set A subset A of a metric space is open if for each $a \in A$, there is an open ball centred at a contained in A.

partial volume effect An artifact that appears in MR images when a voxel encompasses two or more types of tissue. The signal intensity recorded for such a voxel is a mixture of the signal intensities for the component tissues.

planar A graph is planar if it can be drawn in the plane without graph edges crossing.

quasi-Newton methods Class of optimization methods that derive the search direction as $p = -B^{-1}\nabla\Phi$, where the matrix B is an approximation to the Hessian (see Newton methods), but easier to update for the next iteration than is the true Hessian.

sagittal Imaging plane. For an upright person, the sagittal plane is perpendicular to the floor and contains the front-back direction.

segmentation Division of a digital image into regions. Each voxel is assigned to a certain class.

shading artifact Refers to a smooth variation in signal intensity across an image; also called intensity nonuniformity.

shortest path A path between two points, x, y such that no other path between x and y is shorter. A shortest path is a geodesic. The converse is not necessarily true.

simple polygon A polygon in the plane is given as an ordered list of vertex positions, $(v_0, v_1, \ldots, v_n \equiv v_0)$, with edges formed by the line segments $v_{i-1}v_i$, $i = 1, \ldots, v_n$. The polygon is simple if edges do not intersect, except at the shared vertex v_i between edges $v_{i-1}v_i$ and v_iv_{i+1}.

sinc Abbreviation of sine cardinal, the function is defined as $\text{sinc}(x) = \sin(\pi x)/\pi x$ for $x \neq 0$, and $\text{sinc}(0) = 1$.

slice Intersection of a plane with a 3D image. Images are often displayed using three orthogonal slices through the volume.

smoothing See filtering.

steepest descent A line search optimization method that always chooses the objective function gradient, $-\nabla\Phi$, as the search direction.

sulcal fundus The "valley floor" of a sulcus.

sulcus Furrow or groove of the brain; see Figure 1.2.

supremum The supremum of a nonempty set $A \subset \mathbb{R}$ is the smallest number that is greater than or equal to all $a \in A$. If A is not bounded from above, the supremum is defined to be ∞.

surjective A mapping $f : A \to B$ is surjective if, for each $b \in B$, there exists at least one $a \in A$ such that $f(a) = b$. Sometimes denoted an "on to" mapping.

tissue classification A segmentation into tissue classes, e.g. gray matter, white matter, CSF, or background.

triangulated graph An embedded graph for which the faces all have three edges.

ventricles Cavities in the brain filled with cerebral spinal fluid.

volume Short for volumetric image.

volumetric image 3D image.

Appendix B

On the Reliability of Triangle Intersection in 3D

Over the course of this thesis work the question arose as to whether two brain surfaces intersect, where each is modelled as a triangulated polyhedron. For example, I had occasion to check whether or not the two surfaces generated by ASP were truly concentric. Surface intersection can be checked using brute force in $O(n^2)$ time by testing each pair of triangles for intersection. However, by enclosing each triangle in an axis aligned bounding box, an algorithm of Edelsbrunner and Overmars [EO85, ZE02] can be used to find the intersecting bounding boxes efficiently, in roughly $O(n \log^2 n + k)$ time for n boxes in total and k pairs of intersecting boxes. Then, only the k triangle pairs whose corresponding bounding boxes overlap need to be tested.

Testing whether a pair of triangles intersect, given their coordinates in floating point numbers, is a straightforward computation, but it turned out to be surprisingly fragile code. This appendix reports on the implementation of a robust triangle intersection test. This is joint work with Sue Whitesides that has previously appeared in a conference publication [RW03]. My contributions to this work include all the coding and experiments, and the majority of the paper text.

B.1 Abstract

Implementors of triangle/triangle intersection tests often opt to forego exact calculations for speed reasons. It is widely known that such code will fail for certain inputs, but it is not evident from the literature that published intersection tests implemented using floating-point arithmetic are not stable. We show how such a test can fail on

a triangle pair that is widely separated in space. We find that an exact intersection test can be implemented with a modest speed penalty.

B.2 Introduction

Surfaces are often modeled using polyhedra composed of many triangles. When two such surfaces are checked for intersection, a preprocessing step using bounding volumes (arranged hierarchically [GLM96] or not [ZE02]) is used to quickly eliminate many triangle pairs from further consideration. For the pairs that remain, a fast method of checking for intersection is required. In geometric computing, speed comes at the cost of reliability.

Reliability of computational geometry algorithms has been studied for many years, see e.g. [Yap97] for an overview. Algorithms are classified as *exact* if the result is always correct for the input given, *robust* if the result is always correct for some perturbation of the input, and *non-robust* otherwise. A robust algorithm for which the perturbation is bounded is called *stable*. Most work on robustness has examined classical computational geometry algorithms such as computing the Delaunay triangulation or the arrangement of a set of lines, for which the combinatorial output must satisfy nontrivial consistency properties. Triangle intersection, having a single bit of output, is trivially robust. The interesting question is whether an algorithm is stable or exact.

Exact computation can be achieved by using extended precision arithmetic to compute each quantity used for a branching test in the algorithm [YD95, LY01]. Extended precision arithmetic is generally implemented in software, and thus comes at a cost in execution time. In some contexts, e.g. graphics or virtual reality systems, speed is paramount and an inexact, but stable, triangle intersection test is perfectly acceptable. We show, however, that published triangle intersection codes of Held [Hel97] and of Möller [Möl97] are not stable. Such code can be made stable using a general arbitrary precision arithmetic package such as LEDA's reals [BMS96] or the CORE library [KLPY99], but the resulting speed penalty is prohibitive for many applications. Instead, we use the orientation tests of Shewchuk [She97] to implement an exact triangle intersection test, with a modest penalty in speed.

B.3 Inexact Intersections

It is instructive to first examine testing for line segment intersection in the plane.

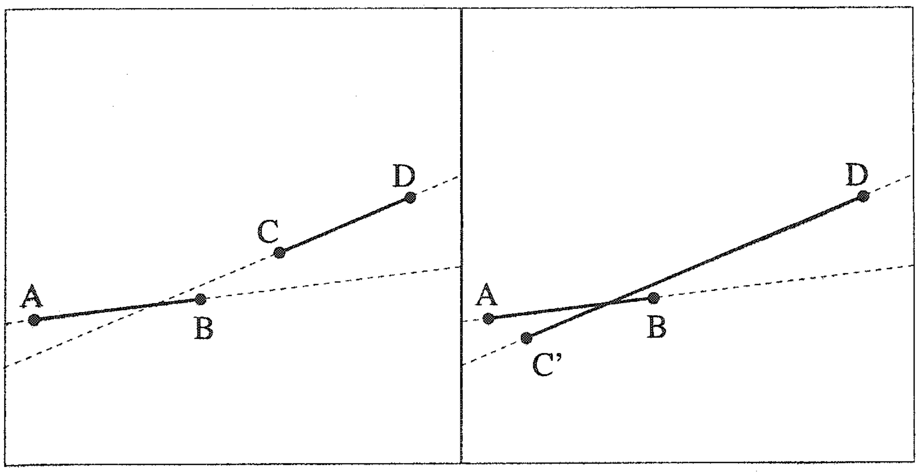


Figure B.1: Problematic case for segment intersection. Point C lies close enough to the line through AB that the floating point computation of the orientation predicate fails, whereas the orientation for point D is correctly computed. The test result is as if the input on the right were given instead. Note that the perturbation from C to C' may be arbitrarily large.

B.3.1 Segment Intersection

It is well known (e.g. O'Rourke [O'R98]) that this problem can be phrased in terms of the 2D orientation predicate, which is given by the sign of a degree-2 polynomial in the coordinate values of three points. The predicate value indicates on which side of the directed line through two points lies the third point. An implementation of this predicate in floating point arithmetic can fail when the value is very small, which corresponds to the three points being nearly collinear. Two non-intersecting segments are shown in the left panel of Figure B.1. If the orientation test for point C with respect to line AB fails, the segment intersection test incorrectly reports that the segments do intersect. In effect, the intersection algorithm behaves as if it were given the perturbed input pictured on the right panel of Figure B.1.

This is a particularly egregious error, since the segments may be arbitrarily far apart. Note also that this problem is exacerbated by the common practice of using an "epsilon" for numeric comparisons, i.e. a floating point number x is deemed to be equal to floating point value y when $|x - y| < \epsilon$. The segments in this example will then be deemed intersecting if the orientation value $< \epsilon$ even when the approximate computation correctly returns a positive number.

Several algorithms for computing the sign of the orientation predicate exactly using existing floating point hardware have been implemented and used for segment intersection testing [OTU87, She97, GR00].

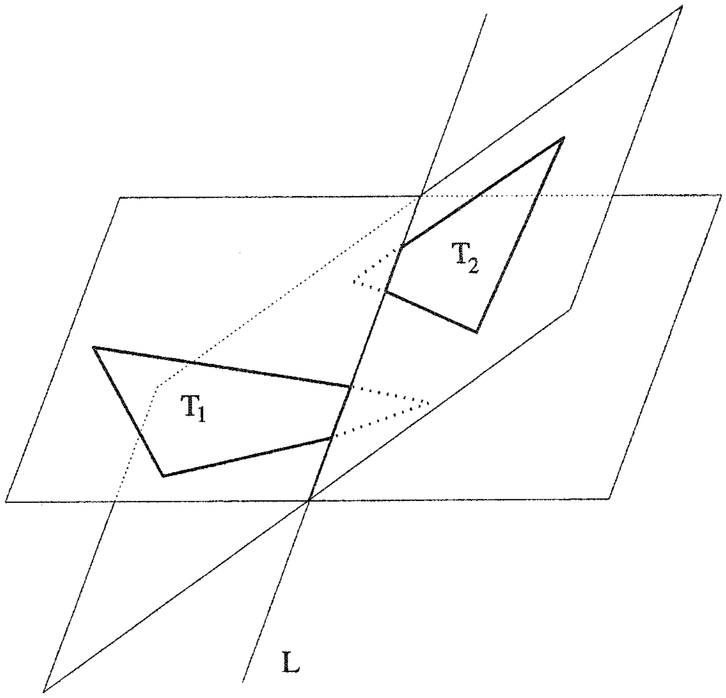


Figure B.2: Triangle intersection. Line L is the intersection between the planes supporting each triangle.

B.3.2 Triangle Intersection

Testing whether two 3D triangles intersect can be reduced to computing the 3D orientation predicate, given as the sign of a degree-3 polynomial in the coordinate values of four points [ABM97, O'R98]. The predicate value indicates on which side of the oriented plane determined by three points lies the fourth point.

Let the input triangles be T_1 and T_2, each specified using coordinates of its three vertices. Let the planes supporting the triangles be denoted by π_1 and π_2, respectively. Testing whether triangle T_2 intersects π_1 is a matter of testing each vertex of T_2 against π_1 using the orientation predicate. If all vertices lie strictly to one side of the plane, T_2 does not intersect π_1; otherwise they do intersect.

The code of Möller [Möl97] operates as follows. Test whether triangle T_1 intersects π_2, then test whether triangle T_2 intersects π_1. If both are true, construct the line $L = \pi_1 \cap \pi_2$, compute intersection of L with each triangle, then test whether the two segments (on L) intersect. This code, being implemented with floating point, suffers from the analogous flaw as the segment-segment intersection previously described. Two nearly-coplanar triangles may be arbitrarily far apart, yet declared intersecting if the orientation test fails for one vertex. We have observed this situation in practice

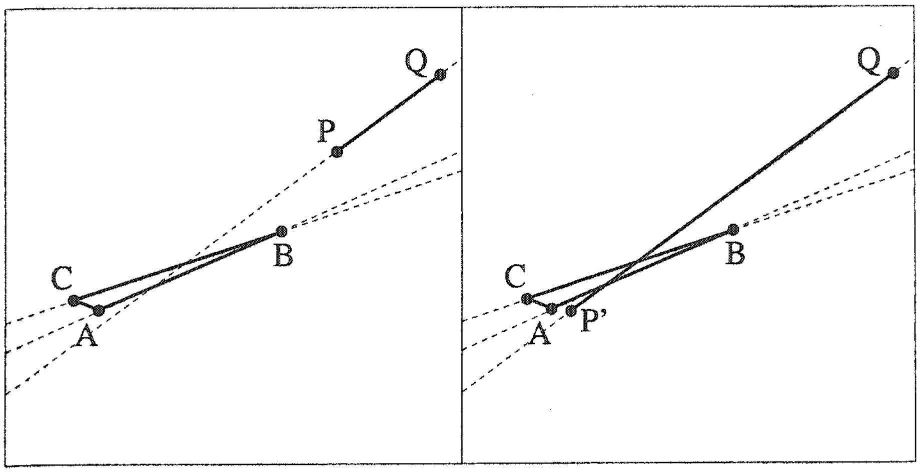


Figure B.3: Potential problem for ERIT method.

for triangles generated by the tessellation of an ellipsoid, and for triangles generated with random coordinate values.

The ERIT code [Hel97] operates by first testing whether π_1 intersects T_2. If so, then the segment $S_2 = T_2 \cap \pi_1$ is constructed and tested for intersection with triangle T_1. The segment/triangle intersection is computed in 2D after projection to a suitable plane. Consider an input (pictured on the left of Figure B.3) in which (after projection) triangle T_1 (ABC in figure) is skinny and segment S_2 (PQ in figure) is nearly collinear with lines supporting edges AB and BC of T_1. Suppose that the location of Q with respect to the edge lines is correctly decided, but the approximate 2D orientation tests place P below lines AB and BC. The intersection test is computing as if it were given the perturbed situation pictured in the right panel of Figure B.3 and incorrectly declares that S_2 intersects triangle T_1, hence the two triangles intersect.

In short, both methods can fail on a widely-separated triangle pair. Using a bounding-box preprocessing step will alleviate but not eliminate this problem.

B.4 Exact Intersection

Triangle intersection can be made exact using an exact orientation test. Using extended precision arithmetic unconditionally to compute the exact value of the orientation test polynomial is possible, but slow. Practical implementations of the orientation test use a floating point filter to handle most cases, reverting to the full computation only when the value is close to zero [BBP98, BFS98, She97]. The idea is that a large fraction of inputs can be decided using floating point arithmetic. The easy cases are filtered out quickly, leaving only a few hard cases for exact computation. For example, in a mesh generation application, Aftosmis et al. [ABM97] report on a helicopter data set for which only 68 out of 1.37 million orientation calculations failed to be resolved by the floating point filter.

One route to an exact computation, advocated by Yap and Dubé [YD95], is to take an existing intersection test and replace the fixed precision calculations with (filtered) arbitrary precision arithmetic so that correct branching decisions are guaranteed. This is made very easy to do with the CORE arithmetic library [COR03] or LEDA [MN99]. Both packages enable exact computation of arbitrary arithmetic expressions, though the generality comes at a high cost in running time. Such packages store an expression graph, used to perform the exact computations in cases where the floating-point filter fails, so they tend to consume a lot of time and memory building the graph [KLN91] even when not needed. We replaced the arithmetic in Möller's code using CORE, for example, and found that it was about 2000 times slower.

Not willing to pay such a high price, we build our intersection test around the highly optimized implementation of the 3D orientation test by Shewchuk [She97]. This code implements a single primitive test, so it can be made much faster than the equivalent computation using CORE or LEDA, e.g. by unrolling loops by hand and allocating memory in local variables.

Since the 3D orientation predicate uses the sign of a polynomial, we could have used instead the exact sign of sum algorithm (ESSA) of Ratschek and Rokne (see [GR00]). ESSA requires that each term of the sum be exactly represented in a machine floating-point number. The 2D orientation predicate is degree two in the coordinate values, so Gavrilova and Rokne were able to use single-precision coordinate values and a double-precision implementation of ESSA. In 3D, the orientation test is degree three in the coordinates, so we cannot represent each term of the sum in a double-precision floating-point word. We would have to limit the inputs to 17 bits of precision when using IEEE double-precision numbers (53-bit significand) for ESSA.

Shewchuk's 3D orientation code works with inputs of either single- or double-precision floating-point numbers. We use his code to implement an exact triangle intersection test as follows. Our code first checks whether triangle T_1 intersects plane π_2. If not, then we have established that the triangles do not intersect and we're done. Otherwise, exactly two edges of T_1 intersect π_2.[1] Each of these edges is

[1] In the general case. Our code also handles the degenerate cases where one vertex or one edge

checked for intersection with T_2, again using applications of the 3D orientation test. If intersection is not yet established, the same procedure is applied with the roles of T_1 and T_2 interchanged.

B.4.1 Timing Experiments

Pairwise Triangle Intersection

Our intersection code is expected to be slower than inexact tests. To get a handle on the performance penalty incurred, we timed triangle intersection tests using our code and using two published codes: TRI_TRI_INTERSECT() of Möller [Möl97], and the TRITRI3D() routine from ERIT 1.1 [Hel97]. Möller's code was updated in 2001 with a newer, slightly faster version that avoids division. Comparisons such as $a/b < c/d$ are re-written as $ad < bc$. This entails evaluating polynomials of larger degree in the coordinates, however. We found that the computation would occasionally overflow the exponent of single precision numbers, so we report times using the original code.

We used the codes as-is. ERIT computes using double precision arithmetic, while Möller's test uses single precision. This potentially gives the latter an advantage in speed, but the limited precision also renders it more fragile. We tested both single precision and double precision in our exact implementation. On some systems, the times were about the same, while others showed single precision to be up to 30% faster. However, the filter is expected to be more effective when using higher precision, so the times reported here use double precision arithmetic.

The speed of these codes is expected to vary across different hardware, compiler, and input data. We report on two platforms. The first is a 360MHz MIPS R12000-based IRIX 6.5 system with 256MB memory, compiled using the MIPSpro 7.30 compiler with flags `-O2 -n32 -mips4`. The second system is a 1GHz Pentium III machine with 1GB memory, running linux; the compiler is GNU GCC version 2.95 with optimization flags `-O2 -march=i686`. We use the CLOCK() system call to measure the time, which reports times in units of 10 milliseconds. We repeat the entire dataset a number of times so that the total execution time is at least 10 seconds, rendering the error due to clock granularity negligible.

The input data matters because Shewchuk's routines are adaptive, with four levels of filtering. We measure first the best case using coordinates randomly chosen in the interval $[0, 1000]$. All determinant signs were resolved at the first level of

of T_1 lies in plane π_2.

Table B.1: Performance of triangle intersection test on pseudo random data. Time is given in microseconds per test while the slowdown factor (SlowF) is the time divided by the time for method Möller.

	MIPS R12000		Pentium III	
Method	Time	SlowF	Time	SlowF
Möller	0.63	1.0	0.80	1.0
ERIT	0.84	1.3	1.2	1.5
Exact (lower)	2.1	3.3	2.6	3.3
Exact (upper)	30	47.6	34	42.5

filtering, giving the lower bound on the speed penalty. We obtain the upper bound by modifying Shewchuk's code so that it never exits early, thereby obtaining a range for the speed penalty.

The results are tabulated Table B.1 in terms of the raw time for a test (averaged over many repetitions of 10000 randomly-chosen triangle pairs) and also a ratio of the time divided by the time for the fastest method (Möller's in our test) giving a slowdown factor. We expect the vast majority of instances to be near the lower end of the range. This depends on the application, and data characteristics, of course. Aftosmis et al. [ABM97] also used Shewchuk's orientation code for their triangle intersection test, and found that only 0.005% of the tests failed the first filter. Even if 1% of the triangle pairs fail the filter, our results indicate a slowdown ratio of 3.7 at most. The exact computation is expected, therefore, to be on the order of 3-4 times slower in practice. It is worth keeping in mind, however, that this slowdown is only for the triangle intersection code. Real-world applications will only spend part of their running time in the triangle intersection routine, so the overall impact on the application is not expected to be as great.

Surface Intersection

An example real-world application is the surface intersection code discussed at the beginning of this appendix. Table B.2 shows the running time for intersecting surfaces comprised of various numbers of triangles. Each run tests for the intersection of the inner and outer surface as generated by ASP, the times shown in the table are an average over 152 such pairs. These timing values are obtained on a linux system with a 2.2 GHz AMD Athlon processor. The time penalty is large for surfaces composed of a small number triangles and decreases for more complex surfaces. This is

Table B.2: Performance of full surface intersection: the time taken to process the bounding boxes as well as the triangle pair tests. Time is shown as mean (standard deviation) values, averaged over intersection problems with 152 different brain surface pairs.

#Triangles	Time (sec.)		Time Penalty
per surface	Möller	Exact	
5k	0.080 (0.003)	0.11 (0.009)	38%
20k	0.54 (0.01)	0.62 (0.04)	15%
80k	3.7 (0.2)	3.8 (0.2)	3%
320k	27.9 (0.9)	28 (0.9)	0.4%

because in the larger problems, the $O(n \log^2 n)$ time for bounding box preprocessing dominates over the $O(k)$ time to process the intersecting pairs. For cortical surfaces, 80k or 320k triangles are typical sizes, so the time penalty for exact intersection is minimal.

B.5 Conclusions

Intersection tests based on approximate arithmetic can fail not only on triangle pairs that are in "grazing" contact, but also on pairs that are widely separated. The common practice of using "epsilon" for numeric comparison exacerbates the problem. Practitioners need to be aware of these issues when implementing primitive geometrical tests.

A straightforward exact implementation can have a relatively small speed penalty, as our experiments have demonstrated.

Bibliography

[AAF99] John Ashburner, Jesper L. R. Andersson, and Karl J. Friston. High-dimensional image registration using symmetric priors. *NeuroImage*, 9:619–628, 1999.

[AAF00] John Ashburner, Jesper L. R. Andersson, and Karl J. Friston. Image registration using a symmetric prior — in three dimensions. *Human Brain Mapping*, 9:212–225, 2000.

[ABM97] M.J. Aftosmis, M.J. Berger, and J.E. Melton. Robust and efficient cartesian mesh generation for component-based geometry. In *35th AIAA Aerospace Sciences Meeting*, Reno NV, USA, January 1997. AIAA97-0196.

[ADPL$^+$03] Yasser Ad-Dab'bagh, Hilleke E. Hulshoff Pol, Jason Lerch, Hugo G. Schnack, Neeltje E. M. van Haren, Eric Fombonne, Alan C. Evans, and Rene S. Kahn. Case-control comparison of cortical thickness maps in schizophrenia: An anatomical mri study. In *9th International Conference on Functional Mapping of the Human Brain*, New York, June 2003.

[AF97] John Ashburner and Karl Friston. The role of registration and spatial normalisation in detecting activations in functional imaging. *Clinical MRI/Developments in MR*, 7(1):26–28, 1997.

[AF99a] John Ashburner and Karl J. Friston. Nonlinear spatial normalization using basis functions. *Human Brain Mapping*, 7(4):254–266, 1999.

[AF99b] John Ashburner and Karl J. Friston. Spatial normalization. In Toga [Tog99], chapter 2, pages 27–44.

[AHTK99a] Sigurd Angenent, Steven Haker, Allen Tannenbaum, and Ron Kikinis. Conformal geometry and brain flattening. In Taylor and Colchester [TC99], pages 271–278.

[AHTK99b] Sigurd Angenent, Steven Haker, Allen Tannenbaum, and Ron Kikinis. On the Laplace-Beltrami operator and brain surface flattening. *IEEE Transactions on Medical Imaging*, 18(8):700–711, 1999.

[AK96] Yali Amit and Augustine Kong. Graphical templates for model registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:225–236, 1996.

[AKM$^+$01] Alexandre Andrade, Ferath Kherif, Jean-Fran,cois Mangin, Keith J. Worsley, Anne-Lise Paradis, Olivier Simon, Stanislas Dehaene, Denis Le Bihan, and Jean-Baptiste Poline. Detection of fMRI activation using cortical surface mapping. *Human Brain Mapping*, 12:79–93, 2001.

[Ale00] Marc Alexa. Merging polyhedral shapes with scattered features. *The Visual Computer*, 16(1):26–37, 2000.

[Ami94] Yali Amit. A non-linear variational problem for image matching. *SIAM Journal on Scientific Computing*, 15(1):207–224, 1994.

[Ami97] Yali Amit. Graphical shape templates for automatic anatomy detection with applications to MRI brain scans. *IEEE Transactions on Medical Imaging*, 16(1):28–40, 1997.

[ANC$^+$97] J. Ashburner, P. Neelin, D. L. Collins, A. Evans, and K. Friston. Incorporating prior knowledge into image registration. *NeuroImage*, 6:344–352, 1997.

[AR95] Nur Arad and Daniel Reisfeld. Image warping using few anchor points and radial functions. *Computer Graphics Forum*, 14(1):23–29, 1995.

[Bad92] A. J. Baddeley. An error metric for binary images. In W. Förstner and H. Ruwiedel, editors, *Robust Computer Vision: Quality of Vision Algorithms*, pages 59–78. Wichmann, Karlsruhe, 1992.

[BB98] Mario Bertero and Patrizia Boccacci. *Introduction to Inverse Problems in Imaging*. Institute of Physics Publishing, 1998.

[BBP98] Hervé Brönnimann, Christoph Burnikel, and Sylvain Pion. Interval arithmetic yields efficient dynamic filters for computational geometry. In *Proc. 14th Annual ACM Symp. Comput. Geom.*, pages 165–174, 1998.

[BFS98] Christoph Burnikel, Stefan Funke, and Michael Seel. Exact geometric predicates using cascaded computation. In *Proc. 14th Annual ACM Symp. Comput. Geom.*, pages 175–183, 1998.

[BGKM98] M. Bakırcıoğlu, U. Grenander, N. Khaneja, and M. I. Miller. Curve matching in brain surfaces using Frenet distances. *Human Brain Mapping*, 6(5):329–333, 1998.

[Bic94] William B. Bickford. *A First Course in the Finite Element Method.* Richard D. Irwin, Inc., second edition, 1994.

[BJM99] M. M. Bakırcıoğlu, S. Joshi, and M. I. Miller. Landmark matching on brain surfaces via large deformation diffeomorphisms on the sphere. In *SPIE International Symposium on Medical Imaging*, San Diego, California, February 1999. SPIE.

[BJO02] Matthew Brett, Ingrid S. Johnsrude, and Adrian M. Owen. The problem of functional localization in the human brain. *Nature Reviews Neuroscience*, 3(3):243–249, 2002.

[BK89] Ruzena Bajcsy and Stane Kovačič. Multiresolution elastic matching. *Computer Vision, Graphics, and Image Processing*, 46:1–21, 1989.

[BM92] Paul J. Besl and Neil D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[BM98] Adrian Baddeley and Ilya Molchanov. Averaging of random sets based on their distance functions. *Journal of Mathematical Imaging and Vision*, 8:79–92, 1998.

[BMS96] Christoph Burnikel, Kurt Mehlhorn, and Stefan Schirra. The LEDA class real number. Technical Report MPI-I-96-1-001, Max-Planck-Institut für Informatik, January 1996.

[BNG96] Morten Bro-Nielsen and Claus Gramkow. Fast fluid registration of medical images. In *Proceedings of Visualization in Biomedical Computing*, volume 1131 of *Lecture Notes in Computer Science*, pages 267–276, 1996.

[Boo89] Fred L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, 1989.

[Boo91] Fred L. Bookstein. *Morphometric Tools for Landmark Data.* Cambridge University Press, 1991.

[Boo99] Fred L. Bookstein. Linear methods for nonlinear maps: Procrustes fits, thin-plate splines, and the biometric analysis of shape variability. In Toga [Tog99], chapter 10, pages 157–181.

[Bor84] G. Borgefors. Distance transformations in arbitrary dimensions. *Computer Vision, Graphics, and Image Processing*, 27:321–345, 1984.

[Bro92] Lisa Gottesfeld Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, December 1992.

[CA01a] Pascal Cachier and Nicholas Ayache. Regularization in image non-rigid registration: I. trade-off between smoothness and intensity similarity. Technical Report 4188, INRIA, 2001.

[CA01b] Pascal Cachier and Nicholas Ayache. Regularization methods in non-rigid registration: II. isotropic energies, filters and splines. Technical Report 4243, INRIA, August 2001.

[CDVE95] George J. Carman, Heather A. Drury, and David C. Van Essen. Computational methods for reconstructing and unfolding the cerebral cortex. *Cerebral Cortex*, 5:506–517, Nov/Dec 1995.

[CE97] D. L. Collins and A. C. Evans. ANIMAL: Validation and applications of nonlinear registration-based segmentation. *International Journal of Pattern Recognition and Artificial Intelligence*, 11(8):1271–1294, 1997.

[CE99] Louis Collins and A. C. Evans. ANIMAL: Automatic nonlinear image matching and anatomical labeling. In Toga [Tog99], chapter 8, pages 133–142.

[CET98] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In H. Burkhardt and B. Neumann, editors, *Proceedings of the European Conference on Computer Vision*, volume 2, pages 484–498. Springer, 1998.

[CET99] T. F. Cootes, G. J. Edwards, and C. J. Taylor. A unified framework for atlas matching using active appearance models. In Attila Kuba, Marin Šámal, and Andrew Todd-Pokropek, editors, *Information Processing in Medical Imaging*, volume 1613 of *Lecture Notes in Computer Science*, pages 322–333, Visegrád, Hungary, 1999. Springer.

[CGA03] CGAL. Computational geometry algorithms library. `http://www.cgal.org`, 2003.

[CH53] R. Courant and D. Hilbert. *Methods of Mathematical Physics*, volume 1. Interscience Publishers, New York, first english edition, 1953.

[CHK91] Hwan Soo Choi, David R. Haynor, and Yongmin Kim. Partial volume tissue classification of multichannel magnetic resonance images - a mixel model. *IEEE Transactions on Medical Imaging*, 10(3):395–407, 1991.

[CHPE95] D. L. Collins, C. J. Holmes, T. M. Peters, and A. C. Evans. Automatic 3-D model-based neuroanatomical segmentation. *Human Brain Mapping*, 3:190–208, 1995.

[Chr99] Gary E. Christensen. Bayesian framework for image registration using eigenfunctions. In Toga [Tog99], chapter 5, pages 85–100.

[CJM97] Gary E. Christensen, Sarang C. Joshi, and Michael I. Miller. Volumetric transformation of brain anatomy. *IEEE Transactions on Medical Imaging*, 16(6):864–877, 1997.

[CL00] Yongchoel Choi and Seungyong Lee. Injectivity conditions of 2D and 3D uniform cubic B-spline functions. *Graphical Models*, 62:411–427, 2000.

[CLGE98] D. Louis Collins, Georges Le Goualher, and Alan C. Evans. Non-linear cerebral registration with sulcal constraints. In *Medical Image Computing and Computer-Assisted Intervention*, volume 1496 of *Lecture Notes in Computer Science*, pages 974–984, Cambridge, USA, October 1998.

[CLR90] T.H. Cormen, C.L. Leiserson, and R.L. Rivest. *Introduction to Algorithms.* MIT Press, 1990.

[CMD$^+$95] A. Collingnon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, and G. Marchal. Automated multi-modality image registration based on information theory. In *Information Processing in Medical Imaging*, pages 263–274. Kluwer Academic Publishers, the Netherlands, 1995.

[CMGA92] Gary E. Christensen, Michael I. Miller, Ulf Grenander, and Yali Amit. Global shape models for anatomical structures. In Poor and Schwartz, editors, *Proceedings of the 26th Conference on Information Sciences and Systems*, page 365, Princeton University, March 1992.

[CMP$^+$01] Pascal Cachier, Jean-François Mangin, Xavier Pennec, Denis Rivière, Dimitri Papadopoulos-Orfanos, Jean Régis, and Nicholas Ayache. Multisubject non-rigid registration of brain MRI using intensity and geometric features. In Niessen and Viergever [NV01], pages 734–742.

[CMR$^+$03] A. Cachia, J.-F. Mangin, D. Riviére, D. Papadopoulos-Orfanos, F. Kherif, I. Bloch, and J. Régis. A generic framework for parcellation of the cortical surface into gyri using geodesic voronoï diagrams. *Medical Image Analysis*, 2003. In press.

[CNPE94] D. Louis Collins, Peter Neelin, Terrence M. Peters, and Alan C. Evans. Automatic 3D intersubject registration of MR volumetric data in standardized Talairach space. *Journal of Computer Assisted Tomography*, 18(2):192–205, March/April 1994.

[Coh96] Laurent D. Cohen. Auxiliary variables and two-step iterative algorithms in computer vision problems. *Journal of Mathematical Imaging and Vision*, 6:59–83, 1996.

[Col94] D. Louis Collins. *3D Model-Based Segmentation of Individual Brain Structures from Magnetic Resonance Imaging Data.* PhD thesis, McGill University, Montreal, October 1994.

[COR03] CORE. The core library project. `http://www.cs.nyu.edu/exact/core/`, 2003.

[Cox69] H. S. M. Coxeter. *Introduction to Geometry.* John Wiley and Sons, second edition, 1969.

[CP96] Simon R. Cherry and Michael E. Phelps. Imaging brain function with positron emission tomography. In Toga and Mazziotta [TM96], chapter 8, pages 191–221.

[CPA99] Pascal Cachier, Xavier Pennec, and Nicholas Ayache. Fast non rigid matching by gradient descent: Study and improvements of the "demons" algorithm. Technical Report 3706, INRIA, 1999.

[CPDE92] D. Louis Collins, Terry M. Peters, Weiqian Dai, and Alan C. Evans. Model based segmentation of individual brain structures from mri data. In Richard A. Robb, editor, *Proceedings of Visualization in Biomedical Computing*, volume 1808 of *SPIE International Symposium on Medical Imaging*, pages 10–23, Chapel Hill, North Carolina, October 1992.

[CPE94] D. Louis Collins, Terry M. Peters, and Alan C. Evans. An automated 3D non-linear deformation procedure for determination of gross morphometric variability human brain. In Richard A. Robb, editor, *Proceedings of Visualization in Biomedical Computing*, volume 2359 of *SPIE International Symposium on Medical Imaging*, pages 180–190, Rochester, Minnesota, October 1994.

[CRM93] Gary E. Christensen, Richard D. Rabbitt, and Michael I. Miller. A deformable neuroanatomy textbook based on viscous fluid mechanics. In Prince and Runolfsson, editors, *Proceedings of the 1993 Conference on Information Sciences and Systems*, pages 211–216, Johns Hopkins University, March 1993.

[CRM94] Gary E. Christensen, Richard D. Rabbitt, and Michael I. Miller. 3D brain mapping using a deformable neuroanatomy. *Physics in Medicine and Biology*, 39:609–618, 1994.

[CRM$^+$95] Gary Christensen, Richard D. Rabbitt, Michael I. Miller, Sarang C. Joshi, Ulf Grenander, Thomas A. Coogan, and David C. Van Essen. Topological properties of smooth anatomic maps. In *Information Processing in Medical Imaging*, pages 101–112, 1995.

[CRM96] Gary E. Christensen, Richard D. Rabbitt, and Michael I. Miller. Deformable templates using large deformation kinematics. *IEEE Transactions on Image Processing*, 5(10):1435–1447, 1996.

[CS03] Charles R. Collins and Kenneth Stephenson. A circle packing algorithm. *Computational Geometry: Theory and Applications*, 25:233–256, 2003.

[CSTM$^+$02] Fabrice Crivello, Thorsten Schormann, Nathalie Tzourio-Mazoyer, Per E. Roland, Karl Zilles, and Bernard M. Mazoyer. Comparison of spatial normalization procedures and their impact on functional maps. *Human Brain Mapping*, 16:228–250, 2002.

[CT91] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory.* John Wiley & Sons, New York, 1991.

[CT99] T. F. Cootes and C. J. Taylor. Statistical models of appearance for computer vision. Technical report, Imaging Science and Biomedical Engineering, University of Manchester, September 1999.

[CZPE01] D. Louis Collins, Alex P. Zijdenbos, Tomáš Paus, and Alan C. Evans. Use of registration for cohort studies. In Hajnal et al. [HHH01], chapter 14, pages 303–329.

[Dav96] Christos Davatzikos. Spatial normalization of 3D brain images using deformable models. *Journal of Computer Assisted Tomography*, 20(4):656–665, 1996.

[Dav97] Christos Davatzikos. Spatial transformation and registration of brain images using elastically deformable models. *Computer Vision and Image Understanding*, 66(2):207–222, May 1997.

[DB95] Chris Davatzikos and R. Nick Bryan. Using a deformable surface model to obtain a shape representation of the cortex. In *Proceedings of International Symposium on Computer Vision*, pages 212–217, Miami, November 1995.

[DB96] Christos Davatzikos and R. Nick Bryan. Using a deformable surface model to obtain a shape representation of the cortex. *IEEE Transactions on Medical Imaging*, 15(6):785–795, December 1996.

[DBETT99] Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs.* Prentis-Hall, Inc., 1999.

[dBvKOS97] Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational Geometry: Algorithms and Applications.* Springer-Verlag, Berlin, 1997.

[DECS99] Heather A. Drury, David C. Van Essen, Maurizio Corbetta, and Abraham Z. Snyder. Surface-based analyses of the human cerebral cortex. In Toga [Tog99], chapter 19, pages 337–361.

[DFS99] Anders M. Dale, Bruce Fischl, and Martin I. Sereno. Cortical surface-based analysis I: Segmentation and surface reconstruction. *NeuroImage*, 9(9):179–194, 1999.

[DGM97] Paul Dupuis, Ulf Grenander, and Michael I. Miller. Variational problems on flows of diffeomorphisms for image matching. Technical Report LCDS 97-4, LCDS, Division of Applied Mathematics, Brown University, February 1997.

[DGM98] Paul Dupuis, Ulf Grenander, and Michael I. Miller. Variational problems on flows of diffeomorphisms for image matching. *Quarterly of Applied Mathematics*, 56(3):687, 1998.

[DHS01] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification.* John Wiley and Sons, second edition, 2001.

[DILF99] J. Hunter Downs III, Jack L. Lancaster, and Peter T. Fox. Surface-based spatial normalization using convex hulls. In Toga [Tog99], chapter 15, pages 263–282.

[DS96] Jr. Dennis, J. E. and Robert B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, volume 16 of *Classics In Applied Mathematics.* Society for Industrial & Applied Mathematic, Philadelphia, 1996.

[DSFL$^+$99] F. Di Salle, E. Formisano, D.E.J. Linden, R. Goebel, S. Bonavita, A. Pepino, F. Smaltino, and G. Tedeschi. Exploring brain function with magnetic resonance imaging. *European Journal of Radiology*, 30:84–94, 1999.

[DSTA95] Jérôme Declerck, Gérard Subsol, Jean-Philippe Thirion, and Nicholas Ayache. Automatic retrieval of anatomical structures in 3D medical images. Technical Report 2485, INRIA, February 1995.

[DVEA$^+$96] Heather A. Drury, David C. Van Essen, Charles H. Anderson, Chris W. Lee, Thomas A. Coogan, and James W. Lewis. Computerized mappings of the cerebral cortex: A multiresolution flattening method and a surface-based coordinate system. *Journal of Cognitive Neuroscience*, 8(1):1–28, 1996.

[ECM$^+$93] A. C. Evans, D. L. Collins, S. R. Mills, E. D. Brown, R. L. Kelly, and T. M. Peters. 3D statistical neuroanatomical models from 305 MRI volumes. In *Proc. IEEE-Nuclear Science Symposium and Medical Imaging Conference*, pages 1813–1817, San Francisco, October 1993.

[EDD$^+$95] Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle. Multiresolution analysis of arbitrary meshes. In *Proceedings of SIGGRAPH'95*, pages 173–182, 1995.

[EHN96] Heinz W. Engl, Martin Hanke, and Andreas Neubauer. *Regularization of Inverse Problems.* Kluwer Academic Publishers, 1996.

[EKCM94] A.C. Evans, M. Kamber, D.L. Collins, and D. MacDonald. An MRI-based probabilistic atlas of neuroanatomy. In S.D. Shorvon et al., editor, *Magnetic Resonance Scanning and Epilepsy*, chapter 48, pages 263–274. Plenum Press, New York, 1994.

[EM94] Herbert Edelsbrunner and Ernst P. Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1):43–72, 1994.

[EMN$^+$92] Alan C. Evans, Sean Marrett, Peter Neelin, Louis Collins, Keith Worsley, Weiqian Dai, Sylvain Milot, Ernst Meyer, and Daniel Bub. Anatomical mapping of functional activation in stereotactic coordinate space. *NeuroImage*, 1(1):43–53, 1992.

[EO85] H. Edelsbrunner and M. H. Overmars. Batched dynamic solutions to decomposable searching problems. *J. Algorithms*, 6:515–542, 1985.

[FA96] Jacques Feldmar and Nicholas Ayache. Rigid, affine and locally affine registration of free-form surfaces. *International Journal of Computer Vision*, 18(2), May 1996.

[FAF$^+$95] K. J. Friston, J. Ashburner, C. D. Frith, J.-B. Poline, J. D. Heather, and R. S. J. Frackowiak. Spatial registration and normalization of images. *Human Brain Mapping*, 2:165–189, 1995.

[FLD01] Bruce Fischl, Arthur Liu, and Anders M. Dale. Automated manifold surgery: Constructing geometrically accurate and topologically correct models of the human cerebral cortex. *IEEE Transactions on Medical Imaging*, 20(1):70–80, January 2001.

[Flo97] Michael S. Floater. Parametrization and smooth approximation of surface triangulations. *Computer Aided Geometric Design*, 14:231–250, 1997.

[FMRR88] Peter T. Fox, Mark A. Mintun, Eric M. Reiman, and Marcus E. Raichle. Enhanced detection of focal brain responses using intersubject averaging and change-distribution analysis of subtracted PET images. *Journal of Cerebral Blood Flow and Metabolism*, 8(5):642–653, 1988.

[Fox95] Peter T. Fox. Spatial normalization origins: Objectives, applications, and alternatives. *Human Brain Mapping*, 3:161–164, 1995.

[FPR85] Peter T. Fox, Joel S. Perlmutter, and Marcus E. Raichle. A stereotactic method of anatomical localization for positron emission tomography. *Journal of Computer Assisted Tomography*, 9(1):141–153, 1985.

[Fre92] John E. Freund. *Mathematical Statistics*. Prentice-Hall, fifth edition, 1992.

[FRS99] Mike Fornefett, Karl Rohr, and H. Siegfried Stiehl. Radial basis functions with compact support for elastic registration of medical images. In *International Workshop on Biomedical Image Registration*, pages 173–185, 1999.

[FSD99] Bruce Fischl, Martin I. Sereno, and Anders M. Dale. Cortical surface-based analysis II: Inflation, flattening, and a surface-based coordinate system. *NeuroImage*, 9:195–207, 1999.

[FSTD99] Bruce Fischl, Martin I. Sereno, Roger B. H. Tootell, and Anders M. Dale. High-resolution inter-subject averaging and a coordinate system for the cortical surface. *Human Brain Mapping*, 8(4):272–284, 1999.

[FvDFH90] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, second edition, 1990.

[GAA97] James C. Gee, David C. Alsop, and Geoffrey K. Aguirre. Effect of spatial normalization on analysis of functional data. In K. M. Hanson, editor, *SPIE Medical Imaging 1997: Image Processing*, volume 3034 of *SPIE*, pages 312–322, 1997.

[GB99] James C. Gee and Ruzena K. Bajcsy. Elastic matching: Continuum mechanical and probabilistic analysis. In Toga [Tog99], chapter 11, pages 183–197.

[GBR+99] Igor D. Grachev, Dmitriy Berdichevsky, Scott L. Rauch, Stephan Heckers, David N. Kennedy, Verne S. Caviness, and Nathaniel M. Alpert. A method for assessing the accuracy of intersubject registration of the human brain using anatomic landmarks. *NeuroImage*, 9:250–268, 1999.

[GDT+02] Mark Galassi, Jim Davies, James Theiler, Brian Gough, Gerard Jungman, Michael Booth, and Fabrice Rossi. *GNU Scientific Library.* Network Theory Limited, second edition, 2002. Software available from `http://www.gnu.org/software/gsl/gsl.html`.

[Gee99] J. C. Gee. On matching brain volumes. *Pattern Recognition*, 32:99–111, 1999.

[GGD91] A.M. Goon, M.K. Gupta, and B. Dasgupta. *Fundamentals of Statistics*, volume 1. The World Press Private Ltd., Calcutta, sixth revised edition, 1991.

[GH99] James C. Gee and David R. Haynor. Numerical methods for high-dimensional warps. In Toga [Tog99], chapter 6, pages 101–113.

[GJC01] Guido Gerig, Matthieu Jomier, and Miranda Chakos. Valmet: A new validation tool for assessing and improving 3D object segmentation. In Niessen and Viergever [NV01], pages 516–523.

[GLB97] J. C. Gee and L. Le Briquer. An empirical model of brain shape. In G. Erickson, editor, *Maximum Entropy and Bayesian Methods.* Kluwer Academic, Dordrecht, 1997.

[GLM96] S. Gottschalk, Ming C. Lin, and D. Manocha. OBBTree: A hierarchical structure for rapid interference detection. In *ACM Siggraph*, 1996.

[GM98] Ulf Grenander and Michael I. Miller. Computational anatomy: An emerging discipline. *Quarterly of Applied Mathematics*, 56(4):617–684, 1998.

[GMT00] Alexandre Guimond, Jean Meunier, and Jean-Philippe Thirion. Average brain models: A convergence study. *Computer Vision and Image Understanding*, 77(2):192–210, 2000.

[Gos86] Ardeshir Goshtasby. Piecewise linear mapping functions for image registration. *Pattern Recognition*, 19(6):459–466, 1986.

[Gos87] Ardeshir Goshtasby. Piecewise cubic mapping functions for image registration. *Pattern Recognition*, 20(5):525–533, 1987.

[GP96] J. C. Gee and P. D. Peralta. Continuum models for Bayesian image matching. In K.M. Hanson and R.N. Silver, editors, *Maximum Entropy and Bayesian Methods*, pages 109–116. Kluwer Academic Publishers, 1996.

[GPS02] Herbert Goldstein, Charles Poole, and John Safko. *Classical Mechanics.* Addison Wesley, third edition, 2002.

[GR00] Marina Gavrilova and Jon Rokne. Reliable line segment intersection testing. *Computer-Aided Design*, 32:737–745, 2000.

[GRAM01] Alexandre Guimond, Alexis Roche, Nicholas Ayache, and Jean Meunier. Three-dimensional multimodal brain warping using the demons algorithm and adaptive intensity corrections. *IEEE Transactions on Medical Imaging*, 20(1):58–69, 2001.

[GT99] Maria Gabrani and Oleh J. Tretiak. Surface-based matching using elastic transformations. *Pattern Recognition*, 32:87–97, 1999.

[Gug63] Heinrich W. Guggenheimer. *Differential Geometry.* Dover, 1963.

[HAC$^+$02] P. Hellier, J. Ashburner, I. Corouge, C. Barillot, and K.J. Friston. Inter-subject registration of functional and anatomical data using SPM. In T. Dohi and R. Kikinis, editors, *Medical Image Computing and Computer-Assisted Intervention*, volume 2489 of *Lecture Notes in Computer Science*, pages 590–597, Tokyo, Japan, 2002. Springer.

[HB00] Pierre Hellier and Christian Barillot. Coupling dense and landmark-based approaches for non rigid registration. Technical Report 1368, IRISA, Rennes Cedex, France, November 2000.

[HBC$^+$01] P. Hellier, C. Barillot, I. Corouge, B. Gibaud, G. Le Goualher, L. Collins, A. Evans, G. Malandain, and N. Ayache. Retrospective evaluation of inter-subject brain registration. In Niessen and Viergever [NV01], pages 258–265.

[HBS$^+$99] Monica K. Hurdal, Philip L. Bowers, Ken Stephenson, De Witt L. Sumners, Kelly Rehm, Kirt Schaper, and David A. Rottenberg. Quasi-conformally flat mapping the human cerebellum. In Taylor and Colchester [TC99], pages 279–286.

[Hei95] Lennart Heimer. *The Human Brain and Spinal Cord: Functional Neuroanatomy and Dissection Guide.* Springer-Verlag, second edition, 1995.

[Hel97] Martin Held. ERIT: A collection of efficient and reliable intersection tests. *Journal of Graphics Tools*, 2(4):25–44, 1997.

[HHC$^+$98] Colin J. Holmes, Rick Hoge, Louis Collins, Roger Woods, Arthur W. Toga, and Alan C. Evans. Enhancement of MR images using registration for signal averaging. *Journal of Computer Assisted Tomography*, 22(2):324–333, 1998.

[HHH01] Joseph V. Hajnal, Derek L.G. Hill, and David J. Hawkes, editors. *Medical Image Registration.* Biomedical Engineering Series. CRC Press, 2001.

[HS80] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. A.I. Memo 572, MIT, AI Lab, April 1980.

[HS81] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[IAC99] Walid S. Ibrahim Ali and Fernand S. Cohen. 3D geometric invariant alignment of surfaces with application in brain mapping. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 238–243, 1999.

[Jäh99] Bernd Jähne. Image warping. In Bernd Jähne, Horst Haßbecker, and Peter Geißler, editors, *Handbook of Computer Vision and Applications Volume 2: Signal Processing and Pattern Recognition*, chapter 9, pages 193–206. Academic Press, 1999.

[JCD+99] Mukta Joshi, Jing Cui, Keith Doolittle, Sarang Joshi, David Van Essen, Lei Wang, and Michael I. Miller. Brain segmentation and the generation of cortical surfaces. *NeuroImage*, 9:461–476, 1999.

[JM00] Sarang C. Joshi and Michael I. Miller. Landmark matching via large deformation diffeomorphisms. *IEEE Transactions on Image Processing*, 9(8):1357–1370, 2000.

[JMW02] Sylvain Jaume, Benoît Macq, and Simon K. Warfield. Labeling the brain surface using a deformable multiresolution mesh. In T. Dohi and R. Kikinis, editors, *Medical Image Computing and Computer-Assisted Intervention*, volume 2488 of *Lecture Notes in Computer Science*, pages 451–458, Tokyo, Japan, 2002. Springer.

[Jos98] Sarang Joshi. *Large Deformation Diffeomorphisms and Gaussian Random Fields for Statistical Characterization of Brain Sub-Manifolds.* PhD thesis, Washington University, 1998.

[JTH+89] Marc L. Jouandet, Mark Jude Tramo, Daniel M. Herron, Allison Hermann, William C. Loftus, Josh Bazell, and Michael S. Gazzaniga. Brainprints: Computer-generated two-dimensional maps of the cerebral cortex in vivo. *Journal of Cognitive Neuroscience*, 1(1):88–117, 1989.

[KB99] Stanislav Kovačič and Ruzena Bajcsy. Multiscale/multiresolution representations. In Toga [Tog99], chapter 3, pages 45–65.

[KCP92] James R. Kent, Wayne E. Carlson, and Richard E. Parent. Shape transformation for polyhedral objects. *Computer Graphics*, 26(2):47–54, 1992. Proceedings of SIGGRAPH'92.

[Ket99] Lutz Kettner. Using generic programming for designing a data structure for polyhedral surfaces. *IEEE Computer Graphics and Applications*, 13:65–90, 1999.

[KG01] Nikolaus Kriegeskorte and Rainer Goebel. An efficient algorithm for topologically correct segmentation of the cortical sheet in anatomical MR volumes. *NeuroImage*, 14:329–346, 2001.

[KGF00] Stefan J. Kiebel, Rainer Goebel, and Karl J. Friston. Anatomically informed basis functions. *NeuroImage*, 11:656–667, 2000.

[Kir96] Andreas Kirsch. *An Introduction to the Mathematical Theory of Inverse Problems*, volume 120 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 1996.

[KLF99] Peter V. Kochunov, Jack L. Lancaster, and Peter T. Fox. Accurate high-speed spatial normalization using an octree method. *NeuroImage*, 19(6):724–737, 1999.

[KLGME01] Noor Kabani, Georges Le Goualher, David MacDonald, and Alan C. Evans. Measurement of cortical thickness using and automated 3-D algorithm: A validation study. *NeuroImage*, 13:375–380, 2001.

[KLN91] Michael Karasick, Derek Lieber, and Lee R. Nackman. Efficient delaunay triangulation using rational arithmetic. *ACM Transactions on Graphics*, 10(1):71–91, January 1991.

[KLPY99] V. Karamcheti, C. Li, I. Pechtchanski, and C. Yap. A core library for robust numeric and geometric computation. In *Proc. 15th Annual ACM Symp. Comput. Geom.*, pages 351–359, 1999.

[KLT$^+$00] P. Kochunov, J. Lancaster, P. Thompson, A. Boyer, J. Hardies, and P. Fox. Evaluation of octree regional spatial normalization method for regional anatomical matching. *Human Brain Mapping*, 11(3):193–206, 2000.

[KLT$^+$01] Peter Kochunov, Jack L. Lancaster, Paul Thompson, Roger Woods, John Mazziotta, Jean Hardies, and Peter Fox. Regional spatial normalization: Toward an optimal target. *Journal of Computer Assisted Tomography*, 25(5):805–816, 2001.

[KLT$^+$02] P. Kochunov, J. Lancaster, P. Thompson, A. W. Toga, P. Brewer, J. Hardies, and P. Fox. An optimized individual target brain in the talairach coordinate system. *NeuroImage*, 17(2):922–927, 2002.

[Kol96] Vasken Kollokian. Performance analysis of automatic techniques for tissue classification in magnetic resonance images of the human brain. Master's thesis, Computer Science, Concordia University, Montreal, Quebec, Canada, November 1996.

[KPH$^+$96] Ulrik Kjems, Peter A. Philipsen, Lars Kai Hansen, Chin-Tu Chen, and Jon Anderson. A nonlinear 3D MRI brain co-registration method. In P. C. Hansen, editor, *Proceedings of the Interdisciplinary Inversion Workshop 4*, Methodology and Applications in Geophysics, Astronomy, Geodesy and Physics, Technical University of Denmark, 1996.

[KS96] Benjamin B. Kimia and Kaleem Siddiqi. Geometric heat equation and nonlinear diffusion of shapes and images. *Computer Vision and Image Understanding*, 64(3):305–322, November 1996.

[KSK98] Takashi Kanai, Hiromasa Suzuki, and Fumihiko Kimura. Three-dimensional geometric metamorphosis based on harmonic maps. *The Visual Computer*, 14:166–176, 1998.

[LA99] Hava Lester and Simon R. Arridge. A survey of hierarchical non-linear medical image registration. *Pattern Recognition*, 32:129–149, 1999.

[Lan87] Serge Lang. *Linear Algebra.* Springer-Verlag, 1987.

[LBBE03] Jason Lerch, Neda Bernasconi, Andrea Bernasconi, and Alan Evans. Cortical atrophy in left temporal lobe epilepsy. In *9th International Conference on Functional Mapping of the Human Brain*, New York, June 2003. Available on CD-Rom in NeuroImage, Vol. 19, No. 2.

[LBS99] Stéphane Lavallée, Eric Bittar, and Richard Szeliski. Elastic registration and inference using oct-tree splines. In Toga [Tog99], chapter 16, pages 283–296.

[LC87] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *Computer Graphics*, 21(4):163–169, 1987.

[LCHS96] Seung-Yong Lee, Kyung-Yong Chwa, James Hahn, and Sung Yong Shin. Image morphing using deformation techniques. *Journal of Visualization and Computer Animation*, 7(1):3–23, 1996.

[LFD+99] Jack L. Lancaster, Peter T. Fox, Hunter Downs, Daniel S. Nickerson, Trish A. Hander, Mohammed El Mallah, Peter V. Kochunov, and Frank Zamarripa. Global spatial normalization of human brain using convex hulls. *The Journal of Nuclear Medicine*, 40(6):942–955, 1999.

[LGBB97] Georges Le Goualher, Christian Barillot, and Yves Bizais. Modeling cortical sulci with active ribbons. *International Journal of Pattern Recognition and Artificial Intelligence*, 11(8):1295–1315, 1997.

[LGPC+99] Georges Le Goualher, Emmanuel Procyk, D. Louis Collins, Raghu Venugopal, Christian Barillot, and Alan C. Evans. Automated extraction and variability analysis of sulcal neuroanatomy. *IEEE Transactions on Medical Imaging*, 18(3):206–217, 1999.

[LGS99] Thomas M. Lehmann, Claudia Gönner, and Klaus Spitzer. Survey: Interpolation methods in medical image processing. *IEEE Transactions on Medical Imaging*, 18(11):1049–1075, 1999. Addendum in volume 20, pages 660–665.

[Lin93] Ming C. Lin. *Efficient Collision Detection for Animation and Robotics.* PhD thesis, University of California, Berkeley, 1993.

[LMS01] Mark Lanthier, Anil Maheshwari, and Jörg-Rüdiger Sack. Approximating weighted shortest paths on polyhedral surfaces. *Algorithmica*, 30(4):527–562, 2001.

[LOB99] Tristan Lewis, Robyn Owens, and Adrian Baddeley. Averaging feature maps. *Pattern Recognition*, 32(9):331–339, 1999.

[Loh98] Gabriele Lohmann. Extracting line representations of sulcal and gyral patterns in MR images of the human brain. *IEEE Transactions on Medical Imaging*, 17(6):1040–1048, December 1998.

[LR75] David Lovelock and Hanno Rund. *Tensors, Differential Forms, and Variational Principles.* John Wiley & Sons, 1975.

[LvC98] Gabriele Lohmann and D. Yves von Cramon. Sulcal basins and sulcal strings as new concepts for describing the human cortical topography. In B. Vemuri, editor, *Workshop on Biomedical Image Analysis*, pages 24–33, 1998.

[LWCS96] Seungyong Lee, George Wolberg, Kyung-Yong Chwa, and Sung Yong Shin. Image metamorphosis with scattered feature constraints. *IEEE Transactions on Visualization and Computer Graphics*, 2(4):337–354, 1996.

[LY01] Chen Li and Chee Yap. Recent progress in exact geometric computation. http://cs.nyu.edu/exact/doc/dimacs.ps.gz, June 2001.

[LZG$^+$03] Jason Lerch, Alex Zijdenbos, Ryoi Goto, Yasuyuki Taki, Kazunori Sato, Ryuta Kawashima, Hiroshi Fukuda, and Alan Evans. Cortical thinning in normal aging. In *9th International Conference on Functional Mapping of the Human Brain*, New York, June 2003. Available on CD-Rom in NeuroImage, Vol. 19, No. 2.

[Mac98] J. D. MacDonald. *A Method for Identifying Geometrically Simple Surfaces from Three-Dimensional Images.* PhD thesis, School of Computer Science, McGill University, Canada, 1998.

[MAE94] David MacDonald, David Avis, and Alan C. Evans. Multiple surface identification and matching in magnetic resonance images. In Richard A. Robb, editor, *Proceedings of Visualization in Biomedical Computing*, volume 2359 of *SPIE International Symposium on Medical Imaging*, pages 160–169, Rochester, Minnesota, October 1994.

[MAE98] David MacDonald, David Avis, and Alan C. Evans. Proximity constraints in deformable models for cortical surface identification. In *Medical Image Computing and Computer-Assisted Intervention*, 1998.

[Mat75] G. Matheron. *Random Sets and Integral Geometry.* John Wiley and Sons, 1975.

[MBC$^+$97] Michael Miller, Ayananshu Banerjee, Gary Christensen, Sarang Joshi, Navin Khaneja, Ulf Grenander, and Larissa Matejic. Statistical methods in computational anatomy. *Statistical Methods in Medical Research*, 6:267–299, 1997.

[MCAG93] Michael I. Miller, Gary E. Christensen, Yali Amit, and Ulf Grenander. Mathematical textbook of deformable neuroanatomies. *Proceedings of the National Academy of Sciences, USA*, 90(24):11944–11948, December 1993.

[MFB+95] Jean-François Mangin, Vincent Frouin, Isabelle Bloch, Jean Régis, and Jaime López-Krahe. From 3D magnetic resonance images to structural representations of the cortex topography using topology preserving deformations. *Journal of Mathematical Imaging and Vision*, 5:297–318, 1995.

[Mir98] Brian Mirtich. V-clip: Fast and robust polyhedral collision detection. *ACM Transactions on Graphics*, 17(3):177–208, 1998.

[MJC99] Michael I. Miller, Sarang C. Joshi, and Gary E. Christensen. Large deformation fluid diffeomorphisms for landmark and image matching. In Toga [Tog99], chapter 7, pages 115–131.

[MKAE00] David MacDonald, Noor Kabani, David Avis, and Alan C. Evans. Automated 3-D extraction of inner and outer surfaces of cerebral cortex from MRI. *NeuroImage*, 12:340–356, 2000.

[MKFK94] Satoshi Minoshima, Robert A. Koeppe, Kirk A. Frey, and David E. Kuhl. Anatomic standardization: Linear scaling and nonlinear warping of functional brain images. *The Journal of Nuclear Medicine*, 35(9):1528–1537, 1994.

[MMV00] Erik G. Miller, Nicholas E. Matsakis, and Paul A. Viola. Learning from one example through shared densities on transforms. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 464–471, 2000.

[MN99] K. Mehlhorn and S. Näher. *The LEDA Platform of Combinatorial and Geometric Computing.* Cambridge University Press, 1999.

[MO63] G. H. Meisters and C. Olech. Locally one-to-one mappings and a classical theorem on schlicht functions. *Duke Mathematical Journal*, 30:63–80, 1963.

[Möl97] Tomas Möller. A fast triangle-triangle intersection test. *Journal of Graphics Tools*, 2(2):25–30, 1997.

[MR99] Henrich Müller and Detlef Ruprecht. Spatial interpolants for warping. In Toga [Tog99], chapter 12, pages 199–220.

[MRB$^+$95] J.-F. Mangin, J. Regis, I. Bloch, V. Frouin, Y. Samson, and J. Lopez-Krahe. A MRF based random graph modelling the human cortical topography. In *Computer Vision, Virtual Reality and Robotics in Medicine*, 1995.

[MS98] R. Malladi and J. A. Sethian. A real-time algorithm for medical shape recovery. In *Proceedings of International Conference on Computer Vision*, pages 304–310, Mumbai, India, January 1998.

[MTE$^+$95] John C. Mazziotta, Arthur W. Toga, Alan Evans, Peter Fox, and Jack Lancaster. A probabilistic atlas of the human brain: Theory and rationale for its development. *NeuroImage*, 2:89–101, 1995.

[MV98] J. B. Antoine Maintz and Max A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, 1998.

[NFD97] Mads Nielsen, Luc Florack, and Rachid Deriche. Regularization, scale-space, and edge detection filters. *Journal of Mathematical Imaging and Vision*, 7:291–307, 1997.

[Nis96] Dwight G. Nishimura. *Principles of Magnetic Resonance Imaging.* Department of Electrical Engineering, Stanford University, 1996.

[NV01] Wiro J. Niessen and Max A. Viergever, editors. *Medical Image Computing and Computer-Assisted Intervention*, volume 2208 of *Lecture Notes in Computer Science*, Utrecht, The Netherlands, 2001. Springer.

[NW99] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization.* Springer-Verlag, 1999.

[OKA90] Michio Ono, Stefan Kubik, and Chad D. Abernathey. *Atlas of the Cerebral Sulci.* Georg Thieme Verlag, Stuttgart, Germany, 1990.

[O'R98] Joseph O'Rourke. *Computational Geometry in C.* Cambridge University Press, second edition, 1998.

[OTU87] T. Ottmann, G. Theimt, and C. Ullrich. Numerical stability of geometric algorithms. In *Proceedings of the third annual symposium on Computational geometry*, pages 119–125. ACM Press, 1987.

[PFTV88] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, 1988.

[PGR96] Suguna Pappu, Steven Gold, and Anand Rangarajan. A framework for non-rigid matching and correspondence. *Advances in Neural Information Processing Systems*, 8:795–801, 1996.

[PM90] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, July 1990.

[PM91] M. H. Protter and C. B. Morrey. *A First Course in Real Analysis.* Springer-Verlag, second edition, 1991.

[PTK85] T. Poggio, V. Torre, and C. Koch. Computational vision and regularization theory. *Nature*, 317(26):314–319, 1985.

[RB01] Chris Rorden and Matthew Brett. Stereotaxic display of brain lesions. *Behavioural Neurology*, 12:191–200, 2001.

[RCB97] Anand Rangarajan, Haili Chui, and Fred L. Bookstein. The softassign procrustes matching algorithm. In James Duncan and Gene Gindi, editors, *Information Processing in Medical Imaging*, volume 1230 of *Lecture Notes in Computer Science*, pages 29–42, Vermont, USA, 1997. Springer.

[RCD99] Anand Rangarajan, Haili Chui, and James S. Duncan. Rigid point feature registration using mutual information. *Medical Image Analysis*, 4:1–17, 1999.

[RCRM01] Torsten Rohlfing and Jr. Calvin R. Maurer. Intensity based non rigid registration using adaptive multilevel free form deformation with an incompressibility constraint. In Niessen and Viergever [NV01], pages 111–119.

[RCSG93] J. Rademacher, V. S. Caviness, Jr., H. Steinmetz, and A. M. Galaburda. Topographical variation of the human primary cortices: Implications for neuroimaging, brain mapping, and neurobiology. *Cerebral Cortex*, 3:313–329, 1993.

[RDFR99] Nicolas Royackkers, Michel Desvignes, Houssam Fawal, and Marinette Revenu. Detection and statistical analysis of human cortical sulci. *NeuroImage*, 10:625–641, 1999.

[RECW03] Steven Robbins, Alan C. Evans, D. Louis Collins, and Sue Whitesides. Tuning and comparing spatial normalization methods. In Randy E. Ellis and Terry M. Peters, editors, *Medical Image Computing and Computer-Assisted Intervention*, volume 2879 of *Lecture Notes in Computer Science*, pages 910–917, Montreal, Canada, November 2003. Springer.

[RFD+95] Nicolas Royackkers, Houssam Fawal, Michel Desvignes, Marinette Revenu, and Jean marcel Travere. Morphometry and identification of brain sulci on three-dimensional MR images. In Yves Bizais, Christian Barillot, and Rober Di Paola, editors, *Information Processing in Medical Imaging*, Lecture Notes in Computer Science, pages 379–380. Springer, 1995.

[RFS01] D. Rueckert, A.F. Frangi, and J.A. Schnabel. Automatic construction of 3D statistical deformation models using non-rigid registration. In Niessen and Viergever [NV01], pages 77–84.

[RGA+97] P. E. Roland, Stefan Geyer, Katrin Amunts, Thorsten Schormann, Axel Schleicher, Aleksander Malikovic, and Karl Zilles. Cytoarchitectural maps of the human brain in standard anatomical space. *Human Brain Mapping*, 5:222–227, 1997.

[RGW+94] P.E. Roland, C.J. Graufelds, J. Wåhlin, L. Ingelman, M. Andersson, A. Ledberg, J. Pedersen, S. Åkerman, A. Dabringhaus, and K. Zilles. Human brain atlas: For high-resolution functional and anatomical mapping. *Human Brain Mapping*, 1:173–184, 1994.

[RHP00] Maryam E. Rettmann, Xio Han, and Jerry L. Prince. Watersheds on the cortical surface for automated sulcal segmentation. In *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, pages 20–27, 2000.

[RHXP02] Maryam E. Rettmann, Xio Han, Chenyang Xu, and Jerry L. Prince. Automated sulcal segmentation using watersheds on the cortical surface. *NeuroImage*, 15(2):329–344, 2002.

[RM95] Detlef Ruprecht and Henrich Müller. Image warping with scattered data interpolation. *IEEE Computer Graphics and Applications*, 15(2):37–43, 1995.

[RMA99] Alexis Roche, Grégoire Malandain, and Nicholas Ayache. Unifying maximum likelihood approaches in medical image registration. Technical Report 3741, INRIA, France, July 1999.

[RMA00] Alexis Roche, Grégoire Malandain, and Nicholas Ayache. Unifying maximum likelihood approaches in medical image registration. *International Journal of Imaging Systems and Technology*, 11:71–80, 2000.

[RMPA98] Alexis Roche, Grégoire Malandain, Xavier Pennec, and Nicholas Ayache. Multimodal image registration by maximization of the correlation ratio. Technical Report 3378, INRIA, France, August 1998.

[RMPO$^+$00] Denis Rivière, Jean-François Mangin, Dimitri Papadopoulos-Orfanos, Jean-Marc Martinez, Vincent Frouin, and Jean Régis. Automatic recognition of cortical sulci using a congregation of neural networks. In Scott L. Delp, Anthony M. DiGioia, and Branislav Jaramaz, editors, *Medical Image Computing and Computer-Assisted Intervention*, volume 1935 of *Lecture Notes in Computer Science*, pages 40–49, Pittsburgh, USA, 2000. Springer.

[RMPO$^+$02] Denis Rivière, Jean-François Mangin, Dimitri Papadopoulos-Orfanos, Jean-Marc Martinez, Vincent Frouin, and Jean Régis. Automatic recognition of cortical sulci using a congregation of neural networks. *Medical Image Analysis*, 6(2):77–92, 2002.

[RSH$^+$99] D. Rueckert, L.I. Sonoda, C. Hayes, D.L.G. Hill, M.O. Leach, and D.J. Hawkes. Nonrigid registration using free-form deformations: Applications to breast MR images. *IEEE Transactions on Medical Imaging*, 18(8):712–721, 1999.

[Ruc96] William Rucklidge. *Efficient Visual Recognition Using the Hausdorff Distance*, volume 1173 of *Lecture Notes in Computer Science*. Springer, 1996.

[RW03] Steven Robbins and Sue Whitesides. On the reliability of triangle intersection in 3D. In Vipin Kumar, Marina L. Gavrilova, Chih Jeng Kenneth Tan, and Pierre L'Ecuyer, editors, *Computational Science and Its*

Applications, volume 2669 of *Lecture Notes in Computer Science*, pages 923–930, Montreal, Canada, May 2003.

[Saa87] Alan Saalfeld. Joint triangulations and triangulation maps. In *Proc. 3rd Annual ACM Symp. Comput. Geom.*, pages 195–204, 1987.

[Sav01] Robert L. Savoy. History and future directions of human brain mapping and functional neuroimaging. *Acta Psychologica*, 107:9–42, 2001.

[SAVK$^+$02] C.H. Salmond, J. Ashburner, F. Vargha-Khadem, A. Connelly, D.G. Gadian, and K.J. Friston. The precision of anatomical normalization in the medial temporal lobe using spatial basis functions. *NeuroImage*, 17:507–512, 2002.

[SBK$^+$92] G. Szèkely, Ch. Brechbühler, O. Kübler, R. Ogniewicz, and T. Budinger. Mapping the human cerebral cortex using 3D medial manifolds. In *Proceedings of Visualization in Biomedical Computing*, pages 130–144, 1992.

[SC97] Richard Szeliski and James Coughlan. Spline-based image registration. *International Journal of Computer Vision*, 22(3):199–218, 1997.

[SD02] Dinggang Shen and Christos Davatzikos. HAMMER: Hierarchical attribute matching mechanism for elastic registration. *IEEE Transactions on Medical Imaging*, 21(11):1421–1439, 2002.

[SD03] Dinggang Shen and Christos Davatzikos. Very high-resolution morphometry using mass-preserving deformations and HAMMER elastic registration. *NeuroImage*, 18(3):28–41, 2003.

[Ser82] J. Serra. *Image Analysis and Mathematical Morphology.* Academic Press, 1982.

[Set96] J. A. Sethian. *Level Set Methods.* Cambridge University Press, 1996.

[SFF89] Helmuth Steinmetz, Günter Fürst, and Hans-Joachim Freund. Cerebral cortical localization: Application and validation of the proportional grid system in MR imaging. *Journal of Computer Assisted Tomography*, 13(1):10–19, January/February 1989.

[She92] Helen Sherk. Flattening the cerebral cortex by computer. *Journal of Neuroscience Methods*, 41:255–267, 1992.

[She97] Jonathan Richard Shewchuk. Adaptive precision floating-point arithmetic and fast robust geometric predicates. *Discrete and Computational Geometry*, 18(3):305–363, October 1997.

[SHH99] C. Studholme, D.L.G. Hill, and D.J. Hawkes. An overlap invariant entropy measure of 3D medical image alignment. *Pattern Recognition*, 32:71–86, 1999.

[SIO$^+$98] Michio Senda, Kenji Ishii, Keiichi Oda, Norihiro Sadato, Ryuta Kawashima, Motoaki Sgiura, Iwao Kanno, Babak Ardekani, Satoshi Minoshima, and Itaru Tatsumi. Influence of ANOVA design and anatomical standardization on statistical mapping for PET activation. *NeuroImage*, 8:283–301, 1998.

[SKS$^+$99] Motoaki Sugiura, Ryuta Kawashima, Norihiro Sadato, Michio Senda, Iwao Kanno, Keiichi Oda, Kazunori Sato, Yoshiharu Yonekura, and Hiroshi Fukuda. Anatomic validation of spatial normalization methods for PET. *The Journal of Nuclear Medicine*, 40(2):317–322, February 1999.

[SL97] S. Sandor and R. Leahy. Surface-based labeling of cortical anatomy using a deformable atlas. *IEEE Transactions on Medical Imaging*, 16(1):41–54, 1997.

[SL01] David W. Shattuck and Richard M. Leahy. Automated graph-based analysis and correction of cortical volume topology. *IEEE Transactions on Medical Imaging*, 20(11):1167–1177, 2001.

[Smi02] Stephen M. Smith. Fast robust automated brain extraction. *Human Brain Mapping*, 17(3):143–155, 2002.

[SP86] Thomas W. Sederberg and Scott R. Parry. Free-form deformation of solid geometric models. In *Proceedings of SIGGRAPH'86*, volume 20 of *Computer Graphics*, pages 151–160. ACM Press, 1986.

[SPM03] SPM. Statistical parameter mapping software. `http://www.fil.ion.ucl.ac.uk/spm/`, 2003. Wellcome Department of Imaging Neuroscience.

[SS94] Dietrich Stoyan and Helga Stoyan. *Fractals, Random Shapes and Point Fields: Methods of Geometrical Statistics.* John Wiley & Sons, 1994.

[Sub99] Gérard Subsol. Crest lines for curve-based warping. In Toga [Tog99], chapter 14, pages 241–262.

[SZE98] John G. Sled, Alex P. Zijdenbos, and Alan C. Evans. A nonparametric method for automatic correction of intensity nonuniformity in MRI data. *IEEE Transactions on Medical Imaging*, 17(1):87–97, 1998.

[Tau95] Gabriel Taubin. Estimating the tensor of curvature of a surface froma a polyhedral approximation. In *Proceedings of International Conference on Computer Vision*, page 902, 1995.

[TC99] Chris Taylor and Alan C. F. Colchester, editors. *Medical Image Computing and Computer-Assisted Intervention*, volume 1679 of *Lecture Notes in Computer Science*, Cambridge, UK, 1999. Springer.

[TG93] Jean-Philippe Thirion and Alexis Gouron. The marching lines algorithm: New results and proofs. Technical Report 1881, INRIA, 1993.

[Thi98] J.-P. Thirion. Image matching as a diffusion process: an analogy with Maxwell's demons. *Medical Image Analysis*, 2(3):243–260, 1998.

[Thi99] Jean-Philippe Thirion. Diffusing models and applications. In Toga [Tog99], chapter 9, pages 143–155.

[TM96] Arthur W. Toga and John C. Mazziotta, editors. *Brain Mapping: the Methods.* Academic Press, 1996.

[TMM+97] Paul M. Thompson, David MacDonald, Michael S. Mega, Colin J. Holmes, Alan C. Evans, and Arthur W. Toga. Detection and mapping of abnormal brain structure with a probabilistic atlas of cortical surfaces. *Journal of Computer Assisted Tomography*, 21(4):567–581, 1997.

[TMV+01] Paul M. Thompson, Michael S. Mega, Christine Vidal, Judith L. Rapoport, and Arthur W. Toga. Detecting disease-specific patterns of brain structure using cortical pattern matching and a population-based probabilistic brain atlas. In *Information Processing in Medical Imaging*, volume 2082 of *Lecture Notes in Computer Science*, pages 488–501, 2001.

[Tog99] Arthur W. Toga, editor. *Brain Warping.* Academic Press, San Diego, CA, 1999.

[TST$^+$67] J. Talairach, G. Szikla, P. Tournoux, A. Prossalentis, M. Bordas-Ferrer, L. Covello, M. Iacob, and E. Mempel. *Atlas d'Anatomie Stéréotaxique du Télencéphale.* Masson, 1967.

[TST96] P. M. Thompson, C. Schwartz, and A. W. Toga. High-resolution random mesh algorithms for creating a probabilistic 3D surface atlas of the human brain. *NeuroImage*, 3:19–34, 1996.

[TT88] Jean Talairach and Pierre Tournoux. *Co-Planar Stereotaxic Atlas of the Human Brain.* Georg Thieme Verlag, Stuttgart, Germany, 1988.

[TT96] Paul Thompson and Arthur W. Toga. A surface-based technique for warping three-dimensional images of the brain. *IEEE Transactions on Medical Imaging*, 15(4):402–417, August 1996.

[TT99] Paul Thompson and Arthur W. Toga. Anatomically driven strategies for high-dimensional brain image warping and pathology detection. In Toga [Tog99], chapter 18, pages 311–336.

[TT01] A.W. Toga and P.M. Thompson. The role of image registration in brain mapping. *Image and Vision Computing*, 19:3–24, 2001.

[TWMT00] Paul M. Thompson, Roger P. Woods, Michael S. Mega, and Arthur W. Toga. Mathematical/computational challenges in creating deformable and probabilistic atlases of the human brain. *Human Brain Mapping*, 9:81–92, 2000.

[VD99] Marc Vaillant and Christos Davatzikos. Hierarchical matching of cortical features for deformable brain image registration. In *Information Processing in Medical Imaging*, volume 1613 of *Lecture Notes in Computer Science*, pages 182–195, June 1999.

[VDB96] Marc Vaillant, Christos Davatzikos, and R. Nick Bryan. Finding 3D parametric representations of the deep cortical folds. In *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, pages 151–159, June 1996.

[VE02] David C. Van Essen. Windows on the brain: the emerging role of atlases and databases in neuroscience. *Current Opinion in Neurobiology*, 12:574–579, 2002.

[VEDD+01] David C. Van Essen, Heather A. Drury, James Dickson, John Harwell, Donna Hanlon, and Charles H. Anderson. An integrated software suite for surface based analyses of cerebral cortex. *Journal of the American Medical Informatics Association*, 8(5):443–459, 2001.

[VEDJM98] David C. Van Essen, Heather A. Drury, Sarang Joshi, and Michael I. Miller. Functional and structural mapping of human cerebral cortex: Solutions are in the surfaces. *Proceedings of the National Academy of Sciences, USA*, 95:788–795, February 1998.

[VELD+01] David C. Van Essen, James W. Lewis, Heather A. Drury, Nouchine Hadjikhani, Roger B.H. Tootell, Muge Bakircioglu, and Michael I. Miller. Mapping visual cortex in monkeys and humans using surface-based atlases. *Vision Research*, 41:1359–1378, 2001.

[VH99] R. Veltkamp and M. Hagedoorn. State-of-the-art in shape matching. Technical Report UU-CS-1999-27, Utrecht University, the Netherlands, 1999. `http://www.cs.uu.nl/research/techreps/UU-CS-1999-27.html`.

[WCB00] Brian A. Wandell, Suelika Chial, and Benjamin T. Backus. Visualization and measurement of the cortical surface. *Journal of Cognitive Neuroscience*, 12(5):739–752, 2000.

[WCM92] Roger P. Woods, Simon R. Cherry, and John C. Mazziotta. Rapid automated algorithm for aligning and reslicing PET images. *Journal of Computer Assisted Tomography*, 16(4):620–633, July/August 1992.

[WGH+98] Roger P. Woods, Scott T. Grafton, Colin J. Holmes, Simon R. Cherry, and John C. Mazziotta. Automated image registration: I. general methods and intrasubject, intramodality validation. *Journal of Computer Assisted Tomography*, 22(1):139–152, 1998.

[WGW+98] Roger P. Woods, Scott T. Grafton, John D. G. Watson, Nancy L. Sicotte, and John C. Mazziotta. Automated image registration: II. intersubject validation of linear and nonlinear models. *Journal of Computer Assisted Tomography*, 22(1):153–165, 1998.

[WIVA+96] William M. Wells III, Paul Viola, Hideki Atsumi, Shin Nakajima, and Ron Kikinis. Multi-modal volume registration by maximization of mutual information. *Medical Image Analysis*, 1(1):35–51, 1996.

[WMC93] R. P. Woods, J. C. Mazziotta, and S. R. Cherry. MRI-PET registration with automated algorithm. *Journal of Computer Assisted Tomography*, 17:536–546, 1993.

[Woo96] Roger P. Woods. Correlation of brain structure and function. In Toga and Mazziotta [TM96], chapter 12, pages 313–341.

[Woo99] Roger P. Woods. Automated global polynomial warping. In Toga [Tog99], chapter 20, pages 365–376.

[Wor97] Keith J. Worsley. An overview and some new developments in the statistical analysis of PET and fMRI data. *Human Brain Mapping*, 5:254–258, 1997.

[WPS00] Yongmei Wang, Bradley S. Peterson, and Lawrence H. Staib. Shape-based 3D surface correspondence using geodesics and local geometry. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 644–651, 2000.

[WRH+01] Simon K. Warfield, Jan Rexilius, Petra S. Huppi, Terrie E. Inder, Erik G. Miller, William M. Wells III, Gary P. Zientara, Ferenc A. Jolesz, and Ron Kikinis. A binary entropy measure to assess nonrigid registration algorithms. In Niessen and Viergever [NV01], pages 266–274.

[WS98] Yongmei Wang and Lawrence H. Staib. Integrated approaches to non-rigid registration in medical images. In *Workshop on Applications of Computer Vision*, pages 102–108, Princeton, NJ, October 1998. IEEE.

[XPR+99] Chenyang Xu, Dzung L. Pham, Maryam E. Rettmann, Daphne N. Yu, and Jerry L. Prince. Reconstruction of the human cerebral cortex from magnetic resonance images. *IEEE Transactions on Medical Imaging*, 18(6):467–480, 1999.

[Yap97] Chee K. Yap. Robust geometric computation. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry*. CRC Press, 1997.

[YD95] Chee Yap and Thomas Dubé. The exact computation paradigm. In Ding-Zhu Du and Frank Hwang, editors, *Computing in Euclidean Geometry*. World Scientific, second edition, 1995.

[ZE02] Afra Zomorodian and Herbert Edelsbrunner. Fast software for box intersections. *International Journal of Computational Geometry and Applications*, 12:143–172, 2002.

[ZFE02] Alex Zijdenbos, Reza Forghani, and Alan Evans. Automatic "pipeline" analysis of 3-D MRI data for clinical trials: Application to multiple sclerosis. *IEEE Transactions on Medical Imaging*, 21(10):1280–1291, 2002.

[ZSS$^+$99] X. Zeng, L. H. Staib, R. T. Schultz, H. Tagare, L. Win, and J. S. Duncan. A new approach to 3D sulcal ribbon finding from MR images. In Taylor and Colchester [TC99], pages 148–157.

[ZTT99] Yong Zhou, Paul M. Thompson, and Arthur W. Toga. Extracting and representing the cortical sulci. *IEEE Computer Graphics and Applications*, 19(3):49–55, May 1999.

Index