January 2010

Magnetic resonance imaging and surface-based analysis techniques to quantify growth of the developing brain

Andrew Knutsen
Washington University in St. Louis

WASHINGTON UNIVERSITY IN ST. LOUIS

Department of Mechanical, Aerospace and Structural Engineering

Dissertation Examination Committee:
Philip Bayly, Chair
Guy Genin
Cindy Grimm
Terrie Inder
Jeffrey Neil
Larry Taber
David Van Essen

MAGNETIC RESONANCE IMAGING AND SURFACE-BASED ANALYSIS

TECHNIQUES TO QUANTIFY GROWTH OF THE DEVELOPING BRAIN

by

Andrew Knut Knutsen

A dissertation presented to the
Graduate School of Arts and Sciences
of Washington University in
partial fulfillment of the
requirements for the degree
of Doctor of Philosophy

August 2010

Saint Louis, Missouri

Abstract

Human brains, and those of most higher mammals, are gyrencephalic (folded) to accommodate a large cortical surface within the limited volume of the skull. Abnormal folding of the cerebral cortex in humans is associated with a number of neurological dysfunctions and diseases such as schizophrenia and Williams syndrome. To understand the mechanism of gyrification, and to illuminate the underlying causes of abnormal folding, objective, quantitative methods to characterize normal and abnormal development must be developed. The ferret is an excellent model in which to study the development of convolutions in the brain because folding occurs post-natally over a period of several weeks, and the brain can be imaged conveniently in small-animal magnetic resonance (MR) scanners. Here, MR imaging was used to acquire three-dimensional image volumes of the ferret brain *in vivo* at different stages during the period of cortical folding. Through segmentation of these volumes, surface representations of the cortex are generated at each time point. A novel intra-subject registration algorithm (LAndmark Correspondence and Relaxation Of Surface Strain: LACROSS), which provides a point-to-point correspondence between two surfaces, is applied to the cortical surfaces from two ferret kits. The resulting calculations of growth show regional patterns within the cortex, and temporal variations over this period of early brain development.

Acknowledgements

Table of Contents

List of Figures and Tables

Abbreviations

TE – echo time

TR – repetition time

MRI – magnetic resonance imaging

DTI – diffusion tensor imaging

OCT – optical coherence tomography

CT – computed tomography

GM – grey matter

WM – white matter

CSF – cerebrospinal fluid

PDE – partial differential equation

RA – relative anisotropy

FA – fractional anisotropy

YAS – younger anatomical surface

OAS – older anatomical surface

YSS – younger spherical surface

OSS – older spherical surface

ROSS – relaxed older spherical surface

ROAS – relaxed older anatomical surface

Chapter I. Introduction

"Most anatomists must be content with the presentation of these developmental problems in the hope that they may enlist the sympathetic attention of the physicist or mechanical engineer for assistance in their elucidation."

W.E. Le Gros Clark

Abnormal folding of the cerebral cortex in the human brain is associated with abnormal neurologic disease and dysfunction such as schizophrenia (Cernansky et al., 2008; Voets et al., 2008; Wisco et al., 2007), attention-deficit/hyperactivity disorder (ADHD) (Wolosin et al., 2009), William's Syndrome (Thompson et al., 2005; Van Essen et al., 2006), bipolar disorder (Fornito et al., 2007) and others (Pang et al., 2008). In addition, we know that the brain develops differently in the neonatal intensive care unit than it does *in utero* (Inder et al., 2005). Development of the brain *ex utero* for preterm infants born before 37 weeks gestational age (GA) is also associated with an increased risk for cerebral palsy, learning disabilities and behavioral disorders (Hack et al., 2000; Stjernqvist et al., 1999). In addition to developmental delays or neurologic disease or dysfunction, a high financial cost is also associated with being born prematurely or at a low birth weight (Petrou et al., 2000). However, the underlying mechanisms for normal folding of the brain are not well understood.

Measures of spatial and temporal variations in growth are needed to characterize the morphogenetic processes of the folding of the cerebral cortex (Van Essen, 1997). Advances in anatomical imaging techniques, such as computed tomography (CT), magnetic resonance imaging (MRI) and optical coherence tomography (OCT) allow high

resolution images of the brain to be acquired. Repetition of scans at multiple time points provides a sequence of snapshots during development, which can be used to analyze growth of the cortex.

1.1 Magnetic resonance imaging

Magnetic resonance imaging (MRI) is a non-invasive tool used to acquire images without the use of radiation. Magnetic resonance represents the net absorption of electromagnetic energy by nuclei of a naturally-occurring, non-radioactive isotope, which is referred to as a nuclear spin system. If a nuclear spin system (e.g., anything containing water) is placed in a strong magnetic field it is possible to excite the nuclei using a radiofrequency (RF) pulse. The input of energy into the system causes some nuclei to jump between energy states as predicted by quantum mechanics. The excitation of the spin system creates a non-equilibrium spin state, which can be detected using an RF antenna or receiver.

A macroscopic description of MR is defined through the use of spin packets, which is a group of spins that experience the same magnetic field that can be represented by a magnetization vector (Hornak, 1996). In equilibrium, the magnetization vector points along the direction of the magnetic field. The magnetization vector can be removed from equilibrium by introducing RF energy that is equal to the difference in energy between the two spin states. The return to equilibrium after excitation is governed by the time constants T_1 and T_2 (Fig. 1), which depend on environmental factors such as surrounding macromolecules and viscosity (Mori and Zhang, 2006). In imaging the differences in T_1 and T_2 between tissues types provide contrast. In adults the T_2 value of

cerebrospinal fluid (CSF) is longer than white matter (WM) which is longer than grey matter (GM). A T$_2$-weighted image of an adult ferret brain is shown in Fig. 1.2.

Diffusion tensor imaging (DTI) is an MR imaging technique that provides a 3-D description of the microstructural anatomy. Diffusion is represented by translational water motion, which reflects the static anatomy and is less influenced by physiology (Mori and Zhang, 2006). The diffusion tensor, D, is estimated using numerical techniques from a series of diffusion weighted images. The diffusion tensor is a function of a relationship between the measured echo attenuation in each voxel and the applied magnetic field sequence (Bassar and Jones, 2002).

The eigenvalues and eigenvecetors of the diffusion tensor D are calculated by solving the eigenvalue problem. The apparent diffusion coefficient (ADC) is a measure of the magnitude of water diffusion, and is given by

$$ADC = \bar{\lambda} = \frac{\lambda_1 + \lambda_2 + \lambda_3}{3} \text{,} \qquad \text{I.1}$$

where λ_i are the eigenvalues of D. Relative anisotropy (RA) is a normalized measure that reflects whether diffusion is preferential to a particular direction (anisotropic, high RA) or whether it diffuses equally in all directions (isotropic, low RA). RA is given by

$$RA = \sqrt{\frac{\left(\lambda_1 - \bar{\lambda}\right)^2 + \left(\lambda_2 - \bar{\lambda}\right)^2 + \left(\lambda_3 - \bar{\lambda}\right)^2}{6\bar{\lambda}}} \text{.} \qquad \text{I.2}$$

Another measure of anisotropy is Fractional Anisotropy (FA), which is given by

$$FA = \sqrt{\frac{3\left(\lambda_1 - \bar{\lambda}\right)^2 + \left(\lambda_2 - \bar{\lambda}\right)^2 + \left(\lambda_3 - \bar{\lambda}\right)^2}{2\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}}} \text{.} \qquad \text{I.3}$$

Anatomical MRI and DTI are powerful tools that provide insight into the developing brain. Fig. 1.2 shows a plot of ADC and RA from an adult ferret. Another

method to visualize diffusion is from RGB plots. In these plots, red corresponds to medial-lateral diffusion, green corresponds to posterior-anterior diffusion and blue corresponds to inferior-superior diffusion. An RGB plot from an adult ferret is shown in Fig. 1.2.

1.2 Cortical development in humans

The cerebral cortex is made up of six layers of neurons that are intersected by independent columns of neurons (Rakic, 2000). Most neurons are not born in the cortex, but instead are created by progenitor cells that are born in either the ventricular zone (VZ) or the subventricular zone (SVZ). Progenitor cells first undergo symmetric mitosis, which impacts the eventual surface area of the cortex. After symmetric mitosis is complete, cells undergo asymmetric mitosis, which produces one multipotent cell that remains in the VZ/SVZ and the other cell eventually migrates out. Asymmetric cell division affects the thickness of the cortex. Neuronal progenitor cells born from asymmetric cell division migrate out of the VZ/SVZ on radial glia cells, through the intermediate zone (IZ) to the cortical plate (CP), which eventually becomes the mature cortex (Rakic, 2006). The neurons from layers II-VI form the cortex in an inside-out fashion, where the layer VI neurons arrive first and the layer II neurons arrive last. By 25 weeks gestational age (GA), neuronogenesis, neuronal migration and differentiation is essentially complete (Rakic, 1988).

After neuronal migration is complete, cortico-cortico and cortico-thalamic connections begin to form. Synaptogenesis along with neuronal growth and differentiation contribute to growth of the cerebral cortex that occurs after 25 weeks GA

(Welker, 1990). From 24-32 wks PMA, permanent thalmocortical fibers in certain functional regions (frontal, somatosensory, visual and auditory) begin to form synapses within the cortical plate (Kostovic and Judas, 2002). From 33-35 wks PMA, long distance cortico-cortical and callosal connections begin to form within the cortical plate (Jovanov-Milosevic et al., 2006). During this time the first short distance cortico-cortical connections begin to appear as well. By term most long distance cortico-cortico connections have formed (Krostovic and Jovanov-Milosevic, 2006). The major postnatal events are synaptogenesis, synaptic pruning, dendritic arborization and myelination. Most synapses are formed postnatally (Kostovic and Jovanov-Milosevic, 2006). Regional variations in synaptic density (Huttenlocher, 1990; Huttenlocher and Dabholkar, 1997), dendritic spine density and dendritic length (Travis et al., 2005) are present at around term. Also, regional variations in neuronal density, dendritic size, branching complexity and spine density are also seen in the adult (Elston et al., 2001; Haug, 1987).

Development on a macroscopic scale is highlighted by the folding and growth of the cortex. The greatest amount of folding occurs during the third trimester (Dubois et al., 2008). At the start of the third trimester, the cortex is very smooth and resembles an oven mitt. By term, which is at 38-40 weeks GA, almost every fold is present in an incipient form (Chi et al., 1977). Increased cortical surface area may be necessary for more complex operations (Welker, 1990). A folded cortex allows for a larger surface area to fit into a set volume. In mammals, an increase in the size of the body is associated with an increase in the size of the brain (Hoffman, 1989). If throughout evolution cortical surface area is geometrically similar, then the surface area should increase to the two-thirds power of brain volume, which is the rate at which a sphere increases in size (Hoffman,

1989). However, what is seen is that the surface area increases at almost the first power of brain volume. So, as brain volumes have increased in size through evolution, they have become more folded in the process (Hoffman, 1989).

Folding of the cortex is a mechanical process that is intrinsic to the brain (Welker, 1990). Over the last 100 years, scientists have proposed potential mechanisms that underlie folding (Hoffman, 1989; Welker 1990). In 1929, Bok noted that a growth component is essential to any model of cortical folding. Le Gros Clark (1945) postulated that compressive stresses from the skull develop during cortical expansion that causes sulci to develop parallel to non-compressible structures. However, Barron (1950) showed that normal patterns of convolutions still develop when subcortical structures are removed *in utero* before folding begins. Using a projection microscope, Smart and McSherry (1986b) drew radial tissue lines in ferret cortices during development. They observed that the total length of the cortical columns and the depth of the cortical layers tend to be conserved during the formation of gyri. Also, the floors of the sulci move very little in relation to each other and to the inner boundary of the cortex, suggesting that outward folds are produced by an increased expansion of the regions in between sulcal floors.

The two most widely accepted hypotheses of the development of cortical convolutions are buckling induced by differential growth (Richmann et al., 1975) and the tension-based theory of morphogenesis (Van Essen, 1997).

The theory of mechanical buckling suggests that differential growth between the cortex and subcortical structures causes the cortex to buckle (Richmann et al., 1975). The authors apply a mathematical model of buckling on an elastic base and show that the

6

model mimics normal folding and the abnormal folding diseases lissencephaly and microgyria. Microgyric brains are marked by a high frequency of low amplitude folds in the cortex. In contrast, lissencephalic brains are very smooth and do not contain secondary or tertiary folds. The authors found good agreement between their model and observed folding patterns in both normal and abnormal brains. A major concern with this model is that it does not account for the consistency between individuals in the folding pattern of primary and secondary folds. Also, folding does not occur suddenly, so perfect buckling is highly unlikely. Finally, the difference assumed by Richmann et al. (1975) in the elastic modulus between the cortical layers and the elastic foundation (subcortical structures) is too large at a factor of 10.

The theory of tension-based morphogenesis hypothesizes that mechanical tension, working against internally generated hydrostatic pressure, is a major driving force for many aspects of morphogenesis of the central nervous system (Van Essen, 1997). Tension is generated by axons, dendrites and glial processes; hydrostatic pressure comes from the fluid-filled ventricles. According to this model, cortical regions that are highly connected are pulled towards one another, creating outward folds and reducing the axonal distance. Axons have been shown to sustain tension both *in vivo* (Xu et al., 2009) and *in vitro* (Chada et al., 1997; Lamoureax et al., 1989). Tension-based morphogenesis inherently provides a description of why primary and secondary folds are consistent in location between individuals. While the tension-based theory provides a plausible method for describing how convolutions develop in the central nervous system, direct testing is necessary for validation.

Recently, Xu et al. (2010) performed such a study. Small incisions were made in excised brain tissue of ferrets, obtained at a series of times during cortical folding. The idea is that if a cut springs open, then tension is present perpendicular to the cut, and if the cut closes, then the tissue is in compression. The authors argue that if tension is indeed a major driving force in drawing specific regions of the cortex together, then cuts made parallel to sides of a gyrus should open. However, the parallel cuts remained closed but cuts perpendicular to the gyral axis actually opened. The authors also used a finite element model to show that differential growth along with remodeling of the subplate could cause convolutions to develop, while generating stress fields similar those observed in dissection experiments. To explain the consistent locations of gyri and sulci, additional factors must be considered. Subcortical structures could provide geometric imperfections like local bumps or dimples that dictate the ensuing locations of primary and secondary folds. Alternatively, differences in timing of local growth spurts could lead to consistently located gyri and sulci.

1.3 Cortical development in ferrets

The ferret provides a good model for the study of cortical development because it is the smallest laboratory animal with a folded cortex, cortical folding takes place postnatally and ferrets are small enough to fit into small animal imaging scanners. Similar to humans, the cortex in ferrets forms from the radial migration of cells from a proliferative zone, which lines the lumen of the neural tube. Neurons populate what will become the cortex in a columnar fashion, with the oldest born neurons occupying the deepest layers and the newest born neurons migrating to the most superficial layers

(McSherry and Smart, 1986). The majority of neurons in the ferret have migrated to the cortex by the end of the first postnatal week (Noctor et al., 1997). Neurogenesis within the developing ferret cortex occurs in a 2-D gradient in the rostral/causdal and lateral/dorsal directions (McSherry 1984; McSherry and Smart 1986; Jackson et al., 1989; Noctor et al., 1997). Neurons that form the rostrolateral cortex migrate to the cortical plate first while neurons that form the occipital pole migrate last, with a neurogenetic gradient existing in the intervening cortex (McSherry, 1984; McSherry and Smart, 1986). The rostrocaudal gradient has been further confirmed in ferrets using birthdating experiments (Jackson et al. 1989; Noctor et al. 1997), and histochemical and electron micrographic characterization of synaptic development (Voigt et al. 1993).

The boundaries of primary visual (Manger et al. 2002, 2004), auditory (Bizley et al. 2005), and somatosensory areas (Leclerc et al. 1993; Rice et al. 1993; McLaughlin et al. 1998) of the ferret isocortex have been described relative to anatomical landmarks. The approximate locations of these boundaries are shown mapped onto a ferret brain in Fig. 1.5. The boundaries of specific nonprimary (Leclerc et al. 1993; Rice et al. 1993; Manger et al. 2002, 2004) and multimodal areas (Ramsay and Meredith, 2004; Bizley et al. 2007) have also been described. Experiments using cell labeling show that the generation of neurons for the somatosensory area of the cortex occurs earlier in development than for the visual cortex. Neurons that eventually populate the somatosensory cortex are born at up to P2 (Noctor et al., 1997), while for the visual cortex neurons are born up until P14 (Jackson et al., 1989). The rostrocaudal and lateral-dorsal development gradient is seen in both the somatosensory and visual areas as well. In the visual cortex, by P21 most all of the cells generated postnatally are found in their

adult positions (Jackson et al., 1989), which corresponds with the approximate time when ferret kits open their eyes.

Macroscopically the ferret cortex appears smooth at birth, with all folds present in an early form within the first seven days of life (Smart and McSherry, 1986a). Fig. 1.3 shows photographs of fixed ferret brains at ages postnatal day 4 (P4) through adulthood. The largest change in suclal and gyral formation occurs at the same time as neuronal differentiation and maturation (Neal et al., 2006). Also at the same time, the brain grows considerably in size. Even after folding has occurred, the brain continues to change in shape. In the fifth and sixth week of life, the sulcal spaces become more narrowed while the rostral portion extends forward and the gyri become less rounded (Barnette et al., 2009). By the end of the fifth week, the brain is roughly the same size as the adult brain. The naming convention used in this dissertation to refer the specific names of the sulci and gyri of the ferret brain is based off of those identified in Smart and McSherry (Smart and McSherry, 1986a) and Kroenke et al. (Kroenke et al., 2009).

MRI provides additional insight into the development of the ferret brain. On T_2-weighted images of ferret brains early in development, the T_2 of CSF is longer than WM which is long than GM. Because of this GM appears gray, CSF appears bright and WM appears light gray (Fig. 1.4a). As myelination occurs during development, the T_2 of WM shortens. At around 5-6 weeks postnatal age, the T_2 of WM is approximately equivalent to that of GM (Fig. 1.4b) (Barnette et al., 2009). In the adult brain the T2 of WM is shorter than that of GM (Fig. 1.4c).

Early in development, FA in the isocortex is high due to the columnar organization of neurons (Kroenke et al., 2009). Cortical FA begins to decrease after the

completion of neuronal migration as neuronal differentiation and synaptogenesis occur (Kroenke et al., 2009). The rate of decrease in cortical FA varies regionally and is strongly influenced by whether the region a primary (e.g., motor and sensory areas) or non-primary area, architectonic differences between allocortex and isocortex and a regional pattern in the age of neurons that results from the transverse neurogenetic gradient (Kroenke et al., 2009). These regional patterns are similar to those seen in the rat (Huang et al., 2008), baboon (Kroenke et al., 2007) and human (deIpolyi et al., 2005), and correlate well with histological studies in the ferret (McSherry and Smart, 1986; Noctor et al., 1997; Tarui et al., 2005).

The ferret is a good model to characterize the kinematics of the brain during development. Acquiring longitudinal sets of images from the same subject provides snapshots of the brain as it grows and folds over time. Different processes, such as segmentation and registration, must be applied in order to obtain useful information from the acquired images. These image-processing techniques are reviewed in the next section.

1.4 Image processing

Local changes in shape between two acquired images are detected by determining a point-based correspondence. Registration is the process of determining a correspondence between multiple images, surfaces or curves. Approaches for registration in the brain can be separated into three major categories: image-based registration methods (e.g. Beg et al., 2005; Christensen et al., 1996; Christensen et al., 1994; Christensen and Johnson, 2001; Johnson and Chirstensen, 2002; Joshi and Miller, 2000; Miller et al., 1993; Shen and Davatzikos, 2002; Shen and Davatzikos, 2004; Yanovsky et

al., 2008) and reviews (Gholipour et al., 2007; Holden 2008; Klein et al., 2009); surface-based registration methods (e.g. Fischl et al., 1999a,b; Glaunes et al., 2004; Qiu et al., 2009; Litke et al., 2005; Shi et al., 2007; Oguz et al., 2008; Terzopoulos et al., 1987; Thompson and Toga, 1996; Vaillant and Glaunes, 2005; Van Essen et al., 2001; Xue et al., 2007; Yeo et al., 2008) and reviews (Audette et al., 2000; Woods, 2003); and combined volumetric and surface-based methods (Joshi et al., 2009; Liu et al., 2004; Postelinicu et al., 2009).

Existing registration methods employ a number of different numerical techniques to solve for an optimal registration. The gradient descent method is a numerical method that converges towards a local minimum or maximum of a function (Snyman, 2005). For registration, it is often used to find the minimum solution of an energy or cost function. The finite difference method and the finite element method are numerical approaches to estimate the solution to partial differential equations. The finite difference method uses difference equations to estimate derivatives (Morton, 2005). The finite element method divides the object into a set of elements and defines piece-wise linear basis functions on each element (Szabó and Babuška, 1991). Derivatives are calculated analytically from the basis functions.

An advantage to volume-based registration over surface-based is that the image volume provides a set coordinate system that is easy to navigate and straightforward to calculate derivatives. The registration of surfaces presents a number of challenges including the estimation of derivatives and how to constrain coordinate displacements to remain on the surface. One way to simplify the problem is to parameterize the anatomical surface to a simple well-defined shape such as a sphere. All mathematical operations are

performed in the spherical space and mapped back to the anatomical space after completion of the registration. Distortions are introduced by mapping a convoluted surface to a spherical surface. Care must be taken to ensure that these distortions are accounted for during registration. Another approach is to use an implicit representation of the surfaces where both the source and target surfaces are represented as the zero level set of functions (Osher and Sethian, 1988). Numerical techniques are then implemented in the same manner as in volume-based approaches. Below, some of the registration approaches that have been implemented by different groups are described in more detail.

The large deformation diffeomorphic metric mapping (LDDMM) approach (Beg et al., 2005) is an image-based registration algorithm that computes the metric distance between an image and a target image. A vector field is indentified that minimizes a cost function that is a function of smoothness of the vector field and how well the image intensities match between the images. The energy functional is optimized by implementing a steepest descent scheme. The LDDMM algorithm allows for the quantification of shape differences due to different neurological disorders in a large deformation environment.

Vaillant and Glaunès (2005) developed a surface-based registration approach that determines a one-to-one mapping in a large deformation setting by solving an optimization problem similar to the approach by Beg et al. (2005). Surface matching is implemented using a steepest descent algorithm. A major advantage to this approach is that all calculations are performed on the actual anatomical surfaces, so no intermediate space (i.e., a sphere) is required. We sent two surfaces (a cortical surface of a ferret at 14 days and 21 days) to their lab and they applied their algorithm to the surfaces.

Unfortunately, they were unable to register the surfaces. This algorithm depends on the physical location of the two surfaces in space and their orientation with respect to one another. A large amount of growth occurs between the two time points, which could contribute to difficulties in determining a correspondence.

The registration algorithm in CARET software (Van Essen et al., 2001) is a surface-based approach that determines a one-to-one correspondence between surfaces in a spherical space. Landmark points are identified and rigidly aligned with one another. Additionally, distortions between the spherical surfaces are reduced by matching the area of each surface face. The approach in CARET does a good job at aligning features between surfaces but introduces distortions due to performing the registration in a spherical space.

The registration algorithm in FreeSurfer (Fischl et al., 1999) is also a surface-based approach that computes a one-to-one correspondence in a spherical space. Instead of using landmark points, the registration is driven by the alignment of a measure called convexity, which is similar to mean curvature but is less susceptible to noise. The algorithm minimizes an energy functional that is a function of how well the folding patterns match and how well the local areas and distances are preserved. While the algorithm is intended to avoid distortions caused by parameterizing the cortical surface to a sphere, these induced distortions are not used during the registration.

Xue et al. (2007) developed a surface-based registration approach for the longitudinal registration of cortical surfaces in developing human infants. A spherical parameterization is not required, but cortical surfaces are inflated. An initial image-based affine registration is performed between the T_2-weighted images, and the transformation

is applied to the surfaces. The more mature cortical surface is smoothed to decrease the complexity of the folding, which makes the mature cortex qualitatively more similar to the less mature cortex and reduces the likelihood that the solution will converge to a local minimum. The surfaces are registered by moving control points so that a surface similarity function is minimized. The downside to this approach is that secondary and tertiary sulci are smoothed out and are therefore not accounted for in the registration.

Joshi et al. (2009) have developed a framework for the simultaneous registration of a cortical surface and image volume. Combined surface- and volume-based approaches accurately align both the cortical folding patterns as well as subcortical structures. For this algorithm, a cost function is minimized based on image matching and surface curvature matching terms. Smoothness of the deformation is enforced by penalizing distortions though applying an elastic energy term. The cost function is minimized using the gradient descent method.

Postelnicu et al. (2009) developed a similar approach based on a hybrid volume- and surface-based registration. In their method, a surface-based registration is performed and used to set the initial conditions for the volume-based registration. The surface-based registration is the same as the approach described in Fischl et al. (1999). Using the surface-based registration to set the initial conditions for the volume-based registration does a better job aligning the cortex, which volume-based approaches generally have a difficult time doing. The initial conditions are extended into the volume by applying the finite element method is applied to solve equations based on the linear theory of solid mechanics. Finally, an intensity-based registration is performed through the minimization

of an energy term based on smoothness, similarity between intensities, ensure invertibility and minimize distortions.

Shi et al. (2007) take advantage of implicit descriptions of surfaces, which allows standard numerical schemes to be implemented in 3D. Beginning from an initial map, they iteratively solve a PDE on the reference surface. The optimal registration is defined as the minimization of an energy term, which is the sum of a harmonic (smoothness) and data (geometric features) term. Landmarks are also used to aid in the registration. However, when this method was applied to analyze uniform growth of a ferret cortical surface, it was found to introduce artificial distortions between the surfaces, apparently because of discretization error. The implicit surface method, because it relies on discrete voxel size, may not be optimal for longitudinal registration for highly convoluted surfaces like the mammalian brain.

Litke et al. (2005) map open surfaces in 3D to the plane, which simplifies the computations considerably. A PDE that accounts for nonlinear large deformations is solved using the finite element method and a multiresolution approach. The optimal registration is defined as the minimum of an energy function, which is the sum of regularization (smoothness), matching (geometric features) and bending energies. The specific approach of Litke et al. (2005) can only be applied to open surfaces, which would involve making cuts in or only looking at part of the cortical surface. The authors do not expect a one-to-one correspondence to exist between the surfaces, while our approach requires it. We deliberately do not include a penalty on bending energy, since the distortions we wish to quantify during cortical folding include large bending deformations.

Image-based approaches generally are able to accurately align subcortical structures but often have difficulties in aligning the cortex due to the high degree of variability. Furthermore, image volumes contain a large number of voxels, which is computationally intensive. Surface-based registration approaches do a better job at handling the variability of folding patterns in the brain, but require the cortex to be segmented from anatomical images in order to generate the surface mesh. Combined volumetric and surface-based approaches often use surface registration to obtain accurate initial conditions near the cortex to aid the volume-based method, which provides a better registration of the entire brain than just using a volume-based approach. The goal of this study is to quantify growth in the cortex of the ferret brain during development using MRI. As we are interested in how the shape of the cortex changes over time, a surface-based approach is warranted.

In choosing a surface-based approach, it is important that the topology of the surfaces be maintained during registration (e.g., if a point D resides inside the face formed by points A, B and C before registration, then D should always remain inside that face). Also, distortions introduced by mapping an anatomical surface to a parameterized shape, such as a sphere, must be accounted for during registration. A disadvantage of using a surface-based approach is that the acquired images must be segmented in order to generate surfaces. Manual segmentation of MRI volumes is very time consuming and is subject to human error. A number of algorithms have been created that automate the segmentation of an image (e.g. Dale et al., 1999; Van Essen et al., 2001; Zhang et al., 2001). Due to changes in contrast between grey matter and white matter and changes in shape of the cortex these approaches do not work well when applied to preterm and term

infants. Recently, algorithms have been developed that are specifically designed for images of preterm and term infants (Xue et al., 2007; Weisenfeld and Warfield, 2009; Hill et al., 2010).

1.5 Overview of dissertation

The specific aims of this dissertation are:

1. Acquire magnetic resonance images of ferret brains at different postnatal ages.

2. Create cortical surface representations from images.

3. Develop analysis tools to estimate Lagrangian strain and curvature.

4. Develop and validate an intra-subject surface registration algorithm.

5. Apply analysis tools to cortical surfaces to measure spatial variations in growth and deformation during development.

In Chapter II I describe and implement a novel method to calculate Lagrangian strain between two surfaces that can be multi-valued (i.e., for each x and y, more than one z can exist). Using a least-squares approach, second-order polynomial functions were fitted to local patches of the surface. Derivatives are calculated analytically. The method was validated by applying it to a set of test cases with known distributions of surface strain.

A point-to-point correspondence is required to estimate strain between surfaces. Surface registration is a technique that provides such a correspondence between arbitrary surfaces. Chapters III presents a framework for the registration of two surfaces. For our approach the optimal registration is defined by the minimization of an energy function, which is achieved by solving a partial differential equation of motion on a parameterized

surface. The energy function depends on local distortions and on differences in surface matching terms between the surfaces. The solution is obtained using the finite element method. While the solution is obtained in a parameterized space, distortions between the anatomical surfaces are accounted for in the formulation. Surface matching terms are based on mean curvature and geodesic distance from anatomical landmarks (e.g., the medial wall). Chapter IV describes the validation of this approach by application to a series of artificially generated test cases for which the actual solution is known.

In Chapter V I apply the methods described in Chapters II-IV to the characterization of cortical folding. Using MRI, images of the brain were acquired from three ferret kits at one-week intervals for four weeks. Segmentation volumes were generated manually and used to create triangular mesh models of the cortical surface. Global measures of shape were calculated for each surface. From one of the kits, the surface at P14 was registered to P21, and P21 was registered to P28. Local growth was then calculated between the registered surfaces. Finally, in Chapter VI I summarize the results obtained by this approach, discuss its limitations and suggest some directions for future work.

(a) **(b)**
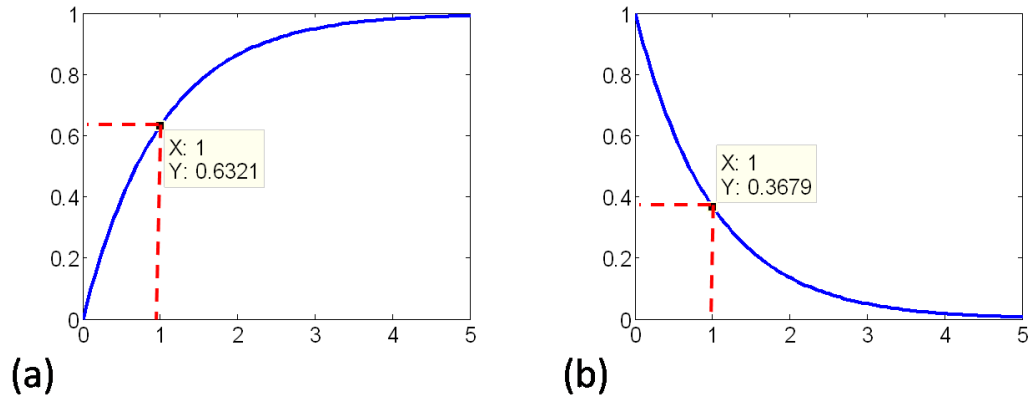
Figure 1.1: MRI principles. (a) Recovery of the net magnetization in the direction of $\mathbf{B_0}$

$\left(M = M_0\left(1 - e^{\frac{-t}{T_1}}\right)\right)$. T_1 is defined as the time it takes for the difference between the net

magnetization vector along $\mathbf{B_0}$ and its equilibrium value to be reduced by a factor of e

(Hornak, 1996). (b) Decay of the net magnetization in the xy-plane $\left(M = M_0 e^{\frac{-t}{T_2}}\right)$. T_2 is

defined as the time required to reduce the transverse magnetization by a factor of e

(Hornak, 1996).
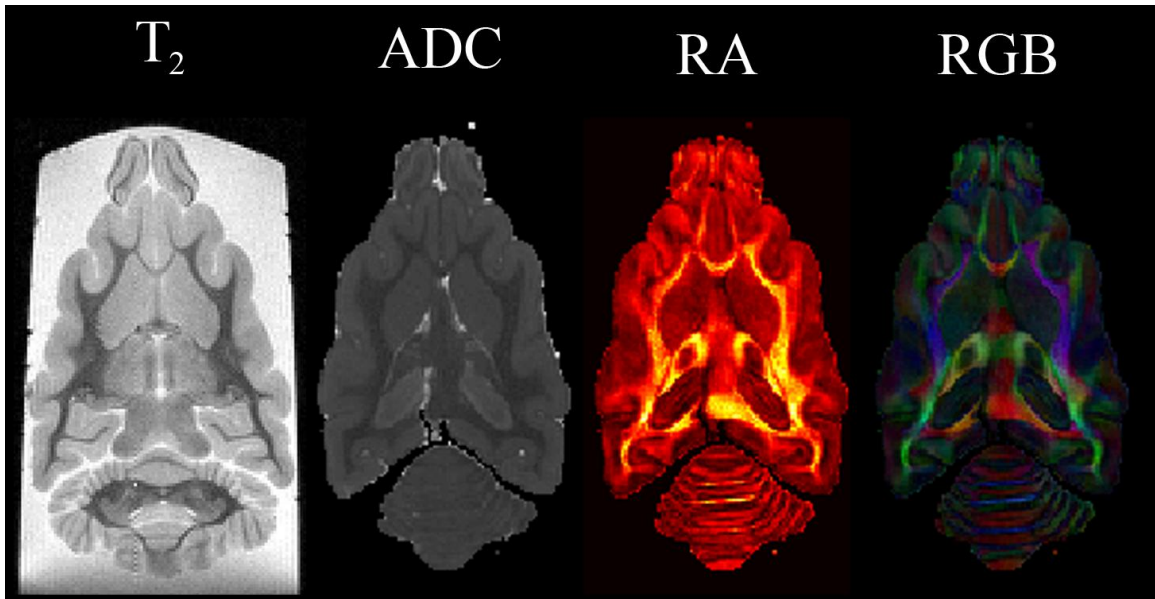
Figure 1.2 : MRI of an adult ferret brain (perfusion-fixed, in Krytox[TM]): T_2-weighted image provides anatomical detail. WM has a shorter T_2 and appears darker than GM. DTI images provide structural information. ADC is the average of the eigenvalues of the diffusion tensor. RA is the ratio of the standard deviation to the average of the eigenvalues. RGB plots assign a color to the primarary eigenvector at each volexl: red is right-left, green is posterior/anterior, blue is inferior/superior.

Figure 1.3: Photograph of ex vivo ferret brains ranging from P4 to Adult. This figure is reprinted with permission of author (Barnette et al., 2009).
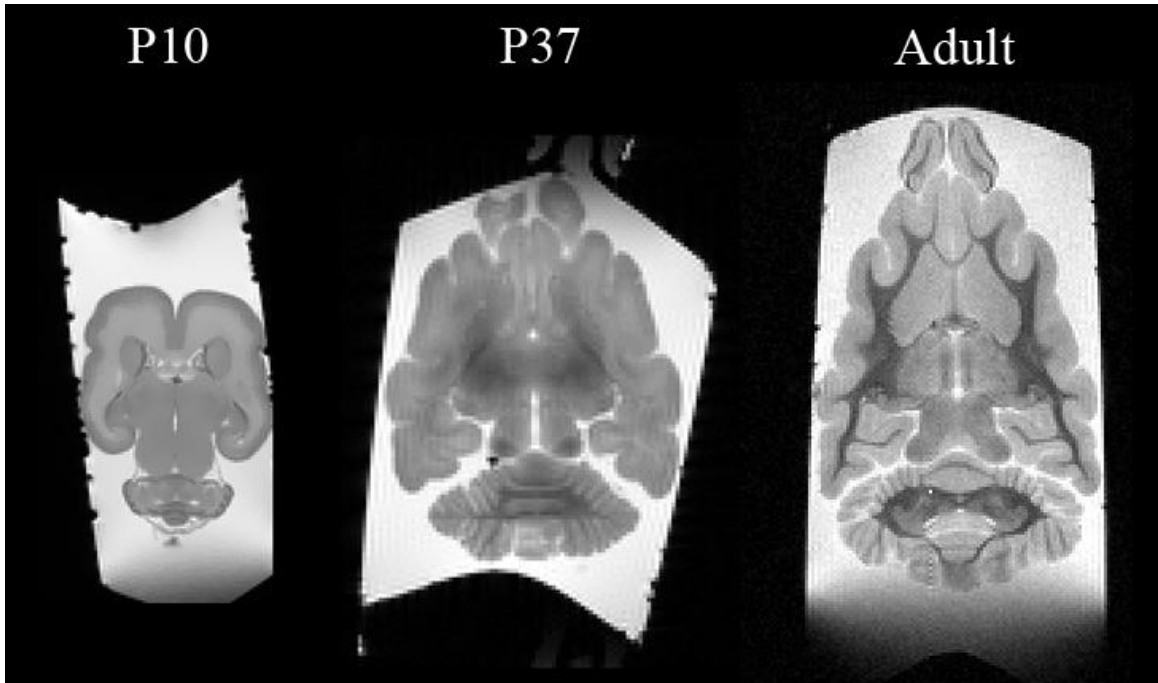
Figure 1.4: T$_2$-weighted images of ferret brains acquired *ex vivo* that shows the change in contrast in the WM that occurs during development due to myelination of the axons. As the T$_2$ of WM decreases during development the WM darkens.
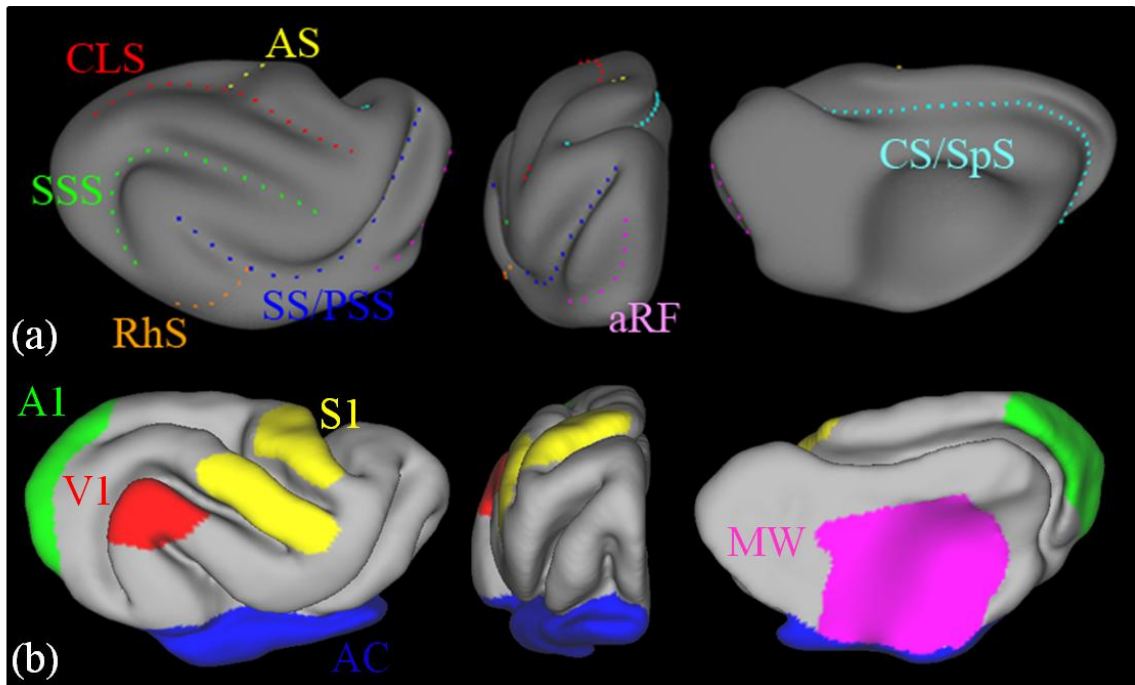
Figure 1.5: (a) Sulci in the ferret brain: coronolateral sulcus (CLS, red); sylvian sulcus/presylvian sulcus (SS/PSS, dark blue); suprasylvian sulcus (SSS, green); cruciate sulcus/splenial sulcus (CS/SpS, cyan); anterior rhinal fissure (aRF, pink); rhinal sulcus (RhS, orange); ansate sulcus (AS, yellow). (b) Primary cortical areas: auditory (A1, red), visual (V1, green) and somatosensory (S1, yellow). Boundaries are approximate. In addition, the allocortex (AC, blue) and the medial wall (MW, purple) are also shown.
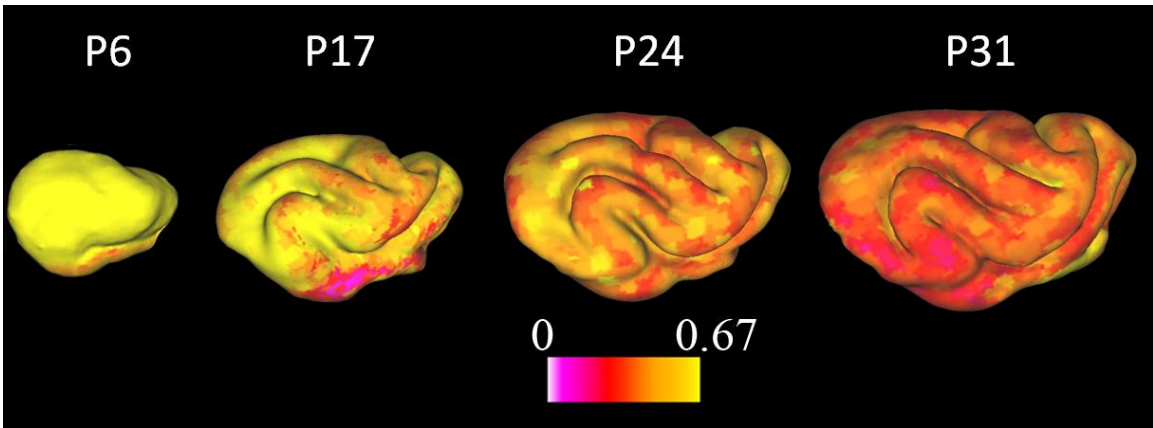
Figure 1.6: Relative anisotropy projected onto cortical surface representations at P6, P17, P24 and P31.

Chapter II. Calculation of surface strain on closed surfaces[1]

In this chapter, we present a method to estimate Lagrangian strain on surfaces that are multi-valued (i.e. for any x and y there may be more than one point on the surface). This capability is important for characterizing growth and morphogenesis of three-dimensional (3-D) bodies, such as embryos or developing organs.

1. Introduction

In the embryo, many morphogenetic processes involve complex 3-D deformations of cell sheets, or epithelia (Davies, 2005; Gilbert, 2003). In studies of morphomechanics, it is useful to quantify these deformations, and researchers have measured strain distributions in epithelia during gastrulation (Zamir et al., 2005a) and early heart development (Ramasubramanian et al., 2006). These analyses, however, generally have been restricted to single-valued surfaces of relatively modest curvature. This limitation is especially problematic when an epithelium folds, a common occurrence during embryogenesis, or when attempting to quantify growth of the cortex during development of the brain.

Here, we present a technique for measuring surface strain that can accommodate the deformation of multiple-valued surfaces. Deformation gradients, strains, and other kinematic quantities are computed using straightforward matrix algebra. After the basic theory is presented, the accuracy of the method is examined for some sample problems with exact solutions. Our method is applicable to a wide range of morphogenetic problems, as well as to functional studies, e.g., strain measurements in the beating heart.

---

[1] The material in this chapter is published in the Journal of Biomechanical Engineering (Filas, Knutsen et al., 2008).

Finally, our method is applied to surfaces created from cortical segmentation volumes of MRIs of ferret brains acquired at different stages of development. The application to two sets of ferret brains is presented in Chapter IV.

2. Methods

2.1 Theoretical and Numerical Methods

2.1.1 Images, Surfaces and Markers

Deformation is measured by following the displacements of a set of fiducial points (markers) attached to a surface. The surface and the marker locations are typically derived from image volumes acquired, for example, by magnetic resonance imaging (MRI), computed tomography (CT), or optical coherence tomography (OCT). Surfaces are created from segmented image volumes via previously-described algorithms (e.g., CARET (Van Essen et al., 2001)) or standard software (Matlab, The Mathworks Inc., Natick, MA). The resulting surface created by these algorithms consists of a set of triangular faces, each defined by the (global) coordinates of its three vertices and by a normal unit vector.

To characterize deformation kinematics, accurate measurements of displacements of points on the surface are needed. In the absence of natural landmarks that can be tracked over time, markers, such as opaque or reflective beads, are attached to the physical surface, so that they move with the material. The location of each marker is tracked over the duration of the imaging study. Typically, the number of markers is much

smaller than the number of vertices that represent the surface. Hence, the distance between vertices is usually small compared to the marker spacing.

2.1.2 Coordinate Systems

Surfaces are originally described with respect to a global Cartesian coordinate system. The base unit vectors of the global system are aligned with the axes of the 3-D image volume (typically a stack of 2-D images).

Local Cartesian coordinate systems are defined in order to analyze the local deformation of surfaces near specific points. Each bead, or other fiducial marker, is used in turn as the origin of such a local system. A local, approximately normal, unit vector, e_3, is taken to be the average of the normals of the five nearest faces on the surface. An orthogonal, approximately tangent unit vector, e_1, is defined by setting one component to zero and enforcing $e_3 \cdot e_1 = 0$ and $|e_1| = 0$. Finally, another orthogonal, approximately tangent unit vector, e_2, is obtained directly using $e_2 = e_3 \times e_1$ (Fig. 2.1).

2.1.3 Analysis of Deformation

The analysis is based on the general nonlinear membrane theory of shells (Novozhilov, 1970, Axelrad, 1987). Consider the deformation of the reference surface S into the current surface s (Fig. 2.1). For each locally analyzed region, the positions of points on these surfaces are described relative to the local Cartesian axes defined above. Relative to these axes, a point with material coordinates X^i on S moves to the spatial

coordinates x^i on s.[2] Tracking individual markers allows us to write the spatial coordinates for a finite number of points in terms of material coordinates, i.e., $x^i = x^i(X^i)$. Next, we assume that the local region of surface S to be analyzed can be described by the relationship $X^3 = X^3(X^1, X^2)$, where X^3 is a single-valued function of the local coordinates X^1 and X^2. Hence, we can write

$$x^i = x^i(X^1, X^2, X^3(X^1, X^2)) = x^i(X^1, X^2).$$

The position vectors to a point on S and its deformed image on s are, respectively,

$$\begin{aligned} \mathbf{R} &= X^\alpha \mathbf{e}_\alpha + X^3(X^\alpha)\mathbf{e}_3 \\ \mathbf{r} &= x^\beta(X^\alpha)\mathbf{e}_\alpha + x^3(X^\alpha)\mathbf{e}_3. \end{aligned} \qquad \text{II.1}$$

Covariant base vectors in S and convected base vectors in s are given by the respective relations (Taber 2004)

$$\mathbf{G}_\alpha = \frac{\partial \mathbf{R}}{\partial X^\alpha} = \mathbf{R}_{,\alpha} \quad \mathbf{g}_\alpha = \frac{\partial \mathbf{r}}{\partial X^\alpha} = \mathbf{r}_{,\alpha}. \qquad \text{II.2}$$

Note that these base vectors are tangent to the surfaces, but are generally not orthogonal or unit vectors. Substituting Eqs. II.1 into II.2 yields

$$\mathbf{G}_\alpha = \mathbf{e}_\alpha + X^3_{,\alpha}\mathbf{e}_3 \quad \mathbf{g}_\beta = x^\beta_{,\alpha}\mathbf{e}_\beta + x^3_{,\alpha}\mathbf{e}_3. \qquad \text{II.3}$$

In addition, *unit* vectors normal to S and s, respectively, are given by

$$\mathbf{G}_3 = \frac{\mathbf{G}_1 \times \mathbf{G}_2}{|\mathbf{G}_1 \times \mathbf{G}_2|} \quad \mathbf{g}_3 = \frac{\mathbf{g}_1 \times \mathbf{g}_2}{|\mathbf{g}_1 \times \mathbf{g}_2|}. \qquad \text{II.4}$$

Finally, contravariant base vectors, $\mathbf{G}^i$ and $\mathbf{g}^i$, are defined by the relations

$$\mathbf{G}_i \cdot \mathbf{G}^j = \mathbf{g}_i \cdot \mathbf{g}^j = \delta_i^j. \qquad \text{II.5}$$

---

[2] In this section, Latin indices take the values 1,2,3 and Greek indices take the values 1,2. The usual summation convention on repeated indices is implied.

where δ_i^j is the Kronecker delta.

In the membrane theory of shells, base vectors typically are written in terms of coordinates that lie entirely within the surface. Here, however, the surface is considered explicitly embedded in 3-D space, with the base vectors written in terms of coordinates that are (approximately) tangent and normal to the surface at only one point. As shown next, this approach makes the analysis relatively simple to program in Matlab using matrix algebra. The only explicit expressions needed in this analysis are those for the covariant base vectors of Eqs. II.3 and II.4. (It is important to note that this method requires the surface normals $\boldsymbol{G_3}$ and $\boldsymbol{g_3}$ to be defined as unit vectors, to prevent erroneous transverse deformation from entering the calculations.)

Equations II.3 and II.4 provide base vectors in the component forms

$$
\begin{aligned}
\boldsymbol{G}_i &= G_{ij}\boldsymbol{e}^j & \boldsymbol{g}_i &= g_{ij}\boldsymbol{e}^j \\
\boldsymbol{G}_i &= G^{ij}\boldsymbol{e}_j & \boldsymbol{g}^i &= g^{ij}\boldsymbol{e}_j
\end{aligned}
\tag{II.6}
$$

Writing all vectors and tensors in terms of components relative to the Cartesian unit basis $\boldsymbol{e}^i = \boldsymbol{e}_i$ allows us to use matrix algebra from here on. With bracketed quantities denoting 3x3 matrices, the components of the base vectors are represented by

$$
\begin{aligned}
[G] &= [G_{ij}] & [g] &= [g_{ij}] \\
[G^*] &= [G^{ij}] & [g^*] &= [g^{ij}].
\end{aligned}
\tag{II.7}
$$

With the components of $[G]$ and $[g]$ known, Eq. II.5 yields matrix equations for $[G^*]$ and $[g^*]$ in the form

$$
\begin{aligned}
[G][G^*]^T = [\mathrm{I}] &\to [G^*] = [G]^{-T} \\
[g][g^*]^T = [\mathrm{I}] &\to [g^*] = [g]^{-T}.
\end{aligned}
\tag{II.8}
$$

where $[I]$ is the identity matrix and T denotes transpose.

The deformation gradient tensor is given by (Taber, 2004) as

$$F = g_j G^i \rightarrow [F] = [g]^T [G^*] .$$ II.9

With F now known, it is straightforward to compute any deformation measure of interest. For example, the components of the right and left Cauchy-Green deformation tensors, respectively, are given by

$$[C] = [F]^T [F], \quad [B] = [F][F]^T .$$ II.10

which provide the components of the Lagrangian and Eulerian strain tensors (Taber, 2004)

$$[E] = \frac{1}{2}([C] - [I]), \quad [e] = \frac{1}{2}([I] - [B]^{-1}) .$$ II.11

Here, we again emphasize that all matrix components are defined relative to the local Cartesian axes.

2.1.4 Stretch Ratios and Strains in Specific Directions

Because the local Cartesian system does not follow the surface at every point, the components of the above tensors generally have no direct physical interpretation. However, meaningful quantities can be extracted relatively easily. For example, if N_1 and N_2 are *orthogonal* unit vectors tangent to S, then physical Lagrangian strain components relative to these directions are given by

$$E_{\alpha\beta} = \{N\}_\alpha^T [E] \{N\}_\beta ,$$ II.12

where curly braces indicate vectors. Similarly, if n_1 and n_2 are orthogonal unit vectors tangent to s, then physical Eulerian strain components are given by

$$e_{\alpha\beta} = \{n\}_\alpha^T [e] \{n\}_\beta .$$
II.13

Alternatively, stretch ratios in the direction $\mathbf{N}$ on S or $\mathbf{n}$ on s can be computed from the respective relations (Taber, 2004)

$$\Lambda^2_{(N)} = \{N\}^T [C] \{N\}, \quad \lambda^{-2}_{(n)} = \{n\}^T [B]^{-1} \{n\} .$$
II.14

Corresponding measures of change in angle between undeformed line elements, $\mathbf{N}_\alpha$ on S, and deformed line elements, $\mathbf{n}_\alpha$ on s, are given by the shear $\Gamma_{(N_\alpha N_\beta)} = \gamma_{(n_\alpha n_\beta)} = (\mathbf{n}_\alpha \cdot \mathbf{n}_\beta) - (\mathbf{N}_\alpha \cdot \mathbf{N}_\beta)$ (Taber, 2004), calculated from either of the relations

$$\Gamma_{(N_\alpha N_\beta)} = \{N\}^T \left(\frac{[C]}{\Lambda_{(N_\alpha)} \Lambda_{(N_\beta)}} - [I] \right) \{N\}_\beta \quad \gamma = \{n\}^T \left([I] - \frac{[B]^{-1}}{\lambda_{(n_\alpha)} \lambda_{(n_\beta)}} \right) \{n\}_\beta .$$
II.15

For the heart and brain problems, strains were computed relative to unit vectors along the local directions of maximum and minimum curvature of s, corresponding approximately to local circumferential and longitudinal directions in the heart or brain tube at any time during development. The components of the curvature tensor $\boldsymbol{\kappa}$ are provided by

$$\kappa_{\alpha\beta} = \frac{\partial^2 x^3}{\partial x^\alpha \partial x^\beta} .$$
II.16

With the 2x2 curvature matrix defined by $[\kappa] = [\kappa_{\alpha\beta}]$, the eigenvalue problem

$$([\kappa] - k[I])\{v\} = 0$$
II.17

yields the principal curvatures k_α and directions $\{v\}_\alpha$. Relative to these orthogonal directions, Eqs. II.13-14 then provide the corresponding Eulerian strains and stretch ratios, respectively.

Principal stretch ratios $\Lambda_i = \lambda_i$ are obtained by solving either of the eigenvalue problems

$$([C] - \Lambda^2[I])\{M\} = 0 \quad ([B] - \lambda^2[I])\{m\} = 0 \ , \qquad \text{II.18}$$

where $\{M\}_i$ and $\{m\}_i$ are eigenvectors in S and s, respectively. Because the normal vectors do not change length, these relations yield unity for one principal stretch ratio, with the corresponding eigenvector normal to the surface.

2.1.5 Piecewise Fitting of Local Surface Functions

The above strain analysis is valid for reference surfaces that can be described by a single-valued function $X^3(X^1, X^2)$. During morphogenesis, however, surfaces often fold, leading to multiple values of X^3 for each X^1, X^2 combination in a global coordinate system. Hence, we compute strains in a piecewise manner, where each region of the surface is single-valued relative to its own local coordinate system (see Coordinate Systems section above). Relative to this system, the analysis requires expressions for the functions $X^3(X^1, X^2)$ and $x^i(X^1, X^2)$ for $i = 1,2,3$. It is important to note that the use of convected base vectors in the strain analysis means that only the reference surface need be single-valued.

Local surfaces were fit to the finite set of points to give local approximations for $X^3(X^1, X^2)$ and $x^i(X^1, X^2)$. Each marker in turn was assigned to be the origin of a local Cartesian coordinate system. The absolute distances of all of the other markers from the local origin were calculated, and surfaces were fitted using markers within a user-

33

specified radius, r. All markers outside that distance (r) were ignored.. The required

functions were the second-order polynomial functions of X^1 and X^2 given by

$$f(X^1, X^2) = a_0 + a_1 X^1 + a_2 X^2 + a_3 X^1 X^2 + a_4 (X^1)^2 + a_5 (X^2)^2 \qquad \text{II.19}$$

The coefficients a_i were determined in a least squares sense (using the *mldivide* function

in Matlab). Then, the derivatives of these functions at the local origin were calculated and

inserted into Eq. II.3. This process was repeated for all markers.

2.1.5 Length Scales: Curvature, Wavelength, Marker Spacing, and Fitting Radius

The characteristic length scales of the surface and the marker distribution can be

used to bound the size of the fitting region. Length scales of a curved surface include the

minimum radius of curvature, and the wavelength of undulations. For a quadratic

function (e.g., Eq. II.18) to provide accurate strain estimates using our method, the error

in the derivatives of the fitted surface (compared to the true surface), should be small

(less than 10%). A quadratic fit to a cylindrical surface can be shown by straightforward

calculation to satisfy this criterion if

$$r < 0.43\rho , \qquad \text{II.20}$$

where ρ is the radius of curvature of the cylinder. A quadratic fit to a sinusoidal function

can be shown to satisfy the derivative error criterion if

$$r < 0.071\lambda , \qquad \text{II.21}$$

where λ is the wavelength of the sinusoid. Note that the wavelength of a sinusoid of

amplitude a is related to its minimum radius of curvature by $\lambda = 2\pi\sqrt{\rho a}$, so that for

curves with amplitudes of the order of the radius of curvature, the criteria of Eqs. II.19

and II.20 are numerically similar. These criteria lead to the following guideline: *the radius of the fitting region should be less than half the local radius of curvature of the surface*. Note that the local radius of curvature may be found from high-resolution, image generated surfaces.

On the other hand, the fitting region must include enough markers so that the six free parameters of Eq. II.19 can be estimated accurately. For example, if at least 8 points are desired in almost all fits, an average ± std. deviation of 20±4 points per fitting region is appropriate. If the marker density is μ markers/unit area (an average marker separation of $\sqrt{\frac{1}{\mu}}$ length units), the radius of the fitting region should satisfy

$$r^2 \pi \mu > 20 \ .$$
II.22

In the current work, a fixed radius for all fitting regions was selected based on these criteria. In principle, the fitting region could be varied for different parts of the surface. Note that these criteria are based on the surface geometry, under the assumption that variations in strain occur at similar length scales. Since variations in marker distribution, measurement error, and actual deformation could cause underlying assumptions to be violated, strain estimates were rejected if either of two fitting requirements were not met: (1) A minimum number of markers ($N_{min} = 8$) must be found within the specified fitting radius, r; and (2) the residual error of each fit must be less than a specified fraction (usually 0.3) of the variance of the data. Finally, although the fitting process reduces the effects of random variations, errors in displacement measurements should be considerably smaller than the displacements themselves.

3. Results

The accuracy of the method for estimating surface strain is evaluated for two surfaces of relatively simple geometry undergoing specified deformations.

3.1 Cylindrical Bending of a Sheet

The undeformed surface is a flat sheet defined in a global Cartesian (X, Y, Z) coordinate system by $0 \leq X \leq 1$, $-2.5 \leq Y \leq 2.5$, and $Z = 0$. The sheet is deformed to partially enclose a circular cylinder, as described by (Fig. 2.2)

$$x = X \quad y = \frac{3}{\pi}\cos\left(\frac{\pi}{3}Y\right) \quad z = -\frac{3}{\pi}\sin\left(\frac{\pi}{3}Y\right). \qquad \text{II.23}$$

All strains are zero, as the plane curls without stretch or shear (the ends of the deformed sheet do not meet). With a dense array of markers (1300 markers/unit area, average marker spacing $\sqrt{\frac{1}{\mu}} = 0.028$ units, final radius of curvature $\rho = 0.96$ units, fitting radius $r = 0.10$ units), estimates of first and second principal strains are almost exactly zero when deforming the plane into a cylinder and when unfurling the cylinder back to the original flat surface (Fig. 2.2, Table 2.1). With a less dense random array of markers (40 markers/unit area, $\sqrt{\frac{1}{\mu}} = 0.16$ units), the radius of the fitting region was increased ($\sqrt{\frac{1}{\mu}}$ = 0.4 units) to include similar numbers of markers in each fit. Errors increased slightly, but remained small (Fig. 2.2, Table 2.1).

3.2 Bending and Torsion of a Cylinder

Next we consider a cylinder specified by $R = 0.2,\ 0 \leq \Theta \leq 2\pi$, and $-1 \leq Z \leq 1$ (Fig. 2.3g). After converting from polar to global Cartesian coordinates ($X = Rcos\Theta$, $Y = sin\Theta$), the cylinder was bent and sheared with the deformed surface coordinates given by

$$x' = Xcos(aZ) + Zsin(aZ) \quad y' = Y \quad z' = -Xsin(aZ) + Zcos(aZ), \qquad \text{II.24}$$

where α is a constant that describes the degree of bending (in this case $\alpha = 0.4$). The bent cylinder was subsequently twisted with fully deformed surface coordinates given by

$$x = x'cos(\beta z') + y'sin(\beta z') \quad y = -x'sin(\beta z') + y'cos(\beta z') \quad z = z', \qquad \text{II.25}$$

where β is a constant that describes the degree of twisting (in this case $\beta = 1$).

Results were first obtained with a dense set of regularly spaced markers (2904 markers/unit area, $\sqrt{\dfrac{1}{\mu}} = 0.018$ units, $\rho = 0.2$ units; $r = 0.05$ units). The average absolute errors of the principal strain values were less than 0.01 with this particular fitting radius (Fig. 2.3).

Because strain estimates depend upon the appropriate size of the fitting region relative to key length scales of the data, this parameter (r) was adjusted (Fig. 2.4, Table 2.1). When the size of the fitting region was not large relative to the average distance between randomly scattered markers ($r = 0.05$), small patches arise where insufficient marker density prevents a fit from being performed (Fig. 2.4a). When the size of the fitting region was large relative to the radius of curvature of the cylinder ($r = 0.25$), errors arise due to insufficient fitting accuracy (Fig. 2.4d); the quadratic function is inadequate to capture the local surface curvature.

To illustrate the potential effects of measurement error, random variations in marker coordinate values were applied in both the undeformed and deformed configurations (maximum coordinate variation was of 1% of the corresponding cylinder dimension). Strain estimates were again compared with the exact strain values for different sizes of the fitting region (Fig. 2.5). When the fitting radius is small, error in displacement measurements adds visible "noise" to strain estimates (Fig. 2.5, Table 2.1); increasing the radius of the fitting region (r) reduces the effect of random measurement errors. On the other hand, if the fitting region is large compared to the scale of actual spatial variations (e.g., $r = 0.25$ in this example), error is introduced (Fig. 2.5, Table 2.1).

3.3 Partial expansion of a shell

Using COMSOL Multiphysics v3.4 (COMSOL Inc, Burlington, MA), a surface that is the combination of a sphere and an ellipsoid was created. The spherical surface was assigned a positive coefficient of thermal expansion, and the ellipsoid a negative coefficient. A uniform temperature increase was applied so that the spherical portion of the surface grew by 10% while the ellipsoidal portion of the surface shrank by 10%. The reference and deformed surfaces are shown in Fig. 2.6(a-b). Principal strains were calculated in COMSOL, and are shown in Fig. 2.6(c-d). The surface coordinates were imported into Matlab along with normal vectors of the reference surface. Using our approach, a fitting radius of 0.20 was used to estimate principal strains. Results are shown in Fig. 2.6(e-f).

The results obtained using our approach match well with the results obtained in COMSOL. The first and second principal strains for the spherical portion of the surface are 0.11 and 0.11, which corresponds to uniform expansion by a factor of 1.1.

4. Discussion

A new method has been presented for computing the deformation of folding surfaces with multiple-valued coordinates. Multi-valued surfaces, which arise in 3-D morphogenetic phenomena such as invagination (Ramasubramanian and Taber, 2007) and brain folding (Welker, 1990; Neal et al., 2007), pose difficulties for methods that rely on a single global fitting function (Hashima et al., 1993; Filas et al., 2007). To handle this problem, we analyze locally single-valued patches of the surface.

Our method extends previous work on strain measurements that are based on tracking the motions of tissue labels using non-invasive imaging technologies such as OCT, MRI, or light microscopy. Several studies have used triangles of markers to measure strains due to the heartbeat in the developing heart (Alford and Taber, 2003;Taber et al., 1994; Tobita and Keller, 2000b; Tobita and Keller, 2000a). The spacing between the markers must be closely controlled to limit the effects of measurement error and to avoid missing large strain fluctuations within the triangle (Alford and Taber, 2003).

Nonhomogeneous strain analyses using arrays of multiple (>3) markers have also been implemented, although not throughout multi-valued surfaces. In one of the earliest of these strain analyses, Hashima et al. (1993) fit polynomial (cubic Hermite) surfaces to 3-D marker coordinates in an end-diastolic reference state and subsequent deformed

states during the cardiac cycle in a canine heart. This enabled longitudinal, circumferential, and shear strains to be calculated over the entire domain of their marker array. More recently, a polynomial least squares fitting approach has been used to calculate strains from combined marker arrays (Kindberg et al., 2007). Particle image velocimetry has also proved useful in describing nonhomogeneous, morphogenetic strains in quail embryos, but so far this approach has only been applied in 2-D (Zamir et al., 2005b; Zamir et al., 2006).

4.1 Numerical Examples

Deformation of a flat sheet into a cylinder with the same surface dimensions (or vice versa) involves large displacements and rotations, but no strain. The current method provides accurate strain estimates even with a relatively sparse set of markers (Fig. 2.2, Table 2.1). The utility of the current method for analyzing complicated 3-D surfaces is further supported by its application to the bending and torsion of a cylinder (Figs. 2.3-5, Table 2.1). In both these examples, with adequate marker distributions and accurate measurement of marker locations, strain estimates coincide closely with exact values.

4.2 Effects of Marker Density, Measurement Error, and Fitting Radius

Analysis of the bent and twisted cylinder also illustrates the effects of marker density and measurement error. As noted above, strain estimates correspond closely to exact solutions when a dense marker array is used. However, in practice, discrepancies can arise because of practical issues intrinsic to polynomial fitting. When the fitting radius is too small, too few markers may be included in the fitting region, and fitting

parameters may not be obtained, or estimates may be sensitive to measurement error. When the fitting radius is too large, highly curved or spatially complex features will be excessively smoothed. The choice of fitting radius to balance these effects is often largely heuristic (Hashima et al., 1993), but should reflect features of the data. Characteristic length scales of the surface (radius of curvature, wavelength) are reasonable first estimates for characteristic length scales of deformation fields. Like any method to characterize local deformation, this technique inherently relies on accurate measurement of surface displacement at a sufficiently dense set of marker locations.

4.3 Applications

Recently, similar techniques have been used to measure strains in limited regions of the looping chick heart over a limited period of development (Filas et al., 2007). Using OCT, beads placed on the surface of the heart were tracked over time. Strain estimates for a stage 11+ to stage 12 chick heart are shown in Fig. 2.7. The current method allows deformations to be tracked for longer periods of time over all regions of the heart containing surface labels. It is important to recognize that, in addition to deformation attributed directly to mechanical stress (e.g., bending), morphogenetic strains can be caused by cell proliferation, cell growth, and cytoskeletal contraction (Soufan et al., 2006; Soufan et al., 2007).

Our strain estimation approach was also applied to the developing chick brain. Using OCT, time-series images were acquired, and beads were tracked on the surface of the developing chick brain. Strain estimates were calculated using the coordinates of the beads, and then mapped onto surfaces created in CARET software from the acquired images. Results are shown in Fig. 2.8.

41

Tag lines applied during an MRI experiment provide a means to track displacements. Tag MRI has been applied to the beating heart (Zerhouni et al., 1988) and to deformation of the brain (Bayly et al., 2005). Using our approach, strains can be calculated using the displacement information calculated.

Another application of our approach is to measure strains between surface representations of the developing brain at different time points. Mesh representations of the cortex are a common format for visualization and numeric analysis of the brain (Van Essen et al., 1998). In order to estimate strain between two cortical surfaces, a point-to-point correspondence is required. Surface registration is the process of determining a correspondence between two surfaces, which will be discussed in detail in Chapter III.

5. Conclusion

A novel method for measurement of surface strain on multi-valued closed surfaces was developed. This method relies on tracking the displacements of fiducial markers on evolving surfaces. Given such measurements, this technique is valuable for characterizing spatiotemporal patterns of growth in biological systems.
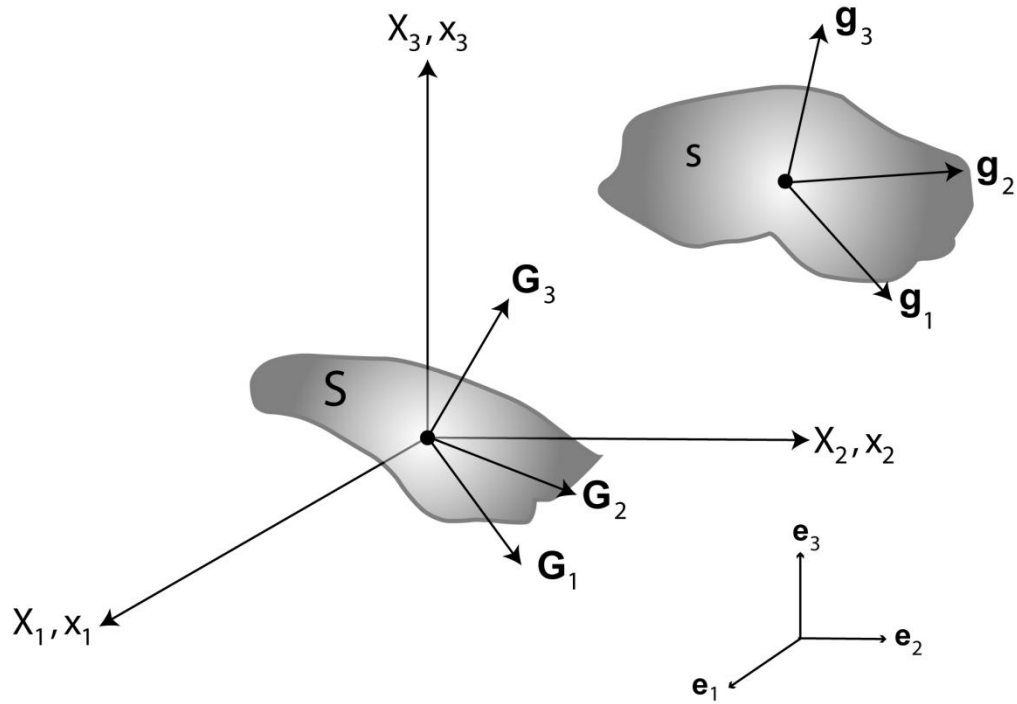
Figure 2.1: Schematic diagram showing: a patch of the reference surface, S; a patch of the deformed surface, s; the local Cartesian (X_i and x_i) coordinate systems; local orthogonal base vectors, $\mathbf{e}_i$; local covariant base vectors, $\boldsymbol{G}_i$, tangent ($\boldsymbol{G}_1$, $\boldsymbol{G}_2$) and normal ($\boldsymbol{G}_3$) to the reference surface; and convected base vectors, $\boldsymbol{g}_i$, tangent and normal to the deformed surface. Note that, in general, the orthogonal base vectors ($\boldsymbol{e}_1$, $\boldsymbol{e}_2$) of the local Cartesian system are not tangent to the surface, and the tangent base vectors ($\boldsymbol{G}_1$, $\boldsymbol{G}_2$ and $\boldsymbol{g}_1, \boldsymbol{g}_2$) are not orthogonal.
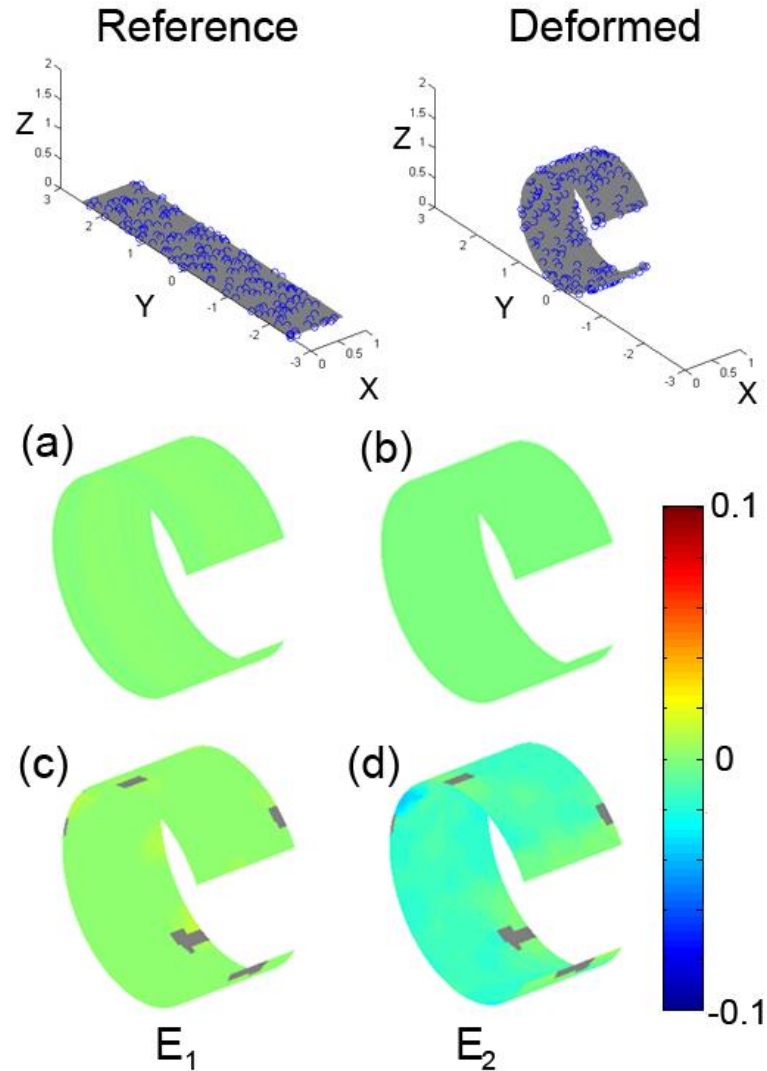
Figure 2.2: Wrapping of a flat sheet into a partial cylinder. (a,b) Estimated principal Eulerian strains using a dense, regular, marker array (1300 markers/unit area); (c,d) Estimated principal Eulerian strains using a less dense marker array (200 randomly distributed markers; 40 markers/unit area; see marker locations in upper panels). Differences between calculated and actual principal strains were on the order of 10^{-14} for the dense marker array and 10^{-2} for the sparse marker array (Table 2.1). Grey regions in the strain plots indicate regions where strain was not calculated due to insufficient marker density.
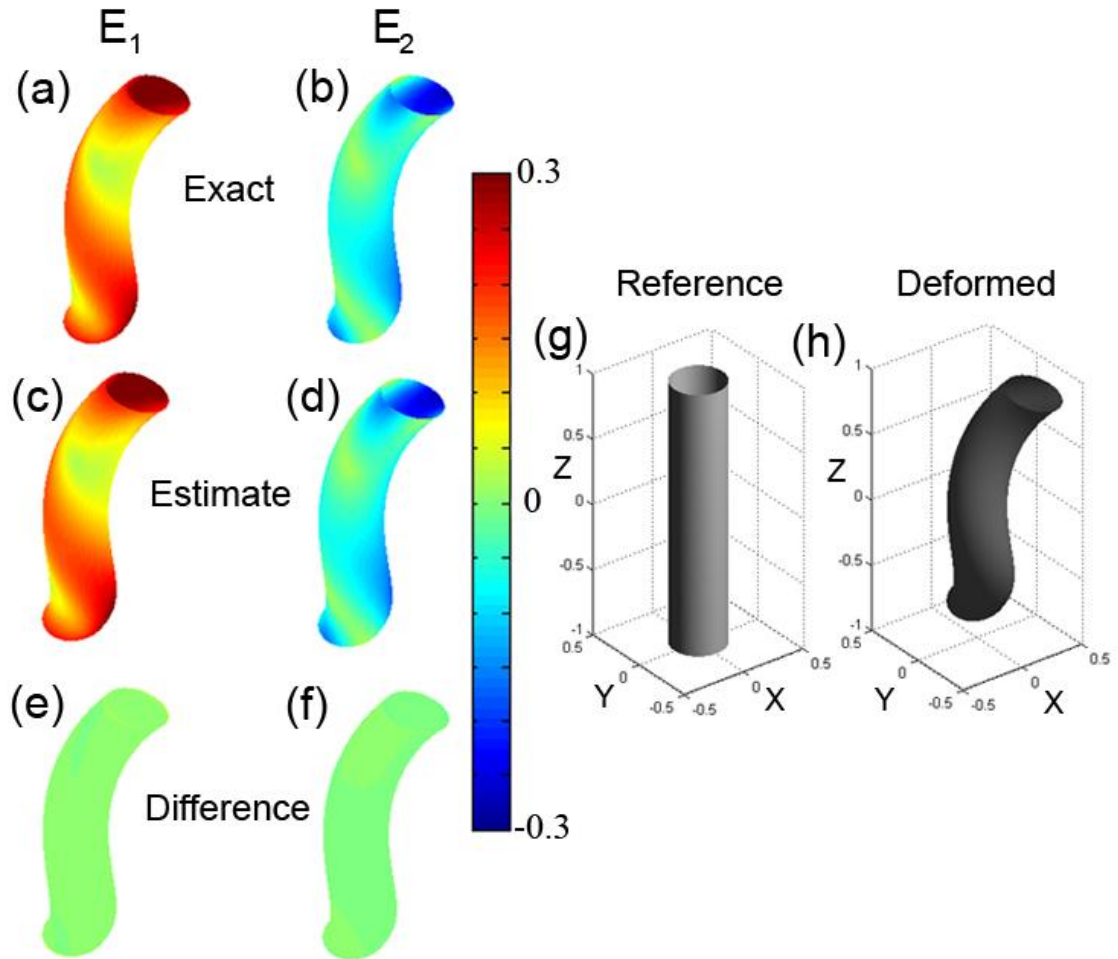
Figure 2.3: Bending and torsion of a cylinder: comparison of exact and estimated Eulerian strains. (a,b) Exact values of first and second principal strains. (c,d) Estimated values obtained with a dense, regular, array of markers (2904 markers/unit area; radius of curvature $\rho \approx 0.20$; fitting radius, $r = 0.05$). (e,f) differences between actual and estimated principal strains (see Table 2.1). (g,h) Surface representations of the undeformed and deformed cylinder.
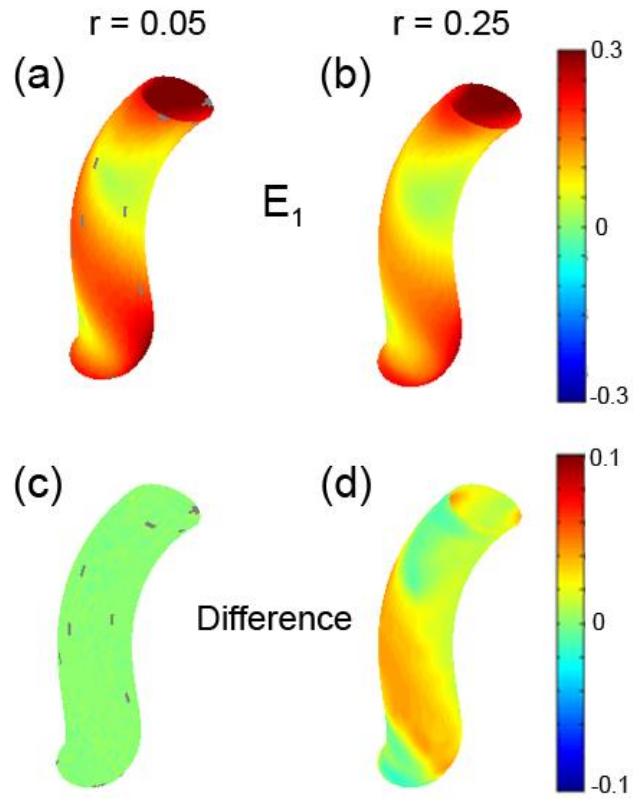
Figure 2.4: Bending and torsion of a cylinder: effect of fitting radius on Eulerian strains. (a,b) First (maximum) principal strain estimated using different fitting radii ($r = 0.05$, $r = 0.25$) with a dense, randomly scattered, set of markers (2904 marker per unit area); (c,d) differences between actual and estimated principal strains (see Table 1).
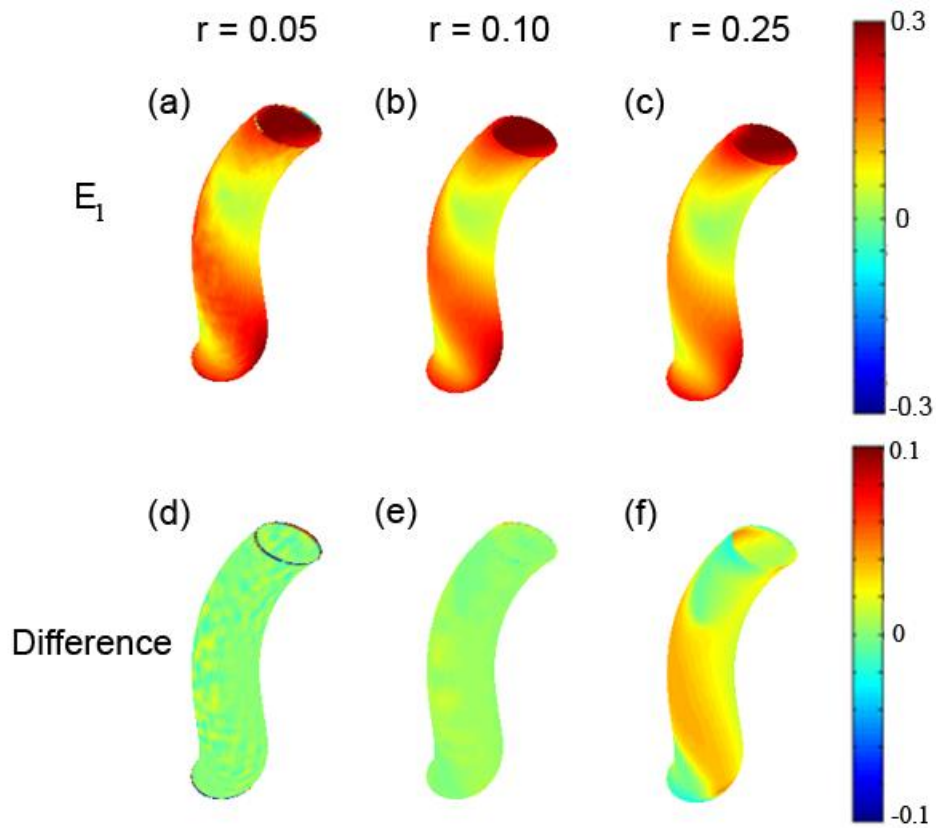
Figure 2.5: Bending and torsion of a cylinder: effect of random error in marker coordinates on first (maximum) principal Eulerian strain E_1. Random perturbations were added to both the reference (X,Y,Z) and deformed (x,y,z) coordinates (maximum error magnitude was $\pm 1\%$ of the corresponding cylinder dimension). (a) Estimates with fitting radius $r = 0.05$. (b) Estimates with fitting radius $r = 0.10$. (c) Estimates with fitting radius $r = 0.25$. (d-f) Respective differences between estimated and exact strain values. Strains are mapped onto the true (error-free) surface of the deformed cylinder. At $r = 0.05$ the added random errors noticeably affect strain estimates. Increasing r to 0.10 smoothes strain estimates while providing accurate surface fitting. Increasing the fitting radius too much (i.e., $r = 0.25 > \frac{\rho}{2}$) visibly increases the fitting error (see Table 2.1).
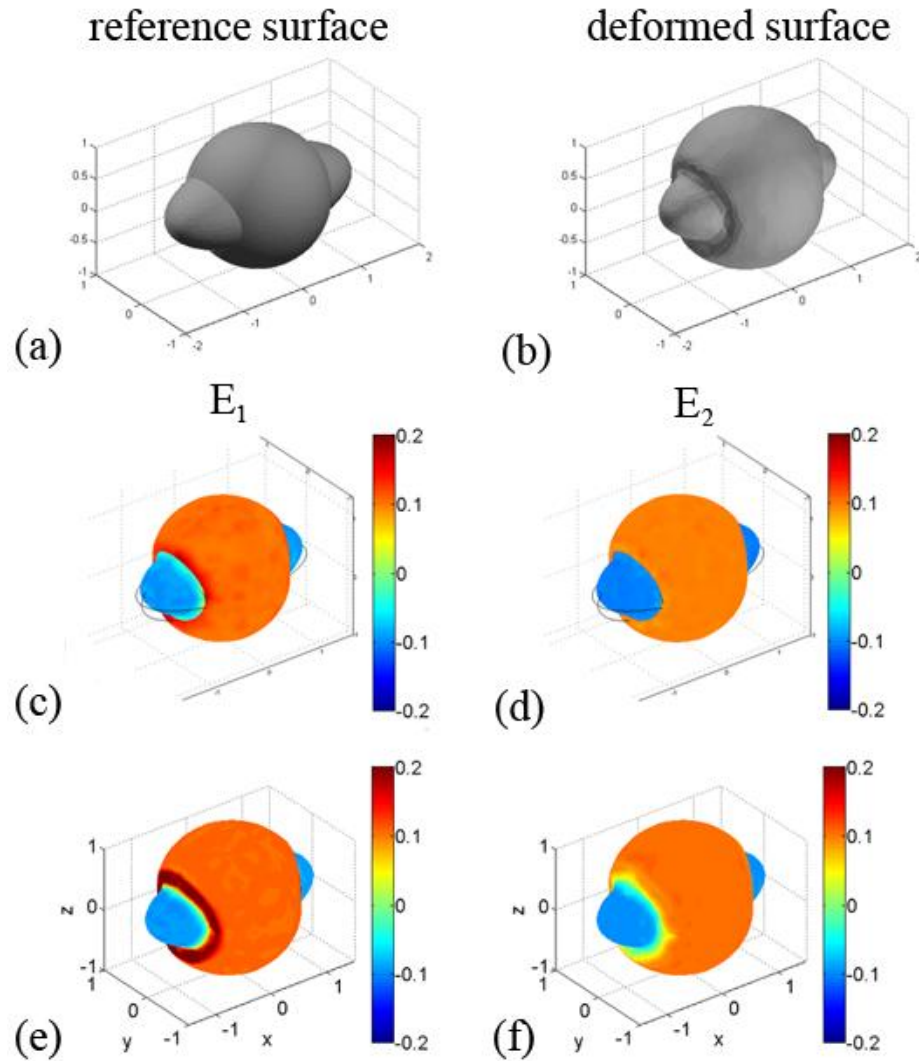
Fig. 2.6: Expansion and contraction of a complex shape: (a) The reference surface was generated in COMSOL by the intersection of the surface of a sphere with an ellipsoid. (b) Thermal expansion was applied in COMSOL to cause the spherical portion to grow by 10% and the ellipsoidal portion to contract by 10%. (c,d) First and second principal strains were calculated from the output. (e,f) The reference and deformed surface geometries were imported to Matlab, and using our approach, the first and second principal strains were estimated. Good agreement was seen between the strains estimated in COMSOL (c,d) and our approach (e,f).
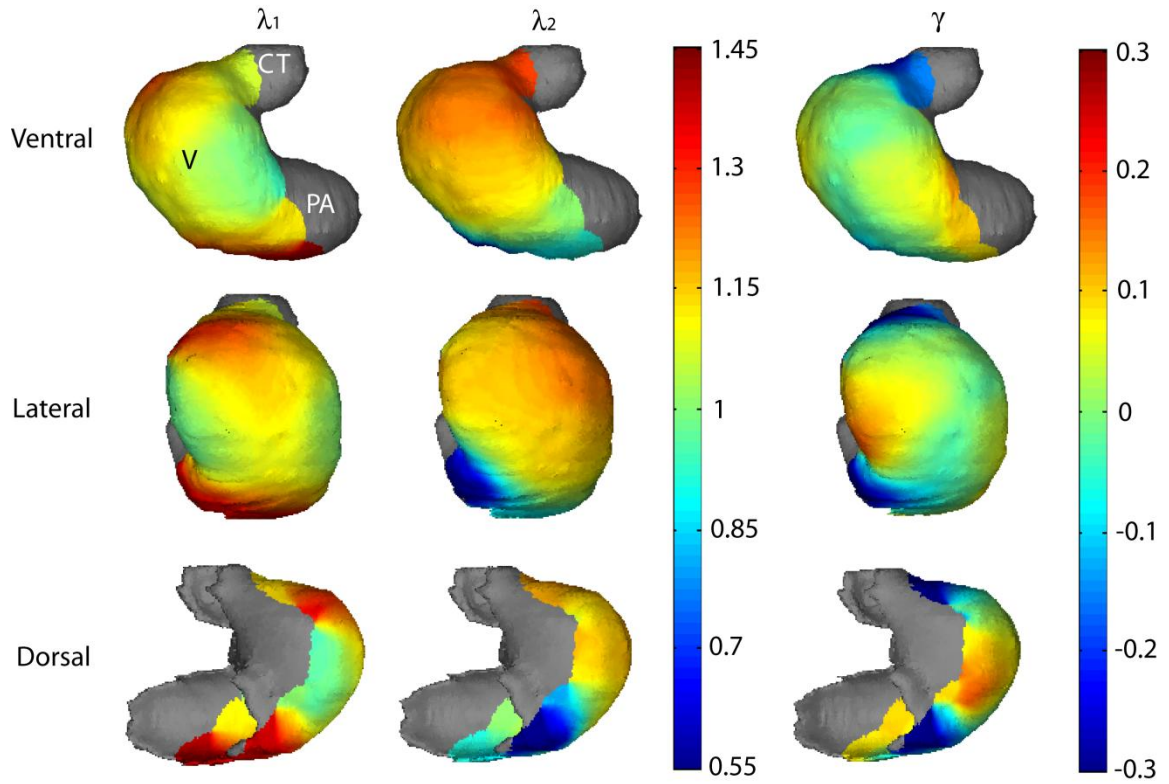
Figure 2.7: Circumferential stretch ratio (λ_1), longitudinal stretch ratio (λ_2), and shear (γ) mapped onto a stage 12 (fully c-looped) embryonic chick heart. Quantities were computed relative to the configuration at stage 11^+ (approximately five hours earlier). The circumferential and longitudinal directions were defined locally as the directions of maximum and minimum curvature, respectively. Orientations show the ventral, lateral, and dorsal surfaces of the heart. V: ventricle; PA: primitive atrium; CT: conotruncus.
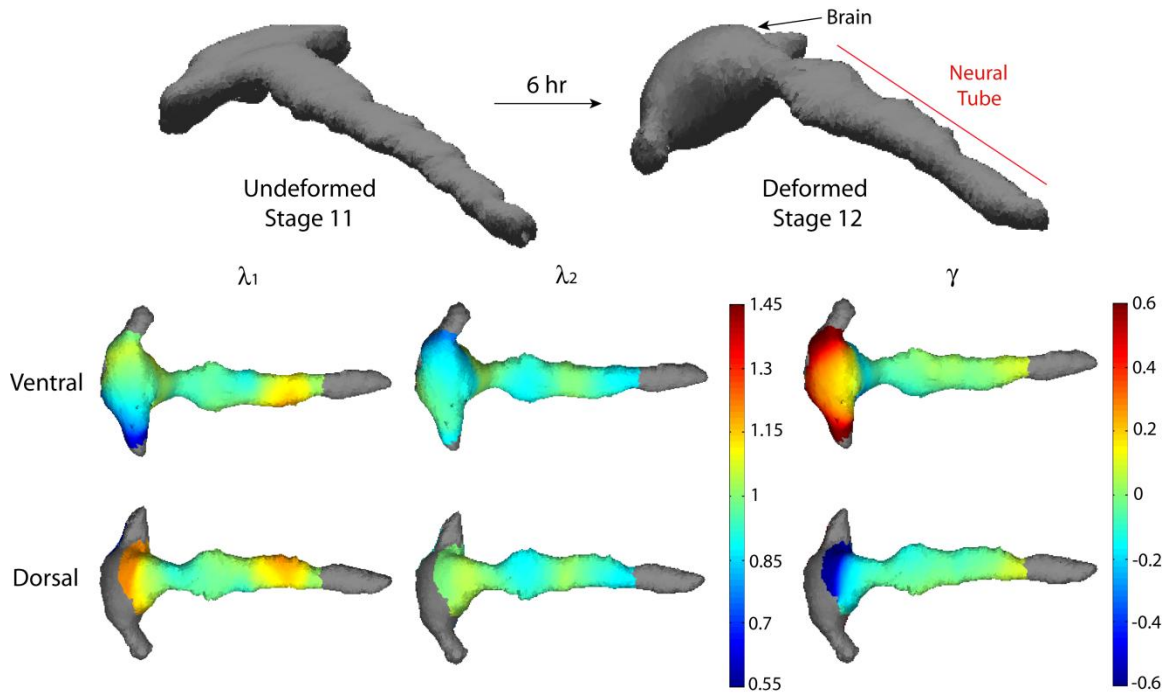
Figure 2.8: Stretch ratios in the directions of maximum curvature (λ_1) and minimum curvature (λ_2), and angular deformation (γ) mapped onto a stage 12 embryonic chick brain. Deformation measures were calculated relative to a stage 11 reference state (≈ 6 hr incubation). Note that, because of the complex geometry, the principal axes of curvature are not uniquely related to anatomical axes. As indicated by arrows, the longitudinal and circumferential directions in the midbrain and hindbrain correspond to the directions of minimum and maximal curvature, respectively. In the forebrain, the situation is reversed. Orientations show the ventral and dorsal surfaces of the brain.

Table 2.1: Summary of errors in strain estimates (compared to exact values) for known deformations of surfaces. Effects of marker density (μ) and fitting radius (r) are shown. $\overline{E_1}$, $\overline{E_2}$: average absolute error in first and second principal strains; σ_1, σ_2: standard deviation of absolute errors in the first and second principal strains; ρ: minimum radius of curvature; $\overline{N}$: the average number of points used in each polynomial fit.

		Sheet to Cylinder (Fig. 4)		Cylinder Bending and Torsion (Fig. 6)		Cylinder Bending and Torsion with Marker Location Error (Fig. 7)		
Parameters of surface and marker distribution	$1/\sqrt{\mu}$	0.03	0.16	0.02	0.02	0.02	0.02	0.02
	ρ	0.96	0.96	0.20	0.20	0.20	0.20	0.20
	r	0.10	0.40	0.05	0.25	0.05	0.10	0.25
	$\overline{N}$	40.2	16.9	24.0	578	23.0	87.4	571
Results	$\overline{E_1}$	0.0000	0.0009	0.0007	0.0191	0.0140	0.0037	0.0188
	σ_1	0.0000	0.0022	0.0005	0.0109	0.0753	0.0034	0.0109
	$\overline{E_2}$	0.0007	0.0125	0.0069	0.0268	0.0098	0.0023	0.0114
	σ_2	0.0001	0.0059	0.0141	0.0339	0.0411	0.0021	0.0059

Chapter III. Registration: Theoretical development

In this chapter, we present a novel approach for the registration of two surfaces that, using the finite element method, minimizes an energy function by solving a partial differential equation on a spherical surface. The energy function includes a "strain energy" term due to distortion and an "error energy" term due to mismatch between surface features. By minimizing the energy function, this method will bring surface features into approximate registration with as little deformation as possible.

1. Introduction

In studies of developmental biomechanics, fiducial markers are often used to track displacements over time.   However, for the developing brain *in vivo* in animals with a folded cortex, markers are not readily available. We acquire anatomical MR images at different stages of development, and create surface representations of the cortex at each time point. In order to calculate growth between cortical surface representations at different time points, surfaces must be registered to one another. Existing registration algorithms designed for inter-subject registration (e.g., CARET, Van Essen et al., 2001) were evaluated first. In order to validate existing algorithms for intra-subject registration, two test cases were created to simulate uniform growth. The cases were motivated by the requirement that in order to quantify inhomogeneous expansion of a surface, an algorithm should first be able to correctly characterize uniform expansion.

For the first test case, the deformed surface coordinates ($\boldsymbol{x}$) were set equal to the reference surface coordinates ($\boldsymbol{X}$) (i.e., zero growth). For the second test case, the reference surface coordinates were multiplied by 1.25 to create the deformed (uniformly

expanded) surface. For both cases, 7 exact landmark curves were defined and used in the registration process. The landmark curves were defined along the base of sulci. The "reference" surface can be thought of as a younger brain and the "deformed" surface as an older brain; any deformation is due to growth.

$$Case\ 1:\ \boldsymbol{x} = \boldsymbol{X}$$

$$Case\ 2:\ \boldsymbol{x} = 1.25\boldsymbol{X}$$

The deformation gradient was estimated for the mapping from the reference surface to the registered surface using the method described in Chapter II. The determinant of the deformation gradient defines the local change in area between the surfaces, and should equal unity for Case 1 and 1.563 for Case 2. However, when an existing registration algorithm for inter-subject registration (CARET) was used, the resulting estimates of areal expansion, which are shown in Fig. 3.1, suggest large spatial variations of growth throughout the cortex. The spatial variations arise because of registration errors, not intrinsic geometric deformation.

The registration algorithm in FreeSurfer (Fischl et al., 1999) and an implicit surface approach by Shi et al., (2007) were also applied to the two test cases. For the algorithm in FreeSurfer, an adult human surface was registered to itself, so the determinant of the deformation gradient should be unity. The results are shown in Fig. 3.2(a,b). For the implicit surface approach of Shi et al. (2007), the coordinates of an adult ferret cortical surface were multiplied by 1.25, as in Case 2. The determinant of the deformation gradient should be 1.563. The results are shown in Fig. 3.2(c,d). The resulting estimates of areal expansion exhibit large spatial variations of growth

throughout the cortex. Again, the spatial variations are due to errors in registration instead of actual growth.

In this chapter we introduce a method to register closed surfaces in 3-D based on the minimization of distortions while matching surface features. Our approach is similar to the methods presented in Shi et al. (2007) and Litke et al. (2005). Shi et al. (2007) take advantage of implicit descriptions of surfaces, which allows standard numerical schemes to be implemented in 3-D. Beginning from an initial map, they iteratively solve a partial differential equation (PDE) on the reference surface. The optimal registration is defined by the minimization of an energy term, which is the sum of a harmonic (smoothness) term and a data (geometric features) term. Landmarks are also used to aid in the registration. However, when this method was applied to analyze uniform growth of a ferret cortical surface, it was found to introduce artificial distortions between the surfaces. The implicit surface method, because it relies on discrete voxel size, may not be optimal for longitudinal registration of highly convoluted surfaces like the mammalian brain.

Litke et al. (2005) map open surfaces to the plane, which simplifies the computations considerably. A PDE that accounts for nonlinear large deformations is solved using the finite element method and a multi-resolution approach. The optimal registration is defined as the minimum of an energy function, which is the sum of regularization (smoothness), matching (geometric features) and bending energies. The specific approach of Litke et al. (2005) can only be applied to open surfaces, which would involve making cuts in or only looking at part of the cortical surface. The authors do not expect a one-to-one correspondence to exist between the surfaces, while our approach requires it. We deliberately do not include a penalty on bending energy, since

the distortions we wish to quantify during cortical folding include large bending deformations.

Our surface registration method is aimed ultimately at longitudinal studies of cortical development. While a number of registration algorithms are available to study inter-subject differences, these algorithms introduce distortions that prevent the accurate calculation of growth. The proposed algorithm reduces these distortions by using the finite element method to solve for displacements that minimize the sum of strain and feature energy between the cortical surfaces. A hierarchical set of test cases of increasing complexity was created to validate our method.

2. Theory

2.1 Kinematics and equilibrium

2.1.1 Surfaces

This study is restricted to surfaces with a closed topology in three dimensional (3-D) Euclidean space. These surfaces contain no edges, and their topology is equivalent to the topology of a sphere. We intend to apply this method to surfaces created from images acquired at different stages of development, which is reflected in our naming convention for each of the surfaces.

Let YAS be the *younger anatomical surface*, with coordinates X, and OAS be the *older anatomical surface*, with coordinates x, where $X, x \in \mathcal{R}^3$. The younger surface corresponds to the reference configuration while the older surface corresponds to the deformed configuration. Because both YAS and OAS have the same topology as a sphere, they can be parameterized to a sphere. In practice, we have done this using

CARET software (Van Essen et al., 2001). Let YSS be the *younger spherical surface* representation of YAS, with coordinates $\boldsymbol{Y}$, and let OSS be the *older spherical surface* representation of OAS, with coordinates $\boldsymbol{y}$, where $\boldsymbol{Y}, \boldsymbol{y} \in \mathcal{R}^3$. We assume that a point-to-point correspondence has been established between YAS and OAS and that the initial correspondence is not optimal. An alternative method for determination of an initial correspondence will be described in Chapter V, section 2.5.

The goal of the algorithm is to solve for displacements on OSS that minimize distortions between YAS and OAS while aligning matching terms on the surfaces, which is accomplished by moving the coordinates on OSS and OAS with respect to YAS. Let ROSS be the *relaxed older spherical surface*, with coordinates $\boldsymbol{y}'$, and let ROAS be the *relaxed older anatomical surface*, with coordinates $\boldsymbol{x}'$, where $\boldsymbol{x}', \boldsymbol{y}' \in \mathcal{R}^3$. The calculated displacement vectors are constrained to remain on the surface OSS, so ROSS retains the same shape as OSS. The surfaces and their relationships are shown in Fig. 3.3. Note that ROAS still corresponds to the deformed configuration, and that ROAS has the same shape as OAS. The difference between OAS and ROAS is their coordinate relationship with respect to the coordinates of YAS.

The kinematics, constitutive relations and equations of equilibrium will be described in the following in the rest of this section. For simplicity, the analysis will first be demonstrated only using OSS and ROSS. Then, the analysis will be extended to include YAS, OAS and ROAS.

2.1.2 Kinematics

The surface coordinates of ROAS are related to those of OAS by

$$x' = x + u,$$
III.1

where u is the displacement vector between OAS and ROAS. The deformation gradient

tensor, F, transforms a differential line element on a reference surface to a deformed

surface, and is defined by (Ogden, 1984) as

$$F = \frac{\partial x'}{\partial x}.$$
III.2

F can also be defined using the displacement vector (Taber, 2004):

$$F = I + (\nabla u)^T.$$
III.3

2.1.3 Constitutive relations

A constitutive law relates stress to displacement (strain). It provides an idealized

description of the macroscopic nature of a material, and varies from material to material

(Ogden, 1984). A basic example is Hooke's Law applied to a uniaxial bar:

$$\sigma = E\epsilon,$$
III.4

where σ is the stress, ϵ is the strain and E is Young's modulus. This relationship tells us

something that we know intuitively, the stiffer a material, the more stress required to

induce a displacement.

In the general theory of elasticity, a strain energy density function, W, is used to

describe the nature of a material. A material whose properties can be characterized

entirely by a scalar strain-energy density function is a hyperelastic material (Taber,

2004). The Cauchy stress (true stress), σ, is given by (Taber, 2004) as

$$\sigma = JF \cdot \frac{\partial W}{\partial F^T},$$
III.5

57

where J is the determinant of $\boldsymbol{F}$. The Cauchy stress refers to the force per unit deformed area. In linear theory, the deformed area is approximately equal to the undeformed area, but in nonlinear theory these two areas can be quite different. Consider a rectangular block of rubber that is pulled at both ends. As the block is stretched, the regions of the block away from the ends become noticeably thinner than before the block was stretched. If the original area was used, the calculated stress would not reflect the true stress in the block.

Sometimes it can be advantageous to calculate the stress tensor in terms of the reference area. The first Piola-Kirchhoff stress tensor, $\boldsymbol{t}$, refers to the force per unit area of the reference surface and is given by (Taber, 2004) as

$$t = \frac{\partial W}{\partial \boldsymbol{F}^T} \,.$$

III.6

The first Piola-Kirchhoff stress can also written as (Holzapfel, 2000; Bonet and Wood, 2008)

$$P = \frac{\partial W}{\partial \boldsymbol{F}} = \boldsymbol{t}^T \,.$$

III.7

By using the first Piola-Kirchhoff stress instead of Cauchy stress, all derivatives are taken with respect to the reference configuration, which will be helpful for the implementation.

A neo-Hookean material is described by a simple strain-energy density function that approximates the response of a rubber-like material; the strain-energy density function of a nearly incompressible neo-Hookean material is given by (Wood and Bonnet, 2008) as

$$W = \frac{1}{2}\left(\mu(I_1^* - 3) + \kappa(J - 1)^2\right),$$

III.8

where μ is the shear modulus, κ is the bulk modulus and

$$I_1^* = I_1 I_3^{-\frac{1}{3}} .$$

<div align="right">III.9</div>

Here, I_1 and I_3 are strain invariants and are given by (Taber, 2004) as

$$\begin{aligned} I_1 &= trace(F^T \cdot F) \\ I_3 &= J^2 = det(F^T \cdot F) . \end{aligned}$$

<div align="right">III.10</div>

2.1.4 Equation of equilibrium

The equation of motion can be derived by applying conservation of linear momentum (Taber, 2004)

$$\rho \frac{\partial^2 u}{\partial t^2} = \nabla \cdot t + f_0 ,$$

<div align="right">III.11</div>

where u is the displacement vector, ρ is the density and f_0 is the body force. If inertia is neglected, and the net body force is give by

$$f_0 = f - v \frac{\partial u}{\partial t} .$$

<div align="right">III.12</div>

Here, f is a matching force and $v \frac{\partial u}{\partial t}$ is a viscous force that opposes motion. By substituting III.12 into III.11 we obtain

$$v \frac{\partial u}{\partial t} = \nabla \cdot t + f ,$$

<div align="right">III.13</div>

where v is the coefficient of damping. Eq. III.13 will now be referred to as the equation of motion. The stationary solution to this equation of motion, with $f = 0$, represents a local minimum of the strain-energy function.

2.1.5 Body force

A body force, $\boldsymbol{f}$, acts on an element of material. For the problem of registration, the body force will be used to align different features between the surfaces. To achieve this we will represent the body force term as

$$\boldsymbol{f} = (f_1 - f_2)\nabla f_2 \, , \qquad \qquad \text{III.14}$$

where f_1 and f_2 are continuous, smooth functions of the spatial coordinates of YAS and ROAS, respectively. Eq. III.14 is the derived from the term (Shi et al., 2007)

$$E_f = \frac{1}{2}(f_1 - f_2)^2 \, . \qquad \qquad \text{III.15}$$

A total energy function that takes into account the strain-energy density function as well as the energy due to features on the surfaces is given by

$$\gamma = W + E_f = \frac{1}{2}(\mu(I_1^* - 3) + \kappa(J - 1)^2) + \frac{1}{2}(f_1 - f_2)^2 \, . \qquad \qquad \text{III.16}$$

2.2 Solving the equation of equilibrium using a parameterized space

Solving the equation of equilibrium on OAS is difficult to implement in practice. An easier approach is to solve it on a simple shape that has a defined relationship to OAS such as OSS. The trick is to take into account the deformations that are induced in mapping OAS to OSS, and to then determine how the displacement vector field on OSS affects the strain-energy density function between YAS and OAS.

2.2.1 Kinematics

The surface coordinates of OSS are related to those of ROSS by

$$y' = y + v,$$

$$\text{III.17}$$

where v is the displacement vector. The deformation gradient between OSS and ROSS is

$$G_1 = \frac{\partial y'}{\partial y} = I + (\nabla v)^T.$$

$$\text{III.18}$$

The deformation gradient between YAS and OAS is

$$F_0 = \frac{\partial x}{\partial X},$$

$$\text{III.19}$$

and the deformation gradient between the surfaces OSS and OAS is

$$H = \frac{\partial x}{\partial y}.$$

$$\text{III.20}$$

The deformation gradient between surfaces ROAS and YAS is

$$F = \frac{\partial x'}{\partial X} = F_1 \cdot F_0.$$

$$\text{III.21}$$

Through the use of OSS and ROSS, F_1 can be written as

$$F_1 = H(x') \cdot G_1 \cdot H(x)^{-1} = H(x') \cdot (I + (\nabla v)^T) \cdot H(x)^{-1}.$$

$$\text{III.22}$$

Note that H is evaluated at x and x'. From here on, let $H(x) = H$ and $H(x') = H'$.

Substituting Eq. III.22 into III.21 gives an expression for F in terms of the displacement

vectors between OSS and ROSS:

$$F = H' \cdot (I + (\nabla v)^T) \cdot H^{-1} \cdot F_0.$$

$$\text{III.23}$$

2.2.2 Calculation of the first Pioloa-Kirchhoff stress on ROSS

The strain-energy density function, W, is calculated in the same way, using Eq.

III.7. The first Piola-Kirchhoff stress is given by

$$P = \frac{\partial W}{\partial F} = H'^{-T} \cdot \frac{\partial W}{\partial G_1} \cdot H^T \cdot F_0^{-T}.$$

$$\text{III.24}$$

61

It is useful to write the first Piola-Kirchhoff stress in terms of the displacement vectors on OSS (G_1). The derivation given below is based on notes from Dr. L.A. Taber (unpublished).

$$P = \frac{\partial W}{\partial F} = \frac{\partial G_1}{\partial F} : \frac{\partial W}{\partial G_1} .$$

III.25

Let $E = G_1, A = H', B = F, C = F_0^{-1}, D = H$, and write

$$G_1 = H'^{-1} \cdot F \cdot F_0 \cdot H \longrightarrow E = A \cdot B \cdot C \cdot D .$$

III.26

Using indicial notation, Eq. III.26 becomes

$$E_{ij} = A_{ia} B_{ab} C_{bc} D_{cj} .$$

III.27

The derivative of G_1 with respect to F is

$$\frac{\partial E_{ij} e_i e_j}{\partial B_{mn} e_m e_n} = A_{ia} (\delta_{am} \delta_{bn}) C_{bc} D_{cj} e_m e_n e_i e_j = A_{ia} C_{bc} D_{cj} e_a e_b e_i$$

III.28

where δ is the Kronecker delta, and is defined as

$$\delta_{ij} = \begin{Bmatrix} 1 & i = j \\ 0 & i \neq j \end{Bmatrix} .$$

III.29

Substituting Eq. III.28 into III.25 gives

$$\frac{\partial E_{ij} e_i e_j}{\partial B_{mn} e_m e_n} : \frac{\partial W}{\partial E_{pq} e_p e_q} = A_{ia} C_{bc} D_{cj} \frac{\partial W}{\partial E_{pq}} (\delta_{ip} \delta_{jq}) e_a e_b = A_{ia} C_l$$

III.30

Eq. III.30 can be rewritten in tensor notation to give

$$\frac{\partial E}{\partial B} : \frac{\partial W}{\partial E} = A^T \cdot \frac{\partial W}{\partial E} \cdot D^T \cdot C^T .$$

III.31

Finally, substituting the original variables back into the expression gives

$$P = \frac{\partial G_1}{\partial F} : \frac{\partial W}{\partial G_1} = H'^{-T} \cdot \frac{\partial W}{\partial G_1} \cdot H^T \cdot F_0^{-T} .$$

III.32

Eq. III.32 describes the first Piola-Kirchhoff stress on ROAS. Now we need an expression for the first Piola-Kirchhoff stress on ROSS that gives the same rate of internal mechanical work as the first Piola-Kirchhoff stress on ROAS during deformation. The rate of internal mechanical work on ROAS is equal to the stress power per unit volume (Taber, 2004):

$$\dot{e} = P : \dot{F}.$$

III.33

Note that $\dot{e}$ is equal to $\frac{\partial e}{\partial t}$. To obtain the same rate of internal mechanical work on ROSS,

$$\dot{e} = Q : \dot{G_1},$$

III.34

where Q is the PK stress on the surface ROSS. Combining Eqs. III.33 and III.34 gives

$$\dot{e} = Q : \dot{G_1} = P : \dot{F}.$$

III.35

Substituting Eq. III.23 into III.35 gives

$$Q : \dot{G_1} = P : \left(\overline{H' \cdot G_1 \cdot H^{-1} \cdot F_0}\right) = P : \left(H' \cdot \dot{G_1} \cdot H^{-1} \cdot F_0\right).$$

III.36

Note that G_1 is the only tensor variable that is a function of time. While the surface coordinates of the surfaces ROAS and ROSS displace over time, the surfaces do not change in shape over time because the displacements are constrained to lie on the surface. Therefore H remains constant over time.

Let A, B and C be second-order tensors. Then from (Holzapfel, 2000), we have

$$A : (B \cdot C) = (B^T \cdot A) : C = (A \cdot C^T) : B.$$

III.37

Taking advantage of these useful relationships, Eq. III.36 becomes

$$Q : \dot{G_1} = (H' \cdot P) : \left(\dot{G_1} \cdot H^{-1} \cdot F_0\right) = \left(H'^T \cdot P \cdot F_0^T \cdot H^{-T}\right) : \dot{G_1}.$$

III.38

So,

$$Q = H'^T \cdot P \cdot F_0^T \cdot H^{-T}.$$

III.39

Substituting Eq. III.32 into III.39 gives

$$Q = H^{'T} \cdot \left(H^{'-T} \cdot \frac{\partial W}{\partial G_1} \cdot H^T \cdot F_0^{-T} \right) \cdot F_0^T \cdot H^{-T} = \frac{\partial W}{\partial G_1} \, . \qquad \text{III.40}$$

2.2.3 Equation of equilibrium

The equation of equilibrium can be derived in the same manner as Eq. III.13. The equation of motion in terms of forces on ROSS is given by

$$v \frac{\partial v}{\partial t} = \widetilde{\nabla} \cdot Q^T + g \, , \qquad \text{III.41}$$

where g is the body force on ROSS and $\widetilde{\nabla}$ is the gradient operator on OSS ($\widetilde{\nabla} = \frac{\partial}{\partial y}$). The body force g acts on the surface ROSS, while the body force f acts on the surface ROAS. These two forces are related by the deformation gradient H:

$$g = H^{-1} \cdot f \, . \qquad \text{III.42}$$

2.3 Derivation of the weak form

It is advantageous to write Eq. III.41 in its weak form. By computing the weak form, the order of spatial derivatives in the divergence of stress term is reduced from 2 to 1. The estimation of derivatives using numeric approximations has error associated with it that increases as the order of the derivative increases. So, using the weak form, the error due to the estimation of derivatives is reduced. We follow the approach described in Szabó and Babuška (1991) in deriving the weak form of the equation of equilibrium.

Using indicial notation, the equation of equilibrium can be written as

$$v \dot{v}_i = Q_{ij,j} + g_i \, . \qquad \text{III.43}$$

64

Note that Latin indices take the value of 1, 2, 3. The usual summation convention with respect to repeated indices is implied. A comma followed by an index represents a spatial derivative.

To obtain the weak form, multiply Eq. III.43 by a test function w_i and integrate over the area:

$$\int_A v\dot{v}_i w_i \, dA = \int_A Q_{ij,j} w_i \, dA + \int_A g_i w_i \, dA .$$

III.44

Consider first the integral that contains the stress tensor Q_{ij}. Applying the product rule allows us to rewrite the product of the stress tensor and the test function as

$$Q_{ij,j} w_i = \left(Q_{ij} w_i \right)_{,j} - Q_{ij} w_{i,j} .$$

III.45

The Divergence Theorem can then be applied:

$$\int_A \left(Q_{ij} w_i \right)_{,j} dA = \int_C Q_{ij} n_j w_i \, dC .$$

III.46

Here n_j is the normal vector in the outward direction on the surface, and C is the edge. No edges exist on the surfaces, so

$$\int_C Q_{ij} n_j w_i \, dC = 0 .$$

III.47

Substituting Eqs. III.45-7 into III.44 and combining all of the terms into a single integral gives

$$\int_A \left(v\dot{v}_i w_i + Q_{ij} w_{i,j} - g_i w_i \right) dA = 0 .$$

III.48

Because this must hold true for an arbitrary domain, the quantity within the integral must be equal to zero. The final form of the weak form of the equation of equilibrium is given by

$$v \dot{v}_i w_i + Q_{ij} w_{i,j} - g_i w_i = 0 \,.$$

<div align="right">III.49</div>

2.4 Non-dimensionalization

Non-dimensionalization of an equation is a useful method to help determine the relative effects of different variables in the equation. Non-dimensional variables must first be defined to replace the dimensional variables. Let τ represent a non-dimensional time, V represent a non-dimensional displacement vector, $\tilde{\nabla}$ represent a non-dimensional gradient operator, $\tilde{Q}$ represent a non-dimensional stress tensor, and G represent a non-dimensional force vector. Relationships between the non-dimensional and the dimensional variables are necessary to convert the dimensional equation of equilibrium to a non-dimensional equation. The choice of variables that relate the dimensional and non-dimensional variables is important because it provides insight into how the variables relate to one another. The relationships between the dimensional and non-dimensional variables are:

$$
\begin{aligned}
t &= t_c \tau \quad & t_c &= \frac{\nu A}{\mu} \\
g &= g_c G \quad & g_c &= \frac{\mu}{\sqrt{A}} \\
Q &= \mu \tilde{Q} \quad & v &= \sqrt{A} V \\
\nabla &= \frac{\tilde{\nabla}}{\sqrt{A}} &
\end{aligned}
$$

<div align="right">III.50</div>

where A is the surface area of OSS.

Applying the chain rule to Eq. III.41 gives

$$\frac{\nu \sqrt{A}}{t_c} \frac{\partial V}{\partial \tau} = \frac{\mu}{\sqrt{A}} \tilde{\nabla} \cdot \tilde{Q} + \frac{\mu}{\sqrt{A}} G \,.$$

<div align="right">III.51</div>

Substituting the expressions in Eq. III.50 for g_c and t_c into Eq. III.51 then gives the final

form of the non-dimensional equation of motion:

$$\frac{\partial V}{\partial \tau} = \tilde{\nabla} \cdot \tilde{Q} + G \, . \qquad\qquad \text{III.52}$$

3. Methods

In this section we describe in detail the implementation of the theory presented in

the previous section. Using COMSOL, the finite element method is applied to solve Eq.

III.49 for displacement vectors on OSS. A flow chart, shown in Fig. 3.4, lists the steps in

the registration algorithm.

3.1 Initial correspondence

A point-to-point correspondence between YAS, OAS and OSS is required. For

the test cases presented in Section 4, the initial relationship between all of the surfaces is

known a priori because mathematical operations are performed to create the OAS. When

analyzing the surfaces created from MRI data acquired at different time points, an initial

correspondence is not known. The steps required to determine an initial correspondence

for actual data will be described in more detail in Chapter IV, Section 2.

3.2 Calculation of deformation gradients and curvature in Matlab

The surfaces that correspond to YAS, OAS and OSS are imported into Matlab

from CARET, which is done by saving the coordinate and topology files (.coord and

.topo) in CARET in the ASCII format. Using the program Wordpad (TM), or similar, the

header information is deleted from each of the files, and the files are saved under a new name. The file can then be loaded into Matlab using the *load* command.

The deformation gradient between the fiducial surfaces YAS and OAS, $\boldsymbol{F_0}$, is calculated using the function, *calc_deformation_gradient*, written in Matlab. This function uses a similar approach to the method described in Chapter II, Section 2. Derivatives are estimated at a given surface point using surface fitted to all other points within a user-defined radius r. The variable r is chosen based on the density of the surface coordinates as well as the curvature. For this approach we require that at least 8 surface coordinates be present within the radius r for any given local patch in order to estimate the coefficients in Eq. II.19. Chapter II, Section 4.2 provides a discussion on the fitting radius.

One consequence of this approach is that if too few surface coordinates are present within the radius r for a specific surface coordinate, then an estimate of the deformation gradient will not be obtained at that surface coordinate. However, an estimate of the deformation gradients $\boldsymbol{F_0}$ and $\boldsymbol{H}$ are required at every surface coordinate. In order to ensure that an estimate is calculated at every surface coordinate, the nearest n points are used for the function *calc_deformation_gradient*, where n is an integer greater than 8 and is defined by the user.

3.3 Implementation in COMSOL

3.3.1 Basic setup in COMSOL

The use of COMSOL to apply the finite element method as a means to solve a PDE is advantageous for this case because COMSOL is compatible with Matlab and also

gives the user a large amount of control. The PDE weak form on the boundary is first

selected in COMSOL with dependent variables u, v and w, where

$$v = ue_x + ve_y + we_z .$$ III.53

A spherical surface of radius 1 is then created in COMSOL. Eq. III.49 is entered as the

weak form (in the 'Boundary Settings'). The equations for F, G_1, Q, W, γ and f are

entered (as 'Scalar Expressions') in component form (i.e., $F = F_{ij}e_ie_j$, where F_{ij} are the

components of the tensor F). The damping coefficient v, bulk modulus, κ, and shear

modulus, μ, are defined in 'Constants'.

F_0, H, f_1 and f_2 need to be imported into COMSOL, which is achieved using two

different approaches. H and f_2 are both functions of the coordinates on the surface

ROSS, and therefore change over time. The components of H and f_2 are entered as

'Functions' in COMSOL. The components of F_0 and f_1 do not vary in time and are

defined as dependent variables in the model navigator using the same approach as the

components of the displacement vector. Both F_0 and f_1 are interpolated into COMSOL

using functions written in Matlab.

The finite element mesh is generated in COMSOL. COMSOL has nine preset

options to define the density of the finite element mesh (Table 3.1). The best resolution

depends on the spatial frequency with which F_0 and H vary over the surface, and will be

discussed in more detail in Section 5.2.

3.3.2. Defining 'Functions' in COMSOL

The components of H and f_2 may be imported into COMSOL as functions of

space using the 'Functions' option, selecting the 'Interpolation' button and entering each

of the variable names in Matlab. The advantage to using this approach is that the functions that are created can be called as a function of the initial coordinates plus the displacement components that are calculated in COMSOL. COMSOL requires that 'Functions' be entered in a specific format and only in two dimensions. For problems on the surface of a sphere, this means that the 'Functions' must be created as a function of the spherical coordinates θ and φ.

3.3.3 Interpolation functions

The interpolation functions work by finding the three closed surface coordinates on OSS to a coordinate on the finite element mesh with the same radius. A weighted average is computed to determine the value of the quantity being interpolated. Suppose that we have some value A_i at each coordinate location on the surface OSS that we want to be interpolated onto each coordinate of the finite element mesh. For a single coordinate on the finite element mesh, the nearest three surface coordinates on OSS are found and the magnitude of the displacement between them, d_i, is computed. The interpolated value of A_i at that specific coordinate of the finite element mesh is given by

$$B = \frac{\frac{A_1}{d_1} + \frac{A_2}{d_2} + \frac{A_3}{d_3}}{\frac{1}{d_1} + \frac{1}{d_2} + \frac{1}{d_3}}. \qquad \text{III.54}$$

The calculation is repeated for every surface coordinate on the finite element mesh. The interpolation function *interp_F0* is provided in the Appendix, section 2.3.

3.3.4 Solution

A "transient analysis" is performed to compute the solution to the equation of equilibrium (Eq. III.49). Displacements were obtained using the time-dependent solver with the generalized minimum residual (GMRES) linear solver (Y. Saad and M. H. Schultz, 1986) with Incomplete LU preconditioning. The projection of spatial derivatives onto the tangent plane was calculated using functions within COMSOL. The tangential components of the spatial gradient are given by (COMSOL manual)

$$(\nabla v)_T = (I - nn^T) \cdot \nabla v \,, \qquad\qquad \text{III.55}$$

where n is the normal vector on OSS. Displacements are constrained to remain on the surface by forcing the radius of the sphere to remain constant as the solution is calculated.

3.4 Post-processing steps in Matlab

3.4.1 Updating surface coordinates

The equation of equilibrium is solved for the displacement vector, v, on OSS. The surface coordinates of ROSS, y', are determined using Eq. III.17. From the one-to-one relationship between OSS and OAS, the surface coordinates of ROAS, x', are determined. In practice, the coordinates of the surface mesh are updated. The custom written functions *project_to_surf* and *surf_to_surf*, which are included in the Appendix, are then used to project the updated coordinates onto OSS and to find the corresponding coordinate on OAS.

The triangular mesh that describes OSS consists of N vertices and M faces. Let the jth face of this mesh be referred to as F^j. The vertices of F^j have coordinates y_1^j, y_2^j and y_3^j, which can be used to form a basis on the face:

$$v_1^j = y_2^j - y_1^j$$
$$v_2^j = y_3^j - y_1^j \quad .$$

III.56

A basis is a set of vectors that are linearly independent and their linear combinations,

$$\alpha v_1^j + \beta v_2^j ,$$

III.57

where α and β are constants, describe all points in the space that the basis vectors exist, which for our case is F^j (Kreyszig, 2006). So, we are able to describe the location of any location on F^j using a linear combination of v_1^j and v_2^j. Because OAS is represented by a corresponding mesh, the faces and vertices have a one-to-one correspondence with those of OSS. Let f^j be the jth face of OAS that corresponds to F^j. We define a basis on f^j as well:

$$V_1^j = x_2^j - x_1^j$$
$$V_2^j = x_3^j - x_1^j \quad ,$$

III.58

where x_1^j, x_2^j and x_3^j are the coordinates of the vertices of the face. Using the coefficients α and β that describe any location on F^j, the corresponding location on f^j is given by

$$\alpha V_1^j + \beta V_2^j \quad .$$

III.59

3.4.2 Convergence

Here we define two measures to determine whether the solution converged. The measures we will use for convergence look at the total displacement and change in energy calculated by the solution. Let ε and ρ be the measures of convergence given by

$$\varepsilon = \frac{1}{N}\sum_{i=1}^{N} |v_i| \le 0.001$$

$$\rho = \left| \frac{\Sigma \gamma}{\Sigma \gamma_0} - 1 \right| \le 0.01 \quad ,$$

III.60

where v is the displacement vector on OSS, N is the number of vertices of the wire mesh and γ_0 and γ are the initial and final energies calculated using Eq. III.62. If neither criterion is met, then Steps 4-6 in Fig. 3.4 are repeated until one of the criterion is met.

3.4.3 Estimation of local growth

For general 3-D analysis, the determinant of the deformation gradient is the ratio of the deformed volume to the undeformed volume. The component of the deformation gradient that is normal to the surface has been normalized for both the reference surface and the deformed surface because the "thickness" of the surface does not change from the reference configuration to the deformed configuration. Another way to think about this is that the both the reference and deformed surfaces have a unit thickness. Because the "thickness" component is the same for the reference state and deformed state, the determinant of the deformation gradient, which is the dilatation ratio, represents the ratio of local surface area between two surfaces:

$$J_0 = det(F_0)$$
$$J = det(F) \quad .$$

III.61

Strain tensors are estimated by applying the approach described in Chapter II:

$$E_0 = \frac{1}{2}(F_0^T \cdot F_0)$$
$$E = \frac{1}{2}(F^T \cdot F) \quad .$$

III.62

73

4. Conclusion

   In this chapter, we present a novel approach for the registration of two surfaces that minimizes an energy function by solving a partial differential equation on a spherical surface. The energy function includes a "strain energy" term due to distortion and an "error energy" term due to mismatch between surface features, so that minimizing the energy function will bring surface features into approximate registration with as little deformation as possible. The finite element method is applied to solve the partial differential equation on the surface. In the next chapter, we will validate our method by applying it to a hierarchical set of test cases.
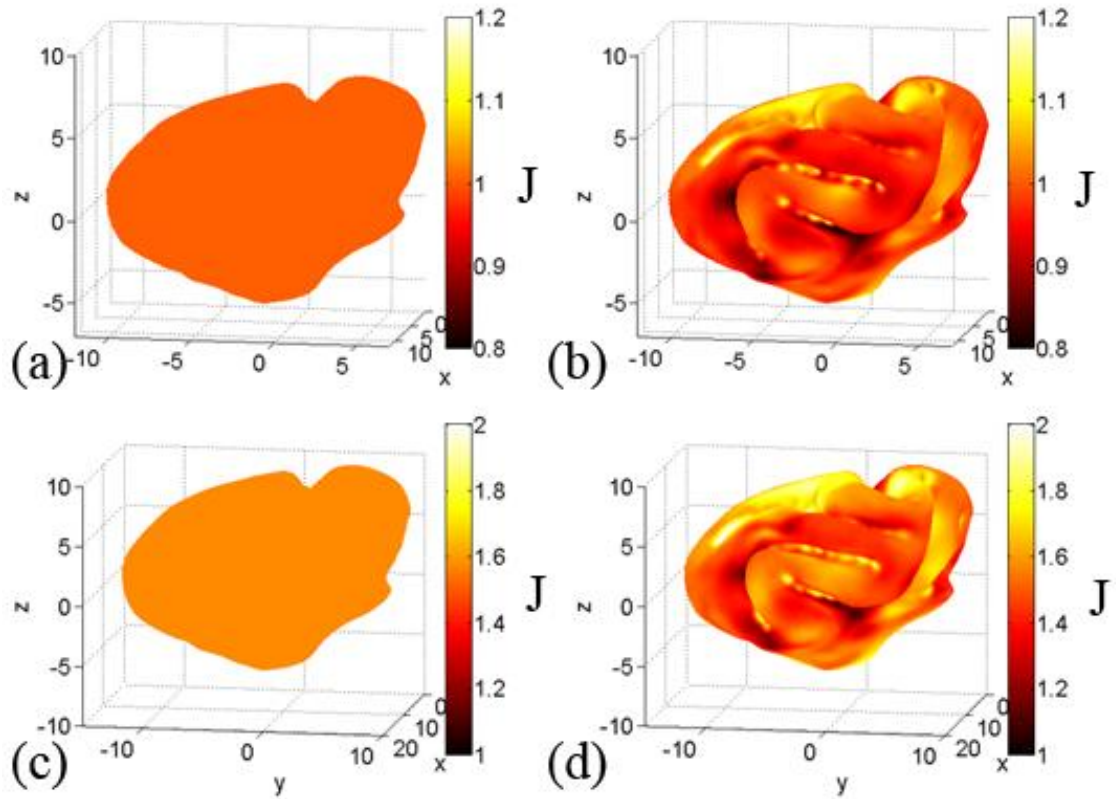
Figure 3.1: This figure shows the results of two test cases that were analyzed using the registration algorithm in CARET software. The first case involves the registration of a P21 ferret cortical surface to itself, and the second case involves the registration of a P21 ferret cortical surface to a surface whose coordinates were multiplied by 1.25. (a,c) The dilatation ratio should be 1.0 for the first case and 1.56 for the second case. (b,d) For both cases, the dilatation ratio of the output from CARET varies spatially, which is due to artificial distortions introduced during registration, not actual variations growth.
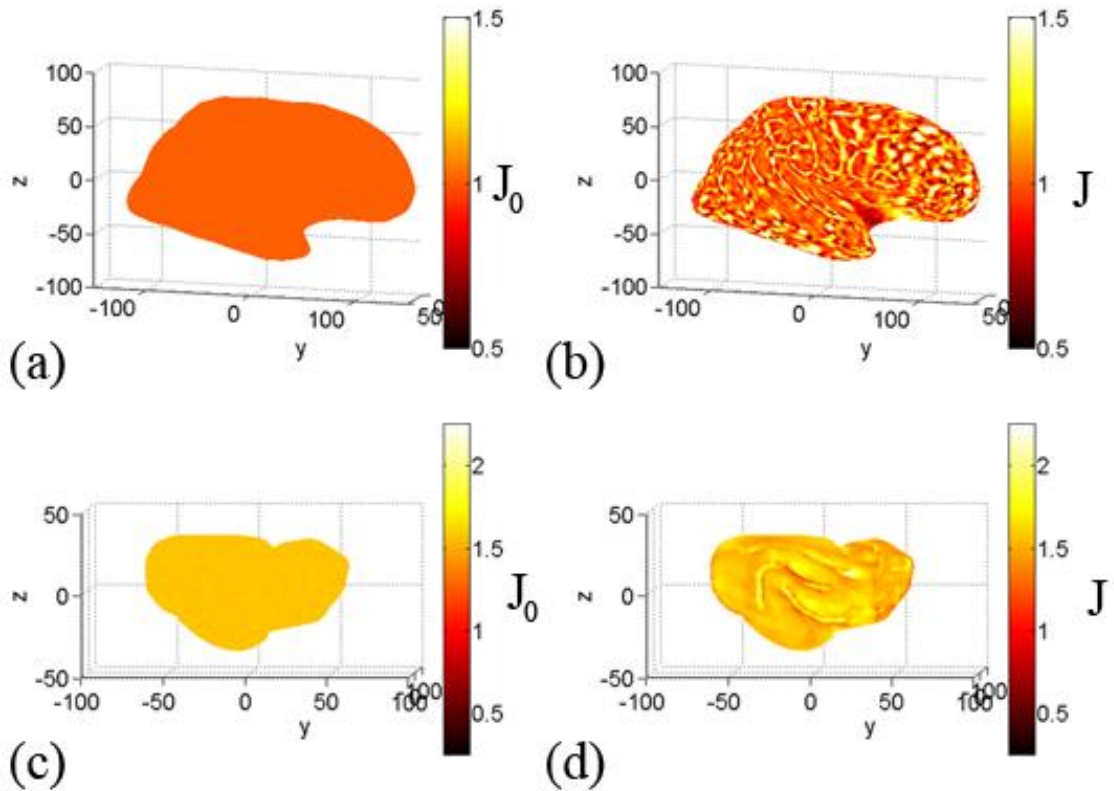
Figure 3.2: Test cases analyzed using the registration method in FreeSurfer and an approach by Shi et al. (2007): An adult human cortical surface was registered to itself. Results are plotted on an inflated representation of the cortical surface (a) Exact dilatation ratio for the registration of a surface to itself. (b) Dilatation ratio calculated from the output of FreeSurfer. Using an approach by Shi et al. (2007), the coordinates of an adult ferret cortical surface were multiplied by 1.25, which mimics an "older" ferret brain. (c) Exact dilatation ratio for the case of uniform expansion by a factor of 1.25. (d) Dilatation ratio estimated from the output of the approach by Shi et al. (2007).
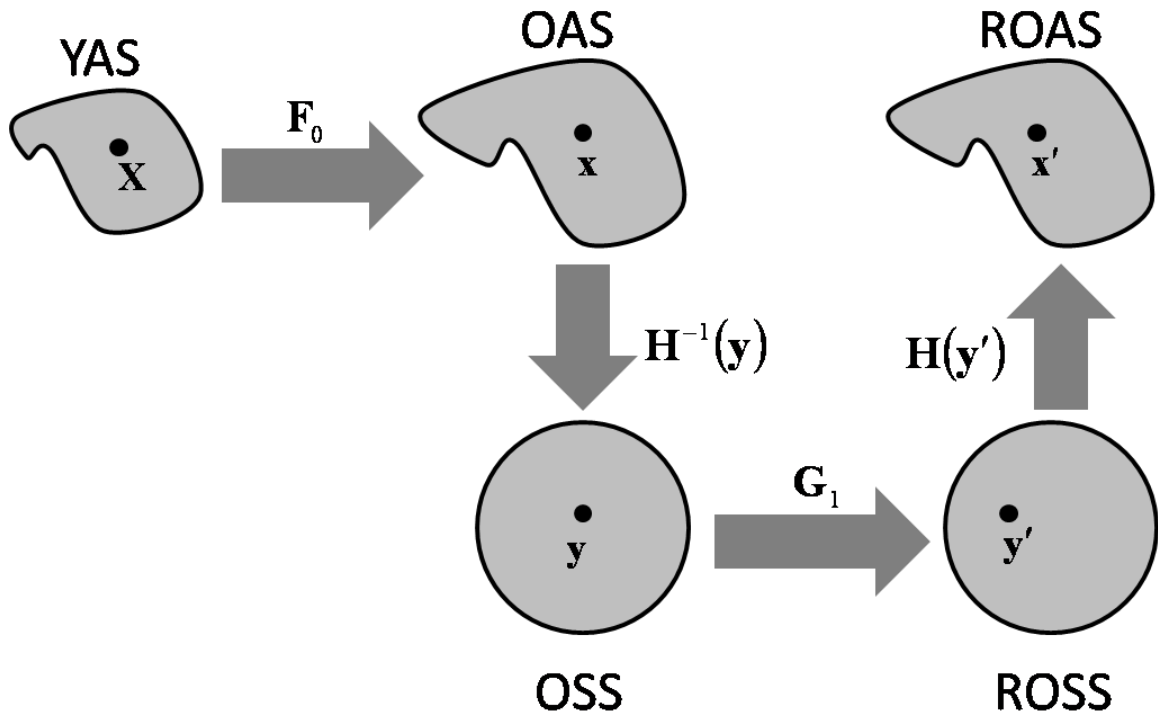
Figure 3.3: Surfaces and surface relationships: YAS is the younger anatomical surface, OAS is the older anatomical surface, ROAS is the deformed older anatomical surface, OSS is the older spherical surface and ROSS is the deformed older spherical surface. F_0 is the deformation gradient between OAS and YAS, H is the deformation gradient between OSS and OAS (as well as DOSS and DOAS) and G_1 is the deformation gradient between ROSS and OSS. The deformation gradient between ROAS and YAS is given by the tensor product of H, G_1 and F_0. G_1 is a function of the displacement vectors between ROSS and OSS. Displacements are calculated using the finite element method that minimizes distortions between YAS and ROAS while aligning matching features.
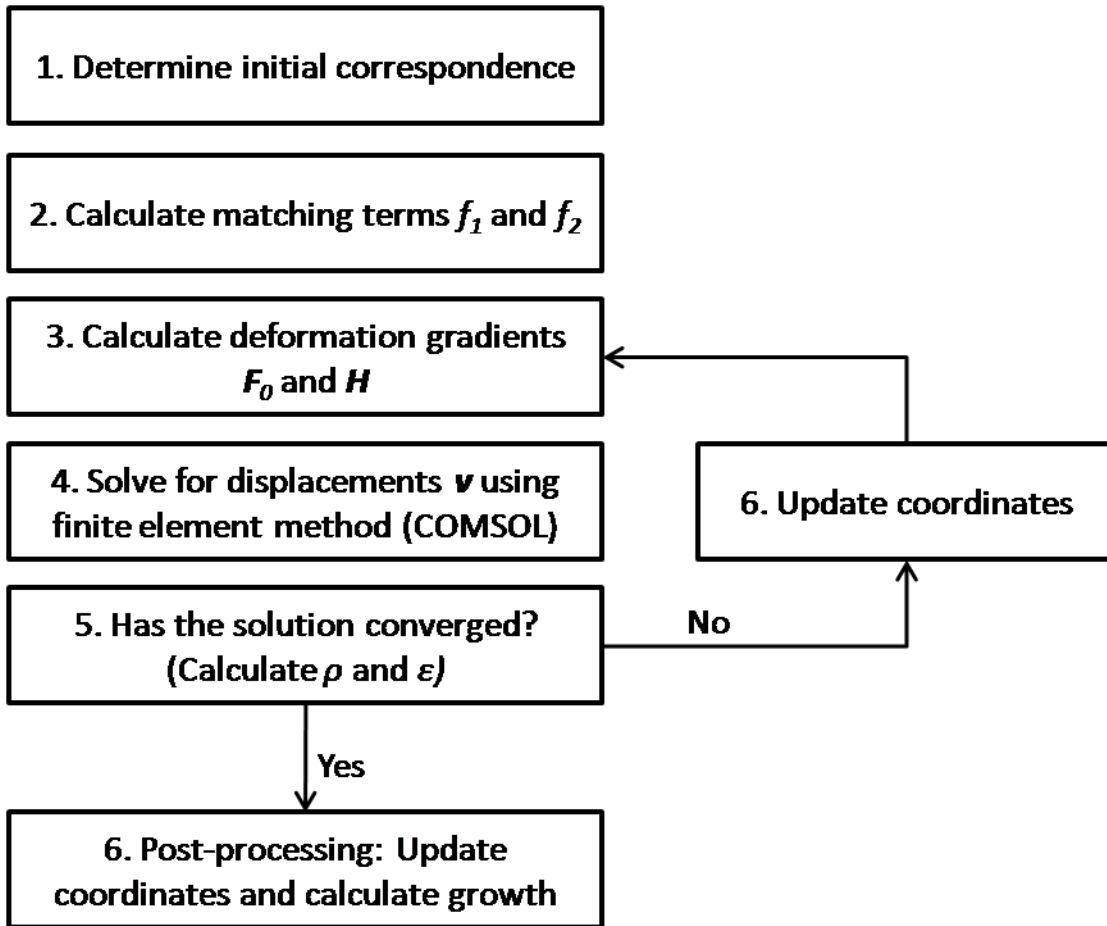
Figure 3.4: Registration algorithm: An initial correspondence is determined between YAS and OAS. The matching terms f_1 and f_2, along with the deformation gradients F_0 and H are calculated. The finite element method is used to solve Eq. III.49 for displacements on OSS. Two measures of convergence are checked. If the convergence criterion is not met, then the coordinates are updated, and steps 3-5 are repeated. Once the convergence criterion is met, the coordinates are updated and the surfaces are analyzed.

Table 3.1: This table shows the number of elements, N, in the finite element mesh for each of the preset settings in COMSOL.

Mesh Setting	N
Extremely Coarse	80
Extra Coarse	152
Coarser	320
Coarse	592
Normal	1240
Fine	1796
Finer	3852
Extra Fine	9068
Extremely Fine	26700

Chapter IV. Registration: Validation and parametric studies

1. Introduction

Here we present a hierarchical set of test cases to validate our approach. For the first two cases, the surface of a sphere was registered to itself. Distortions were introduced for both cases, and surface matching terms were used to align regions in the second case. The effect of the density of the finite element mesh was analyzed in the third case. The fourth case examined the effect of the damping term, ν. The fifth case simulated folding by registering a sphere to a "pumpkin" (lobed surface). The sixth case looked at the effect of the ratio of the shear modulus, μ, to the bulk modulus, κ. Finally, for the seventh case, a closed region on a P14 cortical surface was heavily smoothed and then registered to the original surface.

2.1 Registration of a sphere to itself

2.1.1 Case 1: Registration of a sphere to itself with the introduction of artificial distortions

The surface YAS was set to be a sphere of radius 1 that consisted of 7542 vertices and 15080 faces. Displacements were applied to YAS in spherical coordinates to generate OAS:

$$
\begin{aligned}
\phi &= \Phi \\
r &= R \\
\theta &= \Theta + \frac{0.1}{\pi} sin(4\Theta)cos^2(\Phi) \quad .
\end{aligned}
\tag{IV.1}
$$

The coordinates for OSS were given by

$$y = X.$$ <div style="text-align: right">IV.2</div>

Using the function *calc_deformation_gradient* in Matlab, the tensors H and F_0 were calculated. From F_0, the initial dilatation ratio and strain tensor were determined from Eqs. III.61-62. The surface matching terms f_1 and f_2 were set to zero. The initial distortions are visualized by the plots of the dilatation ratio and strain-energy density function, which are shown in Fig. 4.1(a,c), and by the eigenvectors of the strain tensor, which are shown in Fig. 4.2 (recall that the dilatation ratio is the ratio of the area of a region on OAS to the same region on YAS). The applied displacements created an alternating pattern of regions of expansion and compression between YAS and OAS. The estimation of strain adds information on the directions of stretching and shrinking. The goal in applying the algorithm was to minimize the distortions between YAS and OAS while aligning f_1 and f_2. Because f_1 and f_2 were both set to zero, there should be no distortions present after relaxation.

For this case, the finite element mesh consisted of 1240 elements. A list of the parameters used for this test case is found in Table 4.2. The approach was applied for two iterations before convergence was obtained. The total run time of the two iterations in COMSOL was 12.7 minutes, which does not include the steps performed in Matlab. Measures of convergence and summary statistics are listed in Table 4.3.

Once the solution converged, the calculated displacements were applied to update the coordinates on OSS and OAS and used to calculate the deformation gradient, which was in turn used to calculate the dilatation ratio and strain tensor. The final values of the total energy function and the determinant of the deformation gradient are plotted in Fig. 4.1(b,d), and the eigenvectors associated with the principal strains are plotted in Fig. 4.2.

These images confirm that initial distortions were greatly reduced during the relaxation process.

The average value of the determinant of the deformation gradient over the surface remained unchanged at 1.000, and the standard deviation is reduced from 0.065 to 0.001. A similar reduction is seen with the principal strains. The first and second principal strains are reduced from 0.029 and -0.027 to 0.011 and -0.012, and the standard deviations are reduced from 0.038 and 0.035 to 0.005 and 0.005, respectively.

2.1.2 Case 2: Registration of a sphere to itself with the introduction of artificial distortions and matching surface features

YAS, OAS and OSS were the same as the surfaces in Case 1, and therefore H and F_0 were also the same as in the previous case. As before, F_0 was used to calculate the dilatation ratio and the initial strain tensor. The surface matching terms f_1 and f_2 were defined by

$$f_1 = 10e^{\frac{-25(X_1 - 0.15)^2}{2}}$$
$$f_2 = 10e^{\frac{-25x_1^2}{2}} \quad .$$

IV.3

and are shown in Fig. 4.3. The goal of the registration algorithm is to remove the initial distortions while aligning f_1 and f_2, which requires one side of the surface to be stretched and the other side to be compressed.

The finite element mesh consisted of 1240 elements. A list of the parameters used for this test case is found in Table 4.1. The approach was applied for two iterations before convergence was obtained. The total run time of the two iterations in COMSOL was 48.8

minutes. This time does not include the steps performed in Matlab. Measures of convergence and summary statistics for each iteration are listed in Table 4.2.

Once the solution converged, the calculated displacements were applied to update the coordinates of OSS and OAS. The deformation gradient F was calculated and then used to calculate the dilatation ratio and strain tensor. The initial distortions were removed during the relaxation process. The total energy function and dilatation ratio are plotted in Fig. 4.4(b,d), and the eigenvectors associated with the principal strains are plotted in Fig. 4.5(b,d).

The difference between the surface matching terms decrease by approximately one order of magnitude, and is shown in Fig. 4.3(c,d). The initial distortions, shown in Figs. 4.4(a,c)-4.5, decay during the registration. Because the surface features align with one another, the positive x side of the sphere expands, while the negative x side of the sphere contracts. This pattern is seen in the plots of the strain-energy density function (Fig. 4.4b), dilatation ratio (Fig. 4.4d) and principal strain estimates (Fig. 4.5).

2.1.3 Case 3: Effect of mesh size on the registration of a spherical surface to itself

The goal of this test case was to compare how the mesh density affects the solution. YAS was a sphere of unit radius, described by 7542 vertices and 15080 faces. Displacements were applied to YAS in spherical coordinates to generate OAS:

$$
\begin{aligned}
\phi &= \Phi \\
r &= R \\
\theta &= \Theta + \frac{0.05}{\pi} sin(8\Theta)cos^2(\Phi) \; .
\end{aligned}
$$

IV.4

The coordinates for OSS are given by

$$y = X.$$ <div style="text-align:right">IV.5</div>

Note that the spatial frequency of the distortions increased from the previous two cases, which helps to highlight the effect of the number of elements in the finite element mesh on the solution. Three different settings were used to create the finite element mesh, 'Coarser' (320 elements), 'Normal' (1240 elements) and 'Finer' (3852 elements), shown in Fig. 4.6. The solution was obtained with each mesh, and the results were compared..

The tensors H and F_0 were calculated, and F_0 was used to calculate the initial dilatation ratio and strain tensor, as before. The shape matching terms f_1 and f_2 were set to zero. The initial distortions are visualized by the plotting the dilatation ratio, as shown in Fig. 4.7.

The finite element mesh consisted of 320 elements. The approach was applied for two iterations before convergence was obtained. The total run time of the two iterations in COMSOL was 8.4 minutes, which does not include the steps performed in Matlab. The solution was two more times obtained using 1240 elements until convergence was obtained, and using 3852 elements until convergence was obtained. Table 4.1 lists the parameters used in this test case.

Once the solutions converged, the calculated displacements were applied to update the coordinates OSS and OAS for each of the three finite element meshes. The deformation gradient F was calculated for each mesh, and then used to calculate the dilatation ratio. Fig. 4.7(b,d,f) contains plots of the dilatation ratios obtained for each of the finite element meshes after convergence. Measures of convergence and summary statistics for each iteration are listed in Table 4.3.

For the "Normal" (1240 elements) and "Finer" (1796 elements) meshes, the solution converges towards the expected answer, which is $J = 1$. The standard deviation of the dilatation ratio is reduced to 0.010 for the "Normal" mesh, and to 0.007 for the "Finer" mesh, with an average value of 0.999 for both meshes. When the 'Coarser' mesh was used, the solution met the convergence criterion, but did not do as good a job at removing the initial distortions, as seen in Fig. 4.7(b). The standard deviation of the dilatation ratio is reduced to 0.058, and the mean of the dilatation ratio is 0.999.

2.1.4 Case 4: Effect of viscosity parameter ν on the registration of a sphere to itself

YAS is a sphere of radius 1 that contained 7542 vertices and 15080 faces. Displacements were applied to YAS in spherical coordinates to generate OAS:

$$\begin{aligned} \phi &= \Phi \\ r &= R \\ \theta &= \Theta + \frac{0.1}{\pi}\sin(4\Theta)\cos^2(\Phi) \ . \end{aligned} \qquad \text{IV.6}$$

The coordinates for OSS are given by

$$y = X . \qquad \text{IV.7}$$

The tensors H and F_0 were calculated, and F_0 was used to calculate the initial dilatation ratio and strain tensor. No shape matching terms were used. The initial distortions are visualized by the determinant of the deformation gradient between OAS and YAS, which is shown in Fig. 4.8(a).

Solutions were obtained for each of the three damping coefficients. The finite element mesh consisted of 1240 elements. A list of the parameters used for this test case is found in Table 4.1. The algorithm was applied for two iterations before convergence

85

was obtained. The total run time in COMSOL was 9.3 minutes for $v = 1$, 12.7 for $v = 100$ and 14.1 minutes for $v = 10000$ excluding Matlab processing time. Measures of convergence and summary statistics for each iteration are listed in Table 4.4.

The dilatation ratio is shown in Fig. 4.8(b,d,f). For all each of the damping coefficients, the standard deviation of the dilatation ratio is reduced from 0.0645 to at most 0.001. The rate at which the total energy function changes over time depends on the coefficient of damping (Fig. 4.8c,e,g). As the coefficient of damping increases, the damping force increases, causing a decrease in the rate at which the solution converges. The damping coefficient affects the choice of the user-defined time stepping constants in COMSOL. For the case in which $v = 1$, if the time step was too large (e.g., a range of 0 to 1 in steps of 0.1), then the solver was unable to reach completion. Decreasing the range and the time step, the solver was able to obtain a solution.

2.2 Case 5: Registration of a spherical surface to a "pumpkin" shape

The fifth test case involved a sphere that expanded and folded into a shape like a pumpkin. No surface matching terms were used. YAS was a sphere of radius 1 that was described by 7542 vertices and 15080 faces. Displacements in the radial direction were applied to the reference coordinates to create the deformed surface. The initial deformation gradient was calculated and imported into COMSOL.

$$\begin{aligned} \phi &= \Phi \\ r &= 1.1R + 0.1sin(4\Theta)cos(\Phi) \\ \theta &= \Theta \end{aligned} \qquad \text{IV.8}$$

Using the function *calc_deformation_gradient*, the tensors $\boldsymbol{H}$ and $\boldsymbol{F_0}$ were calculated, and then the initial strain tensor and dilatation ratio were calculated. The finite

element mesh consisted of 1240 elements. A list of the parameters used for this test case is found in Table 4.2. The algorithm was applied for two iterations before convergence was obtained. The total run time in COMSOL was 11 minutes. Measures of convergence and summary statistics are listed in Table 4.5.

As shown in Fig. 4.9, both the spatial distribution of the strain-energy density function and the dilatation ratio show marked reduction in their variability, approaching constant values. The initial strain field, shown in Fig. 4.10, shows large principal strains in areas of high negative curvature. After registration, the magnitudes of the principal strain values are reduced, but still show some spatial variation, as the surface necessarily deforms non-uniformly to accommodate the change in shape.

2.3 Case 6: Effect of the shear modulus and bulk modulus on the registration of a spherical surface to a pumpkin surface

The same reference and deformed surfaces as the fifth case were used. The solution was obtained by solving Eq. III.49 using the finite element method three times, with the shear modulus set at 0.01 Pa, to 1 Pa, to 10 Pa, respectively. The bulk modulus was maintained at 10 Pa. The total run time in COMSOL was 11 minutes for $\mu = 0.01$ Pa, 13.2 minutes for $\mu = 1$ Pa and 17 minutes for $\mu = 10$ Pa. The parameters used for this test case are listed in Table 4.1. Measures of convergence and summary statistics for each iteration are listed in Table 4.5.

The initial distortions are visualized by the dilatation ratio, which is shown in Fig. 4.11(a). The dilatation ratio was calculated for each shear modulus used, and is shown in Fig. 4.11(b,d,f). As the shear modulus increases with respect to the bulk modulus, the

material goes from being incompressible to compressible, and the dilatation ratio changes from being uniform to varying spatially. The standard deviation of the dilatation ratio increases from 0.001 for $\mu = 0.01$ Pa, to 0.008 for $\mu = 1$ Pa, and to 0.075 for $\mu = 10$ Pa.

2.4 Case 7: Registration of a locally smoothed P14 cortical surface to an unsmoothed P14 cortical surface

For the seventh test case, OAS was a P14 cortical surface. To create YAS, a closed region was selected on OAS and smoothed for 500 iterations in CARET. The surfaces are shown in Fig. 4.12, with the selected region highlighted in red. The P14 cortical surface contained 10744 vertices and 21484 faces. A spherical representation of OAS was generated in CARET and was set to OSS.

The border between the smoothed and unsmoothed region was used to create the surface matching terms f_1 and f_2, which are shown in Fig. 4.13(a,b). The initial difference between f_1 and f_2 is small because the boundary of the region is at the same spatial coordinates on both surfaces, and decreases after the solution converged. The initial distortions are visualized in Fig. 4.14(a,c) by the strain-energy density function and the change in area from the deformed surface to the reference surface.

For this case, the finite element mesh consisted of 3852 elements. A list of the parameters used for this test case is found in Table 4.2. The algorithm was applied for three iterations before convergence was obtained. The total run time in COMSOL was 93 minutes. Measures of convergence and summary statistics for each iteration are listed in Table 4.6.

After relaxation, the strain-energy density function becomes constant inside the selected region and remains at approximately zero outside of the region. The same pattern holds true for the dilatation ratio, which approaches a constant of 1.13 within the selected region and remains at 1.00 outside of the region. The eigenvectors associated with the principal strains are shown in Fig. 4.15.

3. Discussion

In this chapter we introduce a method to register closed surfaces in 3-D based on minimization of distortion while matching surface features. Simple test cases illustrate the approach and demonstrate its efficacy.

The numerical solution of the equation of equilibrium identifies a minimum of the objective function. The current approach is not guaranteed to find a global minimum, so that initial conditions are important. For the test cases analyzed above, the initial conditions were determined numerically. When applied to actual data, the determination of initial conditions requires more thought, which will be explained in depth in the next chapter. The procedure is implemented in the COMSOL/Matlab environment. The number of vertices influences the amount of time required to run through the entire registration process. The density of the finite element mesh also affects computational requirements, specifically memory and processing time.

3.1 Calculation of curvature and the deformation gradients

Curvature and the deformation gradients F_0 and H are determined by using the surface coordinates to estimate spatial derivatives on the surface. The surface coordinates

need to be dense, specifically in regions of high curvature, in order to obtain accurate estimates of spatial derivatives. As discussed in Chapter II, section 4.2, at least 6 points are necessary to solve for the coefficients using Eq. II.19 (in practice we require that at least 8 points are used). Also, the radius of the fitting region should be smaller than the radius of curvature. If these conditions are not met then the estimation of spatial derivatives may be inaccurate.

The algorithm requires that curvature, and the deformation gradients F_0 and H be estimated at every surface coordinate. To quantify the 'goodness' of the fit, the residual error is calculated at every coordinate.

$$R^2 = \sum_{i=1}^{N} (z_i - \bar{z}_i)^2$$

$$\bar{z}_i = a_0 + a_1 x_i + a_2 y_i + a_3 x_i^2 + a_4 y_i^2 + a_5 x_i y_i$$

IV.9

where N is the number of surface nodes inside of the local fitting regions, x_i, y_i, z_i are local surface coordinates and a_0, a_1, a_2, a_3, a_4 and a_5 are the coefficients estimated using the linear least squares fitting algorithm in Matlab.

3.2 Density of the finite element mesh

As shown in Case 4, the density of the finite element mesh is important in order to obtain an accurate solution to Eq. III.49. The numeric values of curvature, the deformation gradient and the Jacobian are interpolated into COMSOL at each of the finite element mesh node points. The finite element mesh must be fine enough to resolvethe spatial variations in the values of curvature, the deformation gradient and the

90

dilatation ratio in order to accurately represent those measures in COMSOL. The elements in the finite element mesh are described by second order basis functions. So, the surface measures must be able to be described reasonably well by a second order basis function over the span any given element. This is checked visually and numerically by interpolating the surface measures from the finite element mesh back onto the original surface coordinates in Matlab.

3.3 Selection of the shear modulus, bulk modulus and coefficient of damping

The ratio of the shear modulus to the bulk modulus determines the compressibility of a material. When the bulk modulus is greater than two orders of magnitude larger than the shear modulus, the material is nearly incompressible. For a nearly incompressible material, the model we implement is like a rubber membrane sliding on a surface, whereas for a compressible material, the model is more like an elastic foam layer moving on a surface. As the ratio of the shear modulus to the bulk modulus increased from very small (<< 1) to 1, the converged solution changed from uniform areal expansion to non-uniform areal expansion.

The coefficient of damping determines the viscous force and influences the rate at which the solution converges, as shown in Case 4. Though a small damping coefficient causes the solution to converge more quickly, it may not be the best choice. If the damping coefficient is small and the initial stresses in the material are large, the solver in COMSOL may have difficulties determining a solution. The solver may also have problems if the user-defined time step is too large while the coefficient of damping is small. It is important to pay attention to the convergence of the solver in COMSOL. If

problems arise, increasing the damping coefficient and decreasing the time step are good places to start troubleshooting.

4. Conclusions

A surface registration algorithm that minimizes physical distortions during registration of brain surfaces from an individual is an important tool, which allows conservative estimation of variations in growth during development of the brain. In this paper, we present an approach that is implemented with commercial software, and use a hierarchical set of test cases to validate it. In the next chapter we describe the use of this algorithm to study regional patterns of growth during brain development in the ferret. It should be suitable also for studies involving human subjects.
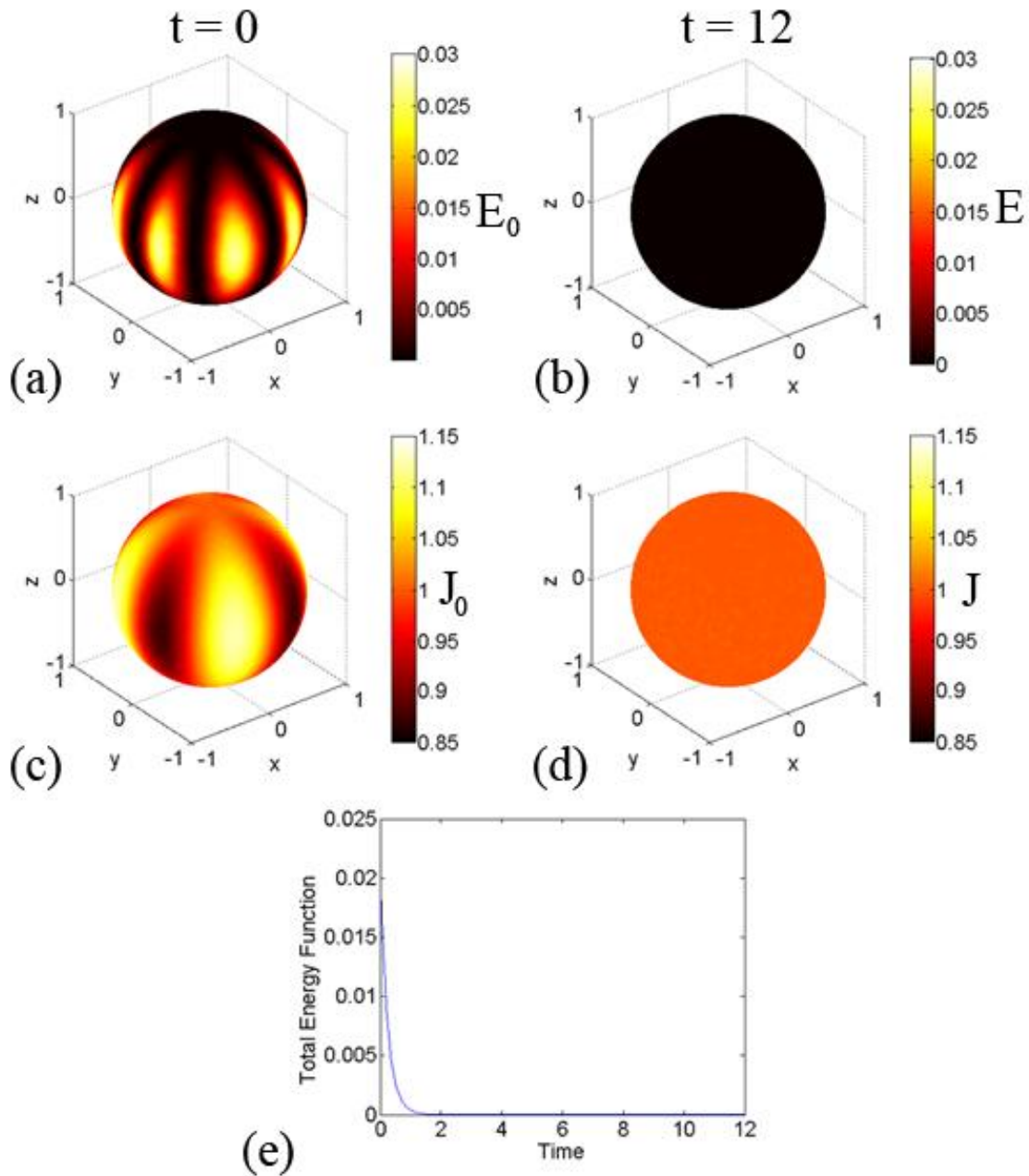
Figure 4.1: Registration of a sphere to itself: No surface matching terms are used in this case. Displacements were applied in spherical coordinates to B_0 to create B_1. Initial distortions are visualized by (a) the total energy function and (c) the change in area between B_0 and B_1. The algorithm is run until the solution converges. The initial distortions have been relaxed away. (d) The determinant of the deformation gradient

between B_1 and B_0 approaches a constant value of one while (b) the total energy function approaches zero over the entire surface. (e) The total energy function decays exponentially to zero.
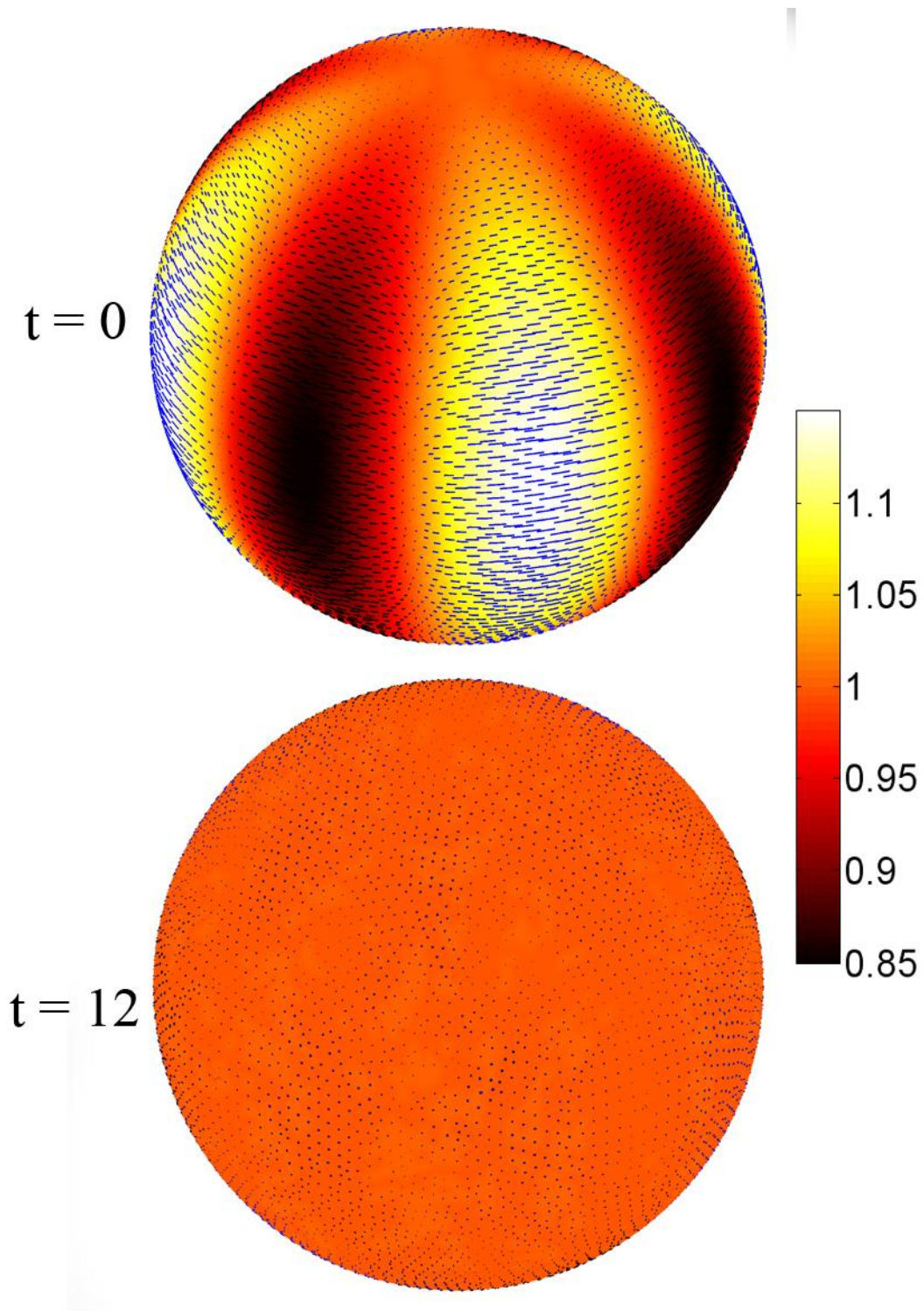
Figure 4.2: Registration of a sphere to itself: Eigenvectors associated with first (blue) and second (green) principal strains are plotted with the dilatation ratio as an underlay. The range of the dilatation ratio is from 0.85 to 1.15, as in Fig. 4.1.
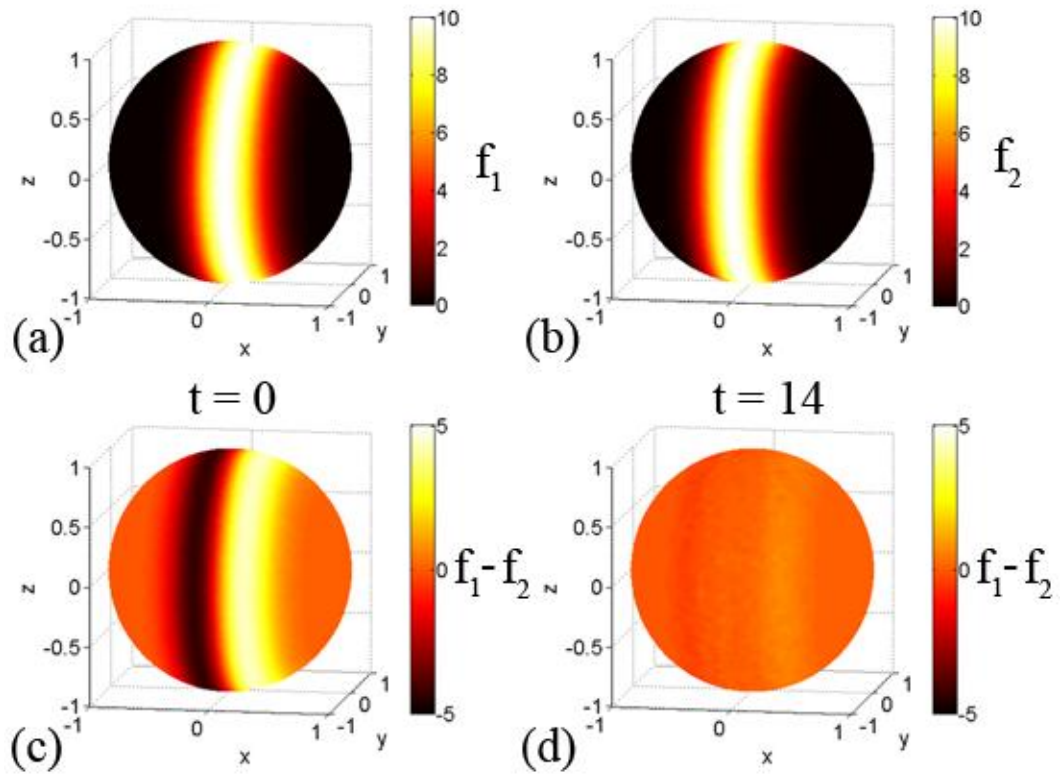
Figure 4.3: Registration of a sphere to itself with surface matching: The surface matching term (a) f_1 contains a band of high intensity on the sphere that is offset in the x-direction from (b) f_2. The initial difference between the surface matching terms is reduced by approximately one order of magnitude after the solution converges.
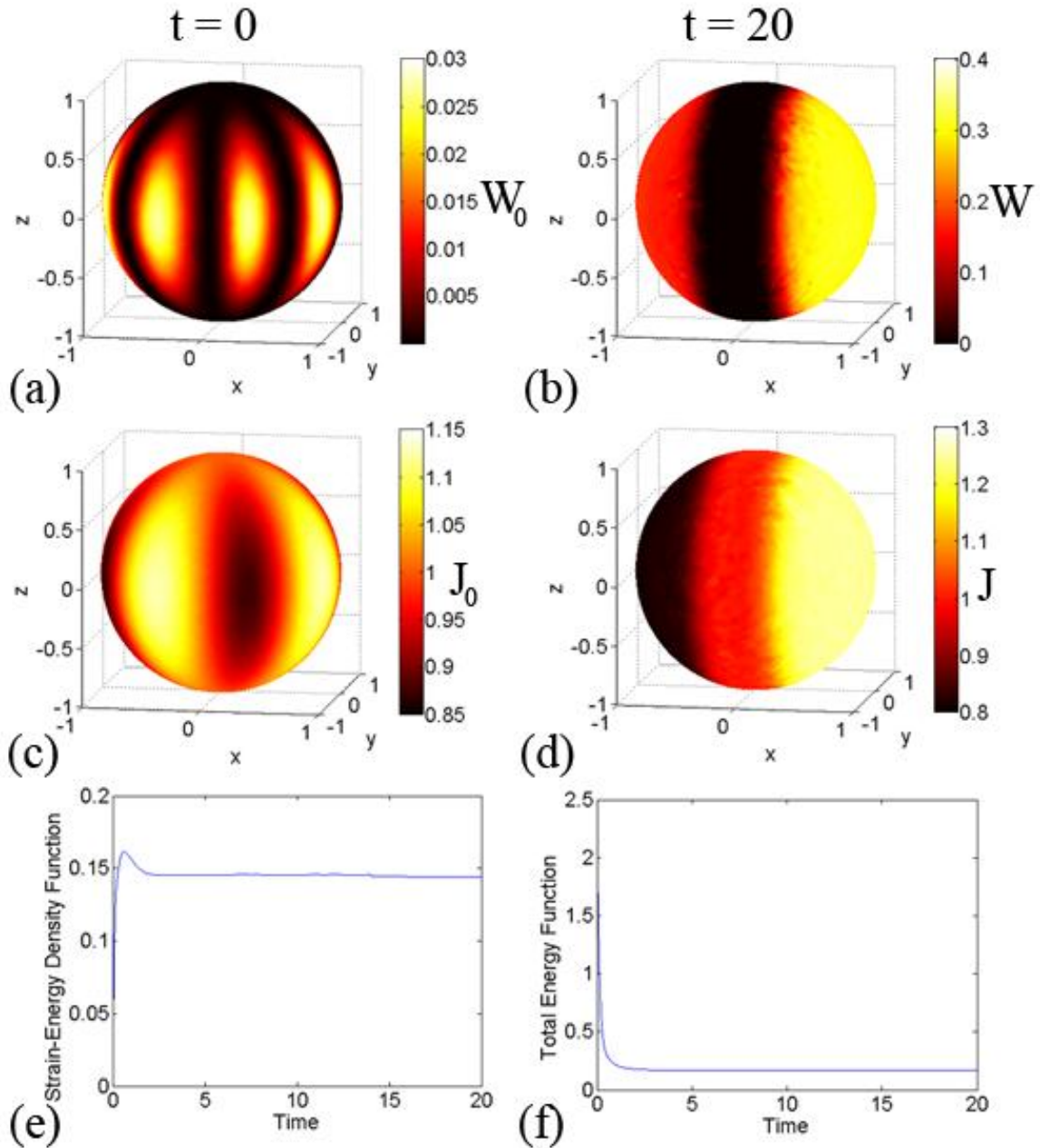
Figure 4.4: Registration of a sphere to itself with surface matching: (a,c) Initial distortions are visualized by the strain-energy density function and the change in area between B_0 and B_1. The algorithm was run until the solution converges. The initial distortions have been relaxed away. For the surface matching terms to align, the coordinates need to shift in the negative x-direction, which causes expansion on the

positive x side of the sphere and compression on the negative x side. (b,d) After relaxation, the strain-energy density function and determinant of the deformation gradient between B_0 and B_1 are separated into three regions: the positive x side of the sphere where expansion has occurred, the negative x side of the sphere where compression has occurred and the center region, which remains constant because of the matching terms.
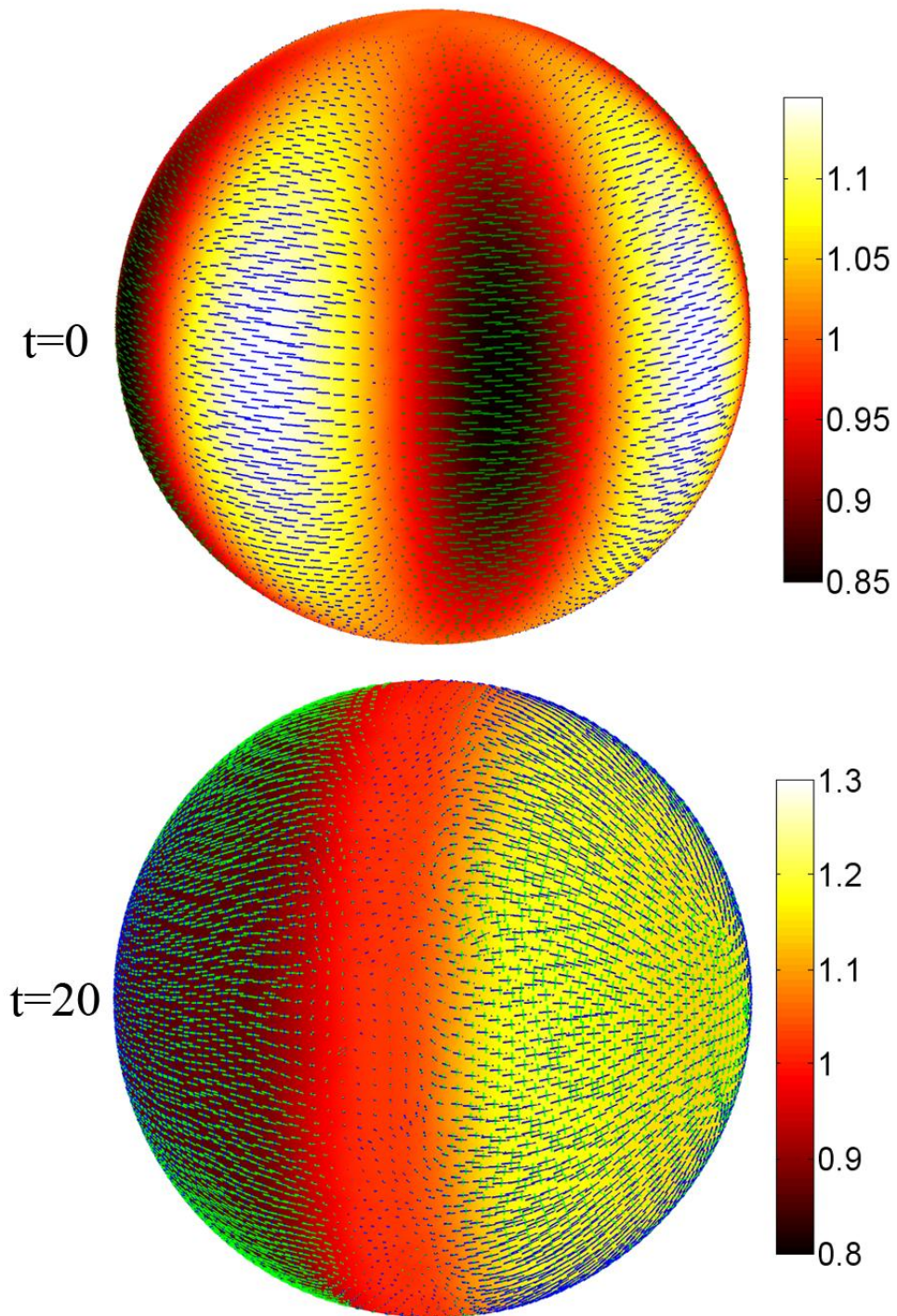
Figure 4.5: Registration of a sphere to itself with surface matching: Eigenvectors associated with first (blue) and second (green) principal strains are plotted with the dilatation ratio as an underlay.
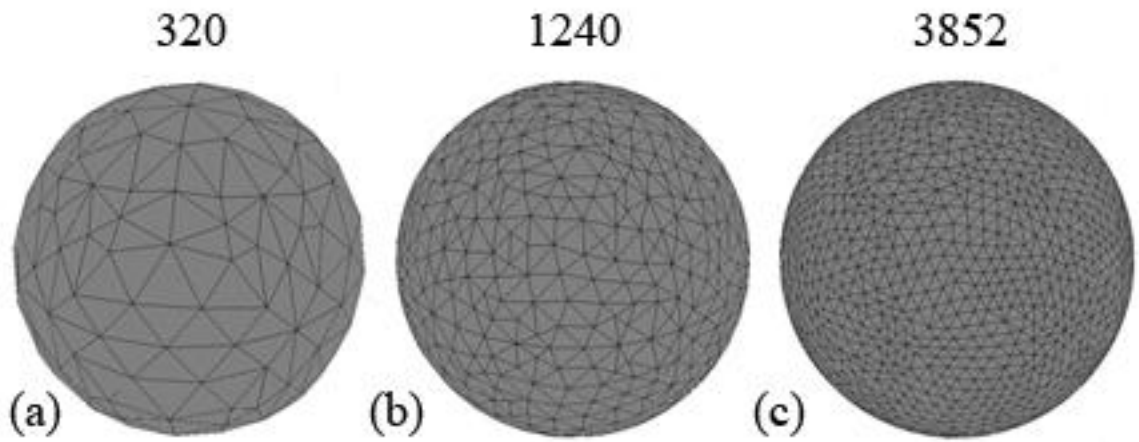
Figure 4.6: Effect of finite element mesh density on solution: (a) 'Coarser' mesh made up of 320 elements. (b) 'Normal' mesh made up of 1240 elements. (c) 'Finer' mesh made up of 3852 elements.
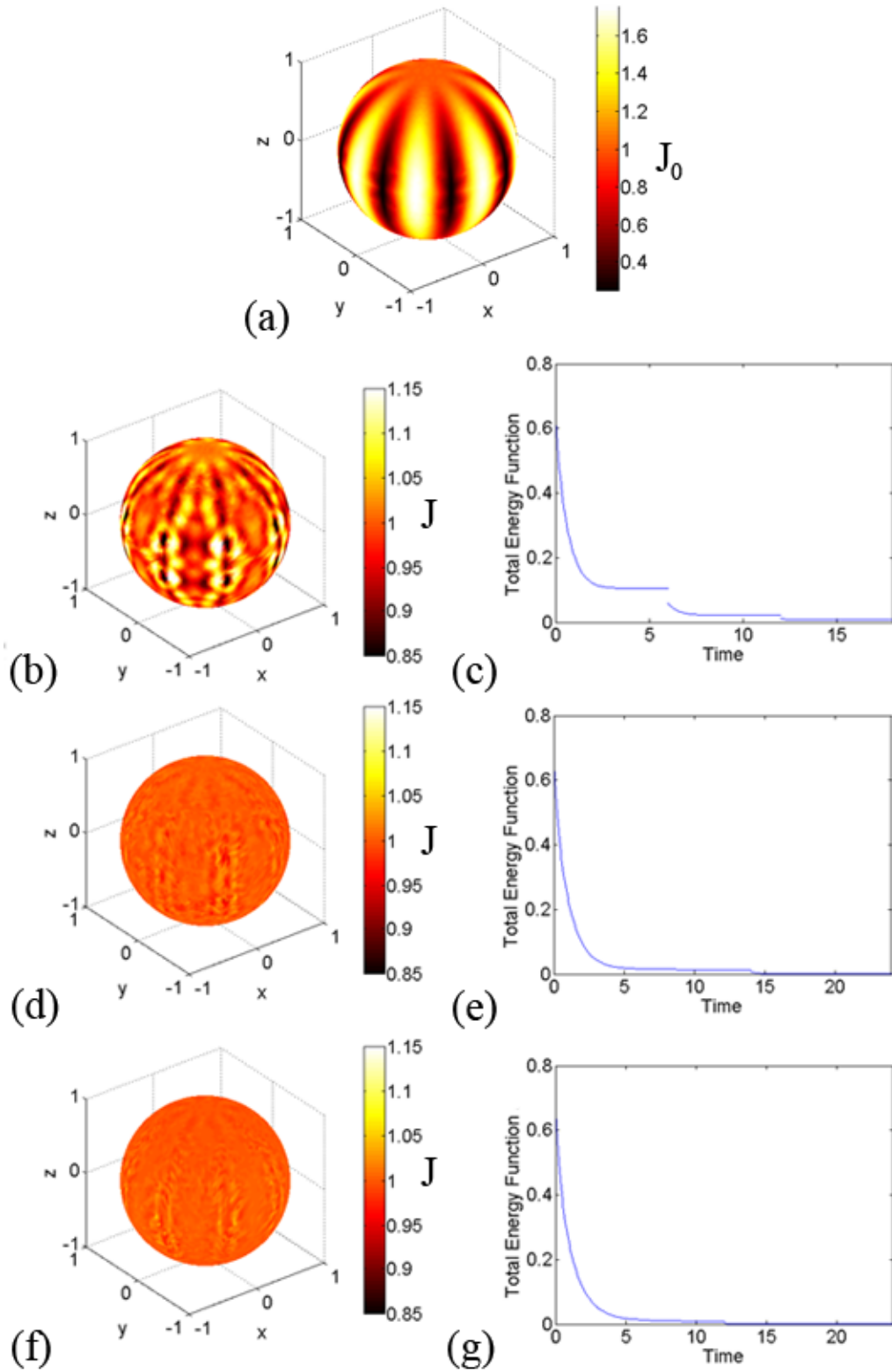
Figure 4.7: Effect of finite element mesh density on solution obtained from the registration of a spherical surface to itself: (a) Initial distortions are visualized by the

change in area between B_0 and B_1. The algorithm was run until the convergence criterion was met for three different finite element mesh densities. (b) Using 320 elements, large distortions still remain. (d,f) Distortions are reduced when a denser mesh was used. If the density of the finite element mesh is too low compared to the spatial frequency of the initial distortions, then the mesh will not be able to accurately represent and remove the distortions. (c,e,g) The total energy function decreased towards a constant for each of the three mesh densities.
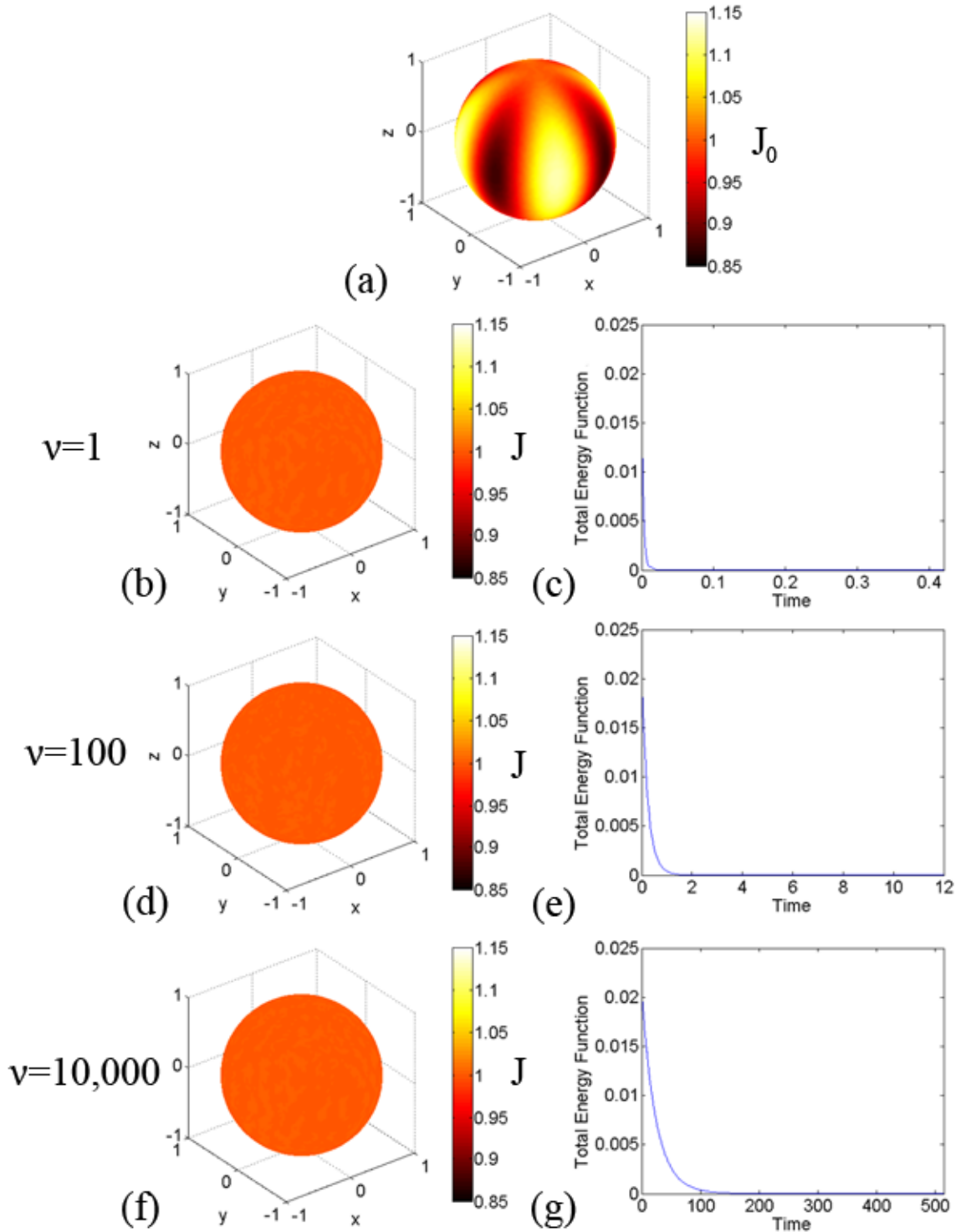
Figure 4.8: Effect of the damping coefficient on solution obtained from the registration of a spherical surface to itself: (a) Initial distortions are visualized by the change in area between B_0 and B_1. The algorithm was run until the convergence criterion was met for

each of the damping coefficients. (b,d,f) The initial distortions were removed for each of the damping coefficients. (c,e,g) The total energy function decreased towards a constant for all three damping coefficients. The rate the total energy function decayed slowed as the damping coefficient was increased. The damping coefficient is chosen based on the magnitude of the stress tensor and body force. If the damping coefficient is too low, the solver in COMSOL has a difficult time converging, and when it is high the solution needs to be run for a large number of iterations to achieve convergence.
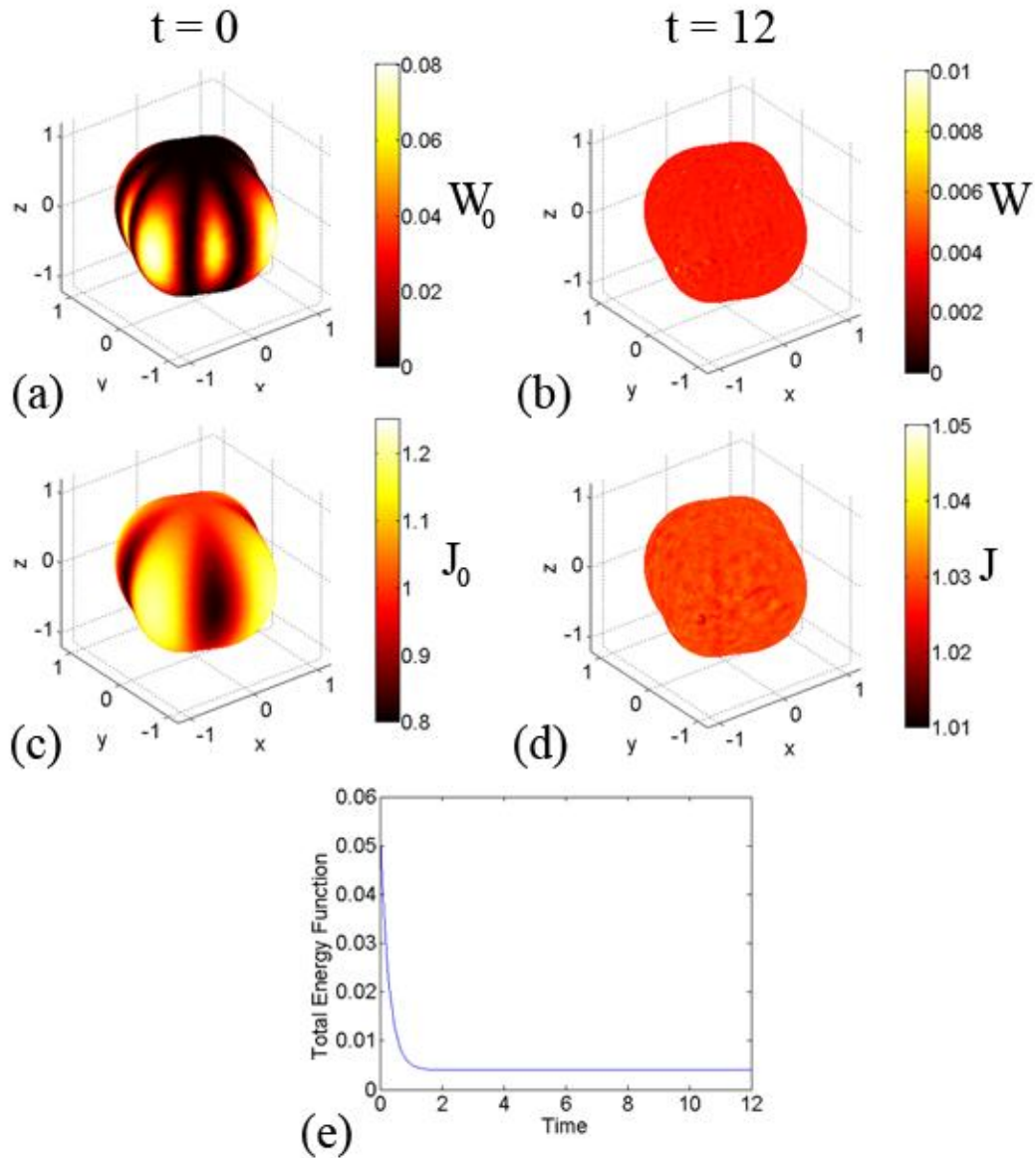
Figure 4.9: Registration of a sphere to a pumpkin: The deformed surface was created by growth and folding of the reference surface. (a,c) Initial distortions are visualized by the strain-energy density function and the determinant of the deformation gradient between B_1 and B_0. The algorithm was run until the convergence criterion was met for each of the damping coefficients. (b,d) After registration, the strain-energy density function and the determinant of the deformation gradient approached a constant. (e) The total energy function decayed to a non-zero constant.
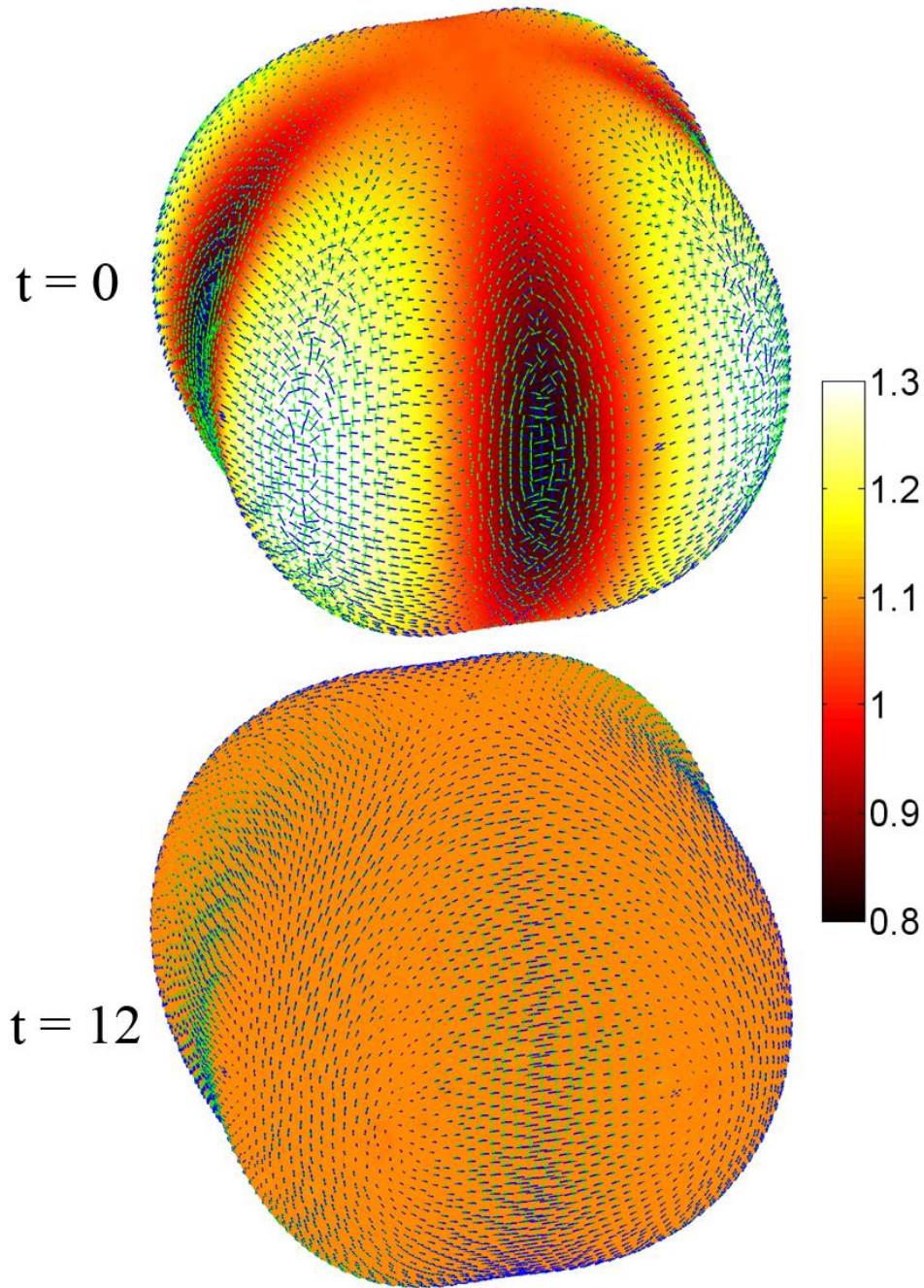
Figure 4.10: Registration of a sphere to a pumpkin: Eigenvectors associated with first (blue) and second (green) principal strains are plotted with the dilatation ratio as an underlay. Even though the dilatation ratio is uniform after relaxation, strain on the surface is non-uniform (i.e., circles drawn on the surface before relaxation becomes an ellipses whose areas are the same).
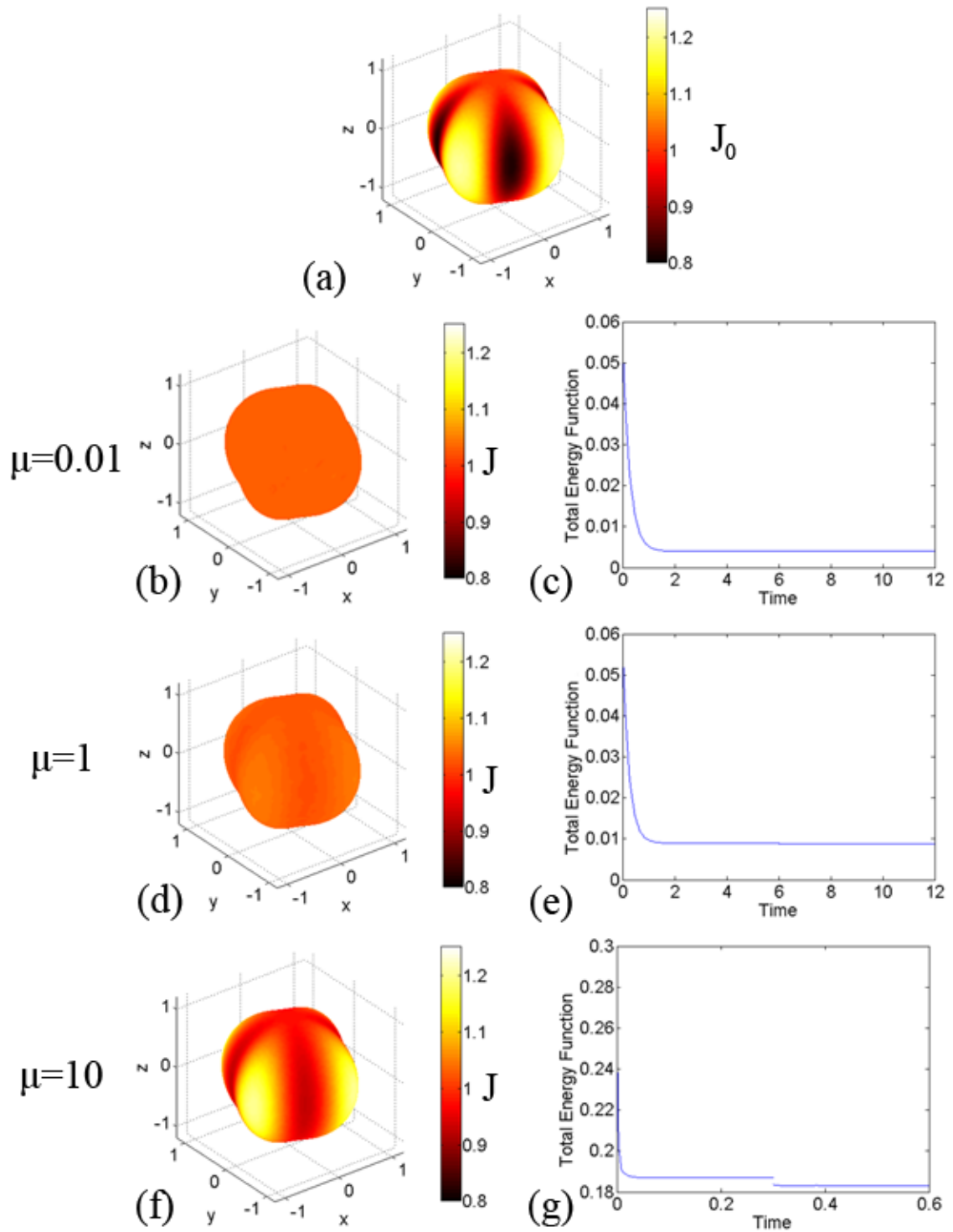
Figure 4.11: Effect of the shear modulus on the registration of a sphere to a pumpkin surface: (a) Initial distortions are visualized by the areal expansion between the YAS and OAS. The algorithm was applied until the solution converged for each value of the shear

modulus (0.1, 1, 10). The bulk modulus remained constant at 10 throughout. As the ratio of the shear modulus to the bulk modulus increases, the material goes from being nearly incompressible (e.g. a rubber-like membrane) to compressible (e.g. a foam). The equilibrium solution also changes from one of (b) uniform areal expansion to (d,f) spatially varying areal expansion.
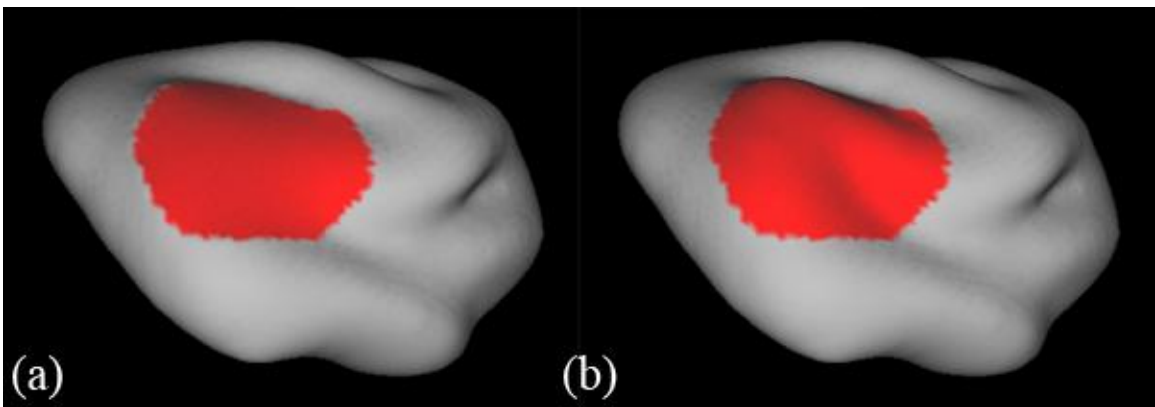
Figure 4.12 Smoothing of a local region on P14 cortical surface: The red region was identified on the P14 cortical surface of Kit A.2, and was iteratively smoothed for 500 iterations in CARET software. The smoothed region mimics a less mature cortex in the selected region. (a) The smoothed surface was set to be YAS. (b) The original surface is set to be OAS.
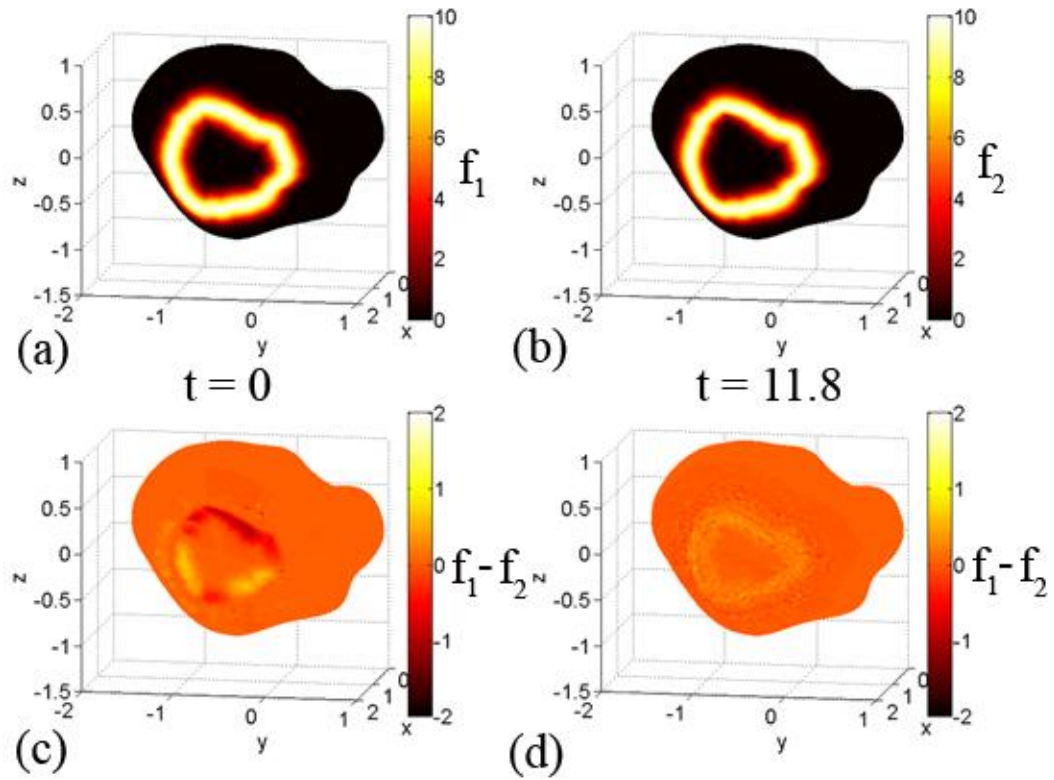
Figure 4.13: Smoothing of a local region on P14 cortical surface: (a,b) The shape matching terms f_1 and f_2 were created from the border that was used to bind the region that was smoothed. (c) The initial difference between the two matching terms was small because the landmark was at the same spatial location on each surface. (d) After applying the registration approach, the difference between the matching terms remains small.
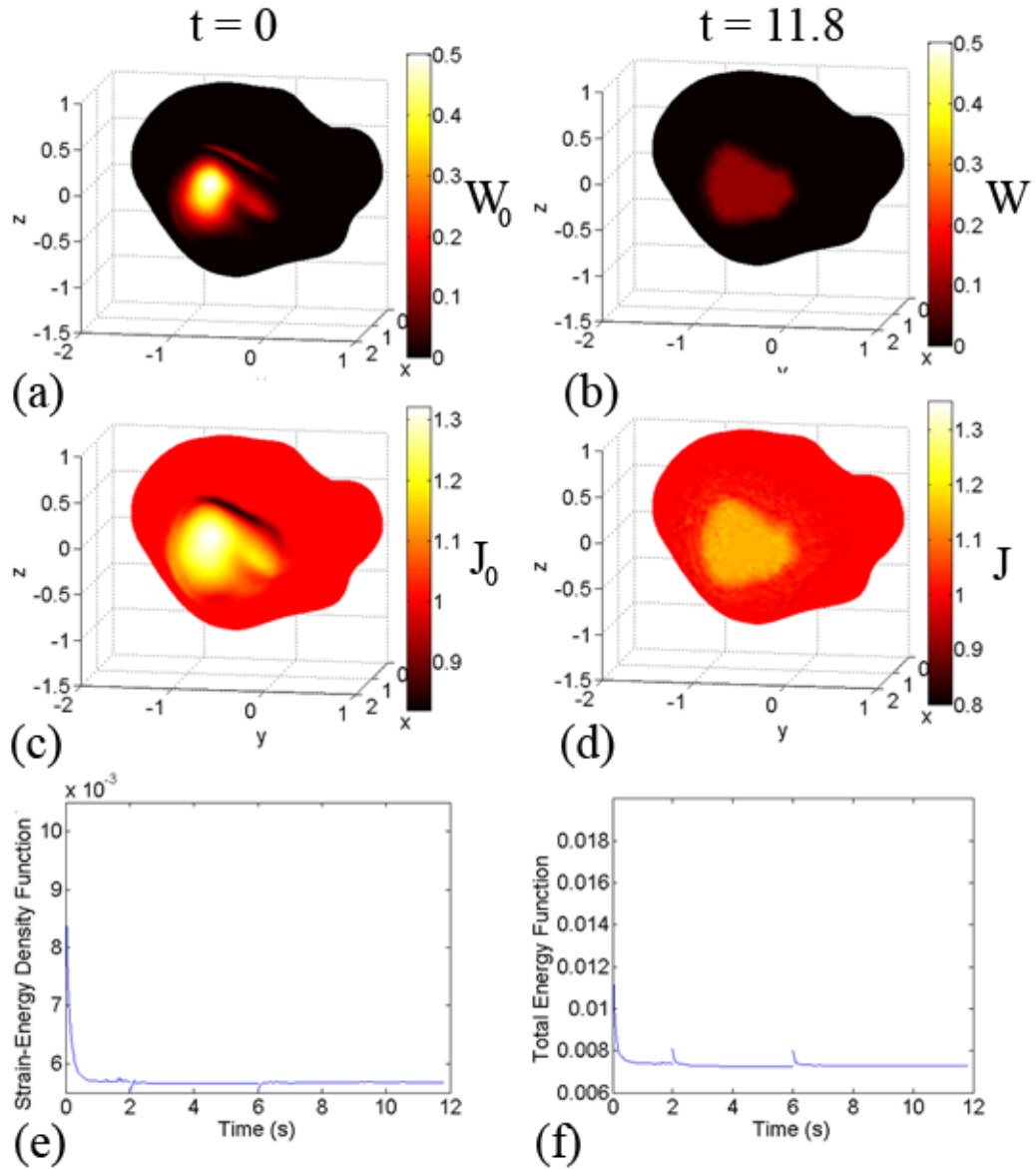
Figure 4.14: Registration of a P14 cortical surface to itself with smoothing applied to a local region: The goal of this test case is to be able to identify local growth on a complex surface. (a,c) Initial distortions are visualized by the strain-energy density function and the determinant of the deformation gradient between YAS and OAS. Only the coordinates within the patch were adjusted, so W_0 and J_0 are constant outside of the patch and vary within it. (b,d) After relaxation, the initial distortions are smoothed out within

the patch. (e,f) As the algorithm proceeds, the rate that the strain-energy density function and total energy function decreases.
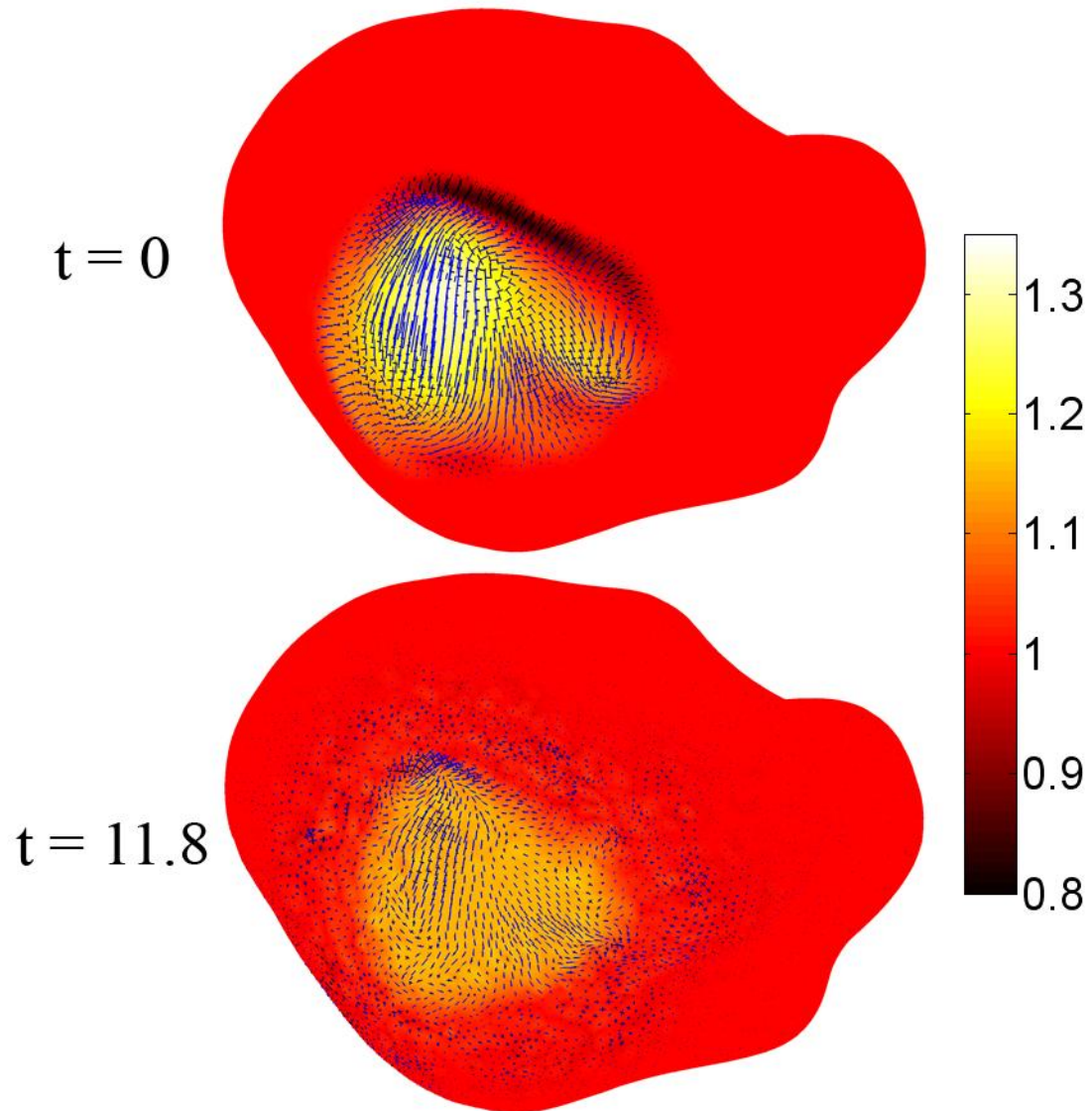
Figure 4.15: Registration of a P14 cortical surface to itself with smoothing applied to a local region: Eigenvectors associated with first (blue) and second (green) principal strains are plotted with the dilatation ratio as an underlay. The range of the dilatation ratio is from from 0.8 to 1.35 (right).

Table 4.1: List of parameters used for Cases 1-7: the shear modulus, μ, bulk modulus, κ, damping coefficient, ν, the number of elements in the finite element mesh, N, and the total run time in COMSOL.

Case	μ (Pa)	κ (Pa)	ν (N-s/m3)	N	Time (min)
1	0.01	10	100	1240	12.7
2	0.01	10	100	1240	48.7
3a	0.01	10	1000	320	8.4
3b	0.01	10	1000	1240	31.6
3c	0.01	10	1000	3852	71.9
4a	0.01	10	1	1240	9.3
4b	0.01	10	100	1240	12.7
4c	0.01	10	10000	1240	14.1
5	0.01	10	100	1234	11
6a	0.01	10	100	1234	11
6b	1	10	100	1234	13.2
6c	10	10	100	1234	17
7	0.01	10	1000	3852	93.9

Table 4.2: Results of Case 1, the registration of a sphere to itself, and Case 2, the registration of a sphere to itself with matching terms. The measures of convergence, which are the ratio of the total energy function at the end of an iteration to the beginning, Γ, and the mean magnitude of the displacement vectors, ρ, are listed for each iteration. The mean and standard deviation of J_0 and J illustrate how the initial distortions are relaxed, and are listed for Case 1. The mean and standard deviation of J were not calculated for Case 2. As the matching terms used in Case 2 align, the positive x side of the sphere is stretched while the negative x side is compressed. So, the goal is not for a uniform value of J over the surface.

Case	Iteration	Γ	ρ	mean(J_0)	std(J_0)	mean(J)	std(J)
1	1	1580	0.012	0.9999	0.0645	0.9997	0.0036
	2	12.7	$<10^{-3}$	-	-	0.9996	0.0007
2	1	13.3	0.11	NA	NA	NA	NA
	2	1.01	0.006	NA	NA	NA	NA

Table 4.3: Results of Case 3, which looks at the effect of the mesh density on the registration of a spherical surface to itself. The measures of convergence, which are the ratio of the total energy function at the end of an iteration to the beginning, Γ, and the mean magnitude of the distance vectors, ρ, are listed for each iteration. The mean and standard deviation of J_0 and J illustrate how the initial distortions are relaxed.

Case	Iteration	Γ	ρ	mean(J_0)	std(J_0)	mean(J)	std(J)
3a	1	6.1	0.029	0.994	0.3834	0.999	0.1231
	2	2.5	0.007	-	-	0.999	0.0712
	3	1.6	0.001	-	-	0.9989	0.0581
3b	1	53.1	0.035	0.994	0.3834	0.9988	0.041
	2	19.5	0.003	-	-	0.9992	0.0126
	3	3.7	$<10^{-3}$	-	-	0.9992	0.0101
3c	1	83.2	0.036	0.994	0.3834	0.9986	0.0278
	2	21.7	0.002	-	-	0.9991	0.01
	3	4.3	$<10^{-3}$	-	-	0.9992	0.0073

Table 4.4: Results of Case 4, which looks at the effect of the damping coefficient, ν, on the registration of a spherical surface to itself. The measures of convergence, which are the ratio of the total energy function at the end of an iteration to the beginning, Γ, and the mean magnitude of the distance vectors, ρ, are listed for each iteration. The mean and standard deviation of J_0 and J illustrate how the initial distortions are relaxed.

Case	Iteration	Γ	ρ	mean(J_0)	std(J_0)	mean(J)	std(J)
4a	1	770.3	0.012	0.9999	0.0645	0.9996	0.0018
	2	4	$<10^{-3}$	-	-	0.9996	0.001
4b	1	1580	0.012	0.9999	0.0645	0.9997	0.0036
	2	12.7	$<10^{-3}$	-	-	0.9996	0.0007
4c	1	699.1	0.012	0.9999	0.0645	0.9996	0.0019
	2	3.7	$<10^{-3}$	-	-	0.9996	0.001

Table 4.5: Results of Case 5, which is the registration of a sphere to a pumpkin shape, and Case 6, which looks at the effect of the ratio of the shear modulus, μ, to the bulk modulus, κ, for the registration of a sphere to a pumpkin shape. The measures of convergence, which are the ratio of the total energy function at the end of an iteration to the beginning, Γ, and the mean magnitude of the distance vectors, ρ, are listed for each iteration. The mean and standard deviation of J_0 and J illustrate how the initial distortions are relaxed.

Case	Iteration	Γ	ρ	mean(J_0)	std(J_0)	mean(J)	std(J)
5	1	13.9	0.022	1.0285	0.101	1.0285	0.0019
	2	1.003	$<10^{-3}$	-	-	1.0285	0.0011
6a	1	13.9	0.022	1.0285	0.101	1.0285	0.0019
	2	1.003	$<10^{-3}$	-	-	1.0285	0.0011
6b	1	6.4	0.02	1.0285	0.101	1.0285	0.0075
	2	1.001	$<10^{-3}$	-	-	1.0284	0.0084
6c	1	1.5	0.008	1.0285	0.101	1.0286	0.0747
	2	1.003	0.001	-	-	1.0285	0.0746

Table 4.6: Results of Case 7, which is the registration of a P14 cortical surface to itself with a locally smoothed region. The measures of convergence, which are the ratio of the total energy function at the end of an iteration to the beginning, Γ, and the mean magnitude of the distance vectors, ρ, are listed for each iteration.

Case	Iteration	Γ	ρ
7	1	2.3	0.002
	2	1.1	0.001
	3	1.1	$<10^{-3}$

Chapter V. Quantification of growth in the ferret brain during development

1. Introduction

The ferret is an excellent animal model in which to study the developing brain because its cortex folds post-natally over a period of less than four weeks, it fits into bore of the small animal MRI scanners at Washington University and it is less expensive and more convenient than primate models. In this chapter I describe the serial MRI acquisition of *in vivo* anatomical images from three ferret kits at one week intervals. The image volumes were segmented to create cortical surfaces. Curvature and sulcal depth were calculated and used to determine global measures of surface shape. Cortical surfaces from two kits were registered using the approach described in Chapters III and IV, and their growth patterns analyzed using the approach described in Chapter II.

2. Methods

2.1 In vivo image acquisition

A ferret litter was obtained from the commercial vendor Marshal Bioresources (North Rose, NY). The litter contained a mother and 4 female kits. The kits were delivered at postnatal day 5 (P5) to a dedicated animal facility at Washington University (WU). The ferrets were stored at the animal facility for the duration of the study. The kits were tattooed, each on a different paw, to distinguish among them.

Two of the kits (Kit A.1 and Kit A.2) were chosen randomly to be scanned serially by MRI. The kits were imaged at days 7, 14, 21 and 28 of life. Each kit was initially anesthetized using 3.5% isoflurane in O_2 in a vented anesthesia chamber. The ferret was then placed in a nose cone with a pallet bar or tooth bar, depending on the age

of the kit. Anesthesia was maintained through the nose cone, starting at 3.5% isoflurane and then decreasing by steps of 0.5% every 15 minutes to 1.5% isoflurane. Anesthesia was then held constant at 1.5% for the duration of the experiment. The oxygen level was held constant at 1.0 L/min during the entire experiment.

The animal's head was kept still in the prone position using a custom-made head support (see Fig. 5.1). The kit's pulse rate and oxygen saturation levels were monitored continuously by a MRI-compatible pulse-oximeter (Nonin Medical, Plymouth, MN) taped to one of the back paws. Pulse rate and oxygen saturation levels were recorded at 15 minute intervals, along with the percentage of isoflurane. All procedures were performed in accordance with NIH and institutional guidelines for the care and use of animals, and approved by the WU Animal Studies Committee. Body temperature was maintained by flowing temperature controlled water through tubing underneath the animal. The kit was kept anesthetized for a total of 120-180 minutes.

Images were acquired by an 11.7T small animal scanner with Varian INOVA consol equipped with a separate transmit and receive coil. A smaller receive coil was used at P7 and P14, while a larger receive coil was used at P21 and P28. T_2-weighted images were acquired using a standard spin-echo pulse sequence. The imaging parameters TE (echo time) and TR (repetition time) were chosen to maximize signal to noise and contrast to noise in the images at each age (Barnette et al., 2009). Images were acquired at a resolution of 250 μm isotropic, which provided sufficient signal to noise while still allowing for the structure of the cortex to be identified. The number of averages acquired depended on the amount of time available during each scan and the

length of time for a single acquisition. The larger the brain, the most time needed for a single acquisition. The parameters used in the experiments are listed in Table 5.1.

In addition to Kits A.1 and A.2, a set of images acquired by Dr. Alan Barnette from a single kit, referred to as Kit 7.2, are analyzed. The MRI volumes for Kit 7.2 were acquired at the same time points, using the same imaging parameters and the same protocol as for Kits A.1 and A.2. Kit 7.2 was part of a different study published by Barnette et al. (2009).

2.3 Segmentation of images and generation of surfaces

All operations in this section were performed using CARET software (Van Essen et al., 2001, http://brainvis.wustl.edu/wiki/index.php/Caret:About). To generate mesh-based representations of the cortical surface, the cortex must be segmented. All images were segmented manually. The pial surface, the boundary between the gray matter (GM) and cerebrospinal fluid (CSF), was chosen to be the edge of the segmentation because it is identifiable on all of the images. Ideally, both a pial surface and a surface at the boundary of the white matter (WM) and GM would be identified and used to create a mid-cortical surface. Unfortunately, the contrast between GM and WM was not strong enough to create an accurate segmentation volume at all of the time points. Hence, the segmentation volume was eroded by one voxel so that the boundary of the segmentation was inside of the cortex, giving an approximation of a mid-cortical surface. A single slice from the segmentation volume of the right hemisphere from each of the image acquisitions is shown in Fig. 5.2.

A mesh representation of the cortical surface was generated from the segmentation volume with a setting selected to allow for the maximum number of vertices and faces in CARET. A smoothing filter was then applied to the surfaces. The cortical surfaces from both hemispheres are shown in Fig. 5.3. The surfaces consisted of approximately 5000-30000 points in space connected by a triangular mesh which contained approximately 10000-60000 faces. "Inflated", "Very Inflated" and "Spherical" representations of the cortical surface were created as well. Using an algorithm in CARET, linear and areal distortions were reduced between the anatomical and spherical surfaces (Van Essen et al., 2001). The default settings were applied.

The medial wall was manually identified using both the anatomical MRI and the cortical surface. The coordinates that lie on the medial wall are not part of the cortex, and were not of interest to our analysis. All of the coordinates within the boundary of the medial wall were selected and removed from the surface. Calculations of surface area, curvature and sulcal depth were performed on surfaces after the medial wall was removed. Surface registration was performed with the medial wall intact, but the boundary was respected.

2.4 Calculation of surface area, curvature and sulcal depth

The area of a cortical surface is calculated by summing the area of each of the triangular faces on the surface. For the ith face on the surface with points x_1^i, x_2^i, x_3^i, the surface area of that face is given by

$$A_i = \frac{1}{2}\left(v_1^i \times v_2^i\right) \cdot \left(\frac{v_1^i \times v_2^i}{\left|v_1^i \times v_2^i\right|}\right),$$
V.1

where

$$v_1^i = x_2^i - x_1^i$$
$$v_2^i = x_3^i - x_1^i \ .$$

V.2

So, the total surface area is given by

$$A = \sum_{i=1}^{M} A_i \ .$$

V.3

Local estimates of principal curvature were calculated at each vertex of the mesh using the custom function *calc_curvature*. The implementation is described in more detail in Chapter II, section 2. Mean curvature, K, is given by

$$K = \frac{1}{2}(\kappa_1 + \kappa_2) \ ,$$

V.4

where κ_1 and κ_2 are the first and second principal curvatures, respectively. Estimates of mean curvature for each of the surfaces are shown in Fig. 5.4.

Sulcal depth, Δ, is a measure of the distance from surface coordinates on the cortical surface to those on a convex hull, which are calculated using CARET software (Van Essen et al., 2001). Fig. 5.5 shows a cortical surface and its corresponding convex hull with sulcal depth values plotted on it.

K and Δ provide local measures of shape on the surface, but can also be used to generate global measures of shape. A variety of global measures of shape have been defined previously based on mean curvature, Gaussian curvature, individual principal curvatures and surface area (Batchelor et al., 2002; Magnotta et al., 1999; Rodriguez-Carranza et al., 2008; Van Essen and Drury, 1997). Two pieces of information, amplitude and frequency, are required to accurately characterize a sine wave. Similarly, we use an average of sulcal depth (amplitude) and an average of mean curvature (frequency) to provide a global statistical description of cortical shape.

The average of sulcal depth, $\bar{\Delta}$, is calculated by integrating sulcal depth over the surface and dividing by the surface area:

$$\bar{\Delta} = \frac{\int_A |\Delta| dA}{A} \quad , \qquad\qquad \text{V.5}$$

As noted in Rodriguez-Carranza et al. (2008), difference in surface area must be accounted taken into account. The example they give is that two spheres with different radii will have different global measures. One method to account for differences in surface area is to uniformly scale all surfaces to have the same area as a sphere of radius one. Mean curvature is then calculated on the scaled surfaces, and the average of mean curvature, $\bar{K}$, is calculated using the same idea as Eq. V.5:

$$\bar{K} = \frac{\int_A |K| dA}{A} \qquad\qquad \text{V.6}$$

In practice $\bar{K}$ and $\bar{\Delta}$ are calculated using a 2-D form of the rectangle rule for integration, which is given by

$$\bar{K} = \frac{\sum_{i=1}^{N} |H_i| dA_i}{A}$$
$$\bar{\Delta} = \frac{\sum_{i=1}^{N} |\Delta_i| dA_i}{A} \quad , \qquad\qquad \text{V.7}$$

where N is the number of surface coordinates, i is the ith surface coordinate and dA_i is a differential area element given by

$$dA_i = \frac{1}{3M} \sum_{j=1}^{M} A_j^f . \qquad\qquad \text{V.8}$$

Here M is the number of faces at the ith surface coordinate, and A_j^f is the area of the jth face of the surface.

2.5 LACROSS Registration Approach

The LACROSS (LAndmark Correspondence and Relaxation Of Surface Strain) registration approach was applied to determine a one-to-one correspondence between the P14 and P21 cortical surfaces (right hemisphere, Kits A.1 and A.2) and the P21 and P28 cortical surfaces (right hemisphere, Kits A.1 and A.2).

2.5.1 Determination of an initial correspondence

An initial correspondence between the surfaces that is in the neighborhood of the final solution must be determined in order to apply the LACROSS registration approach. We use a series of a set of landmark points on both the younger anatomical surface (YAS) and the older anatomical surface (OAS) to determine the initial correspondence. Ideally, landmark points would be identified by vasculature or some identifiable biological landmark. However, in their absence, mean curvature guides the selection of the landmark points. While the landmark points are matched exactly for the initial correspondence, they are not forced to remain exactly aligned when the LACROSS approach is applied.

Approximately 20 landmark points are selected on both the YAS and OAS, the points are projected to the younger spherical surface (YSS) and older spherical surface (OSS), respectively. Triangular meshes are generated from the projected landmark points that contain 20 vertices and 36 faces (Fig. 5.6). Using the custom function *refine_mesh*, the landmark meshes are refined so that they contain a larger number of vertices and faces. The function works by calculating the midpoint on each line segment, turning each face into four new faces. This process is repeated multiple times until the mesh contains

approximately 4900 vertices and 9700 faces. The function *refine_mesh* is provided in Appendix, section 2.6. The younger and older dense landmark meshes act as a blueprint to describe YAS using the surface points of OAS (Fig. 5.7).

2.5.2 Calculation of f_1 and f_2

The functions f_1 and f_2 determine the body force $\boldsymbol{g}$, as described in Eq. III.42, and can be any function of $\boldsymbol{X}$ and $\boldsymbol{x}$, respectively. We want regions of similar mean curvature values to be drawn together, and also want for fixed boundaries to be respected. So, f_1 and f_2 are the sum of two terms: a function of mean curvature, f_c, and a function of distance from manually identified landmarks, f_d:

$$f_1 = f_{1,c} + f_{1,d}$$
$$f_2 = f_{2,c} + f_{2,d} \quad .$$

<div align="right">V.9</div>

Let K_Y and K_O be the mean curvature estimates on YAS and OAS, respectively. A threshold of $\pm 0.5 \ mm^{-1}$ was applied, which forced the maximum and minimum values to be the same on both surfaces. The matching terms due to curvature are given by

$$f_{1,c} = K_Y$$
$$f_{2,c} = K_O$$

<div align="right">V.10</div>

The second term is a function of the geodesic distance of the surface coordinates from the landmark points. Certain regions, such as the boundary of the medial wall, are not readily identified using mean curvature. A different approach must be used in order to ensure that this boundary is respected. First, using CARET, the boundary of the region is identified manually by landmark points. Then the geodesic distance is calculated between every surface coordinate and the nearest landmark point to each coordinate. The goal of

this term is to have a high value when the geodesic distance is small, and the value should

decrease to zero as the distance becomes larger, which is achieved by

$$f_{1,d} = e^{-\frac{d_1^2}{a_1}}$$

$$f_{2,d} = e^{-\frac{d_2^2}{a_2}} \, ,$$

V.11

where a_1 and a_2 are constants, d_1 is the geodesic distance from coordinate X to the

nearest landmark point on the YAS and d_2 is the geodesic distance from coordinate x to

the nearest landmark point on the OAS.

Using a custom function in Matlab, both f_1 and f_2 were iteratively smoothed over

the surface. Smoothing was achieved by computing a weighted average of the mean

curvature at a node with the mean curvature of each of that node's neighbors for each

node on the surface. The code for the smoothing function is provided in the Appendix,

section 2.7.

3. Results

3.1 Surface area, curvature and sulcal depth

Plots of surface area, $\bar{\Delta}$ and $\bar{K}$ over time are shown in Fig. 5.7. Surface area

increases steadily from P7 to P28 by approximately a factor of five. The rate of increase

from P14 to P28 is slightly higher than from P7 to P14. No differences are seen between

the three kits or between the hemispheres. The average of mean curvature increases from

P7 to P28 by approximately a factor of three. The rate of increase over time decreases

with age, and appears to be approaching a constant. However, measurements at later time

points are necessary to discern for certain. Average sulcal depth increases steadily from

P7 to P28. These measures agree with observation that after the first month of life folding

has mainly completed while the brain continues to grow in size. No differences are seen between the hemispheres for both the average of sulcal depth and mean curvature. Some small differences appear to be present between the two siblings (Kits A.1 and A.2) and Kit 7.2.

3.2 Estimation of growth

3.2.1 Registration of P14 to P21 cortical surface: Right hemisphere, Kit A.2

The surface matching functions for P14 and P21 are shown in Fig. 5.8(a-d). An initial correspondence was determined between the surfaces. Fig. 5.9(a,b) shows the dilatation ratio from the initial correspondence. After relaxation, many of the initial distortions are removed (Fig. 5.9c,d) and the surface matching functions become more aligned (Fig. 5.8e-h). Plotting the areal expansion using a more appropriate value range provides insight into local growth between P14 and P21 (Fig. 5.10). The average areal expansion is 1.87 with a standard deviation of 0.22. Values of the dilatation ratio from the medial wall were not included when computing the average and standard deviation. Lower growth is seen in the allocortex (approximately no growth). The largest amount of growth is seen around the CS/SPS sulcus, which lies on the medial side of the brain.

Using CARET software, a number of regions (based on geometry) were selected on the P14 surface and mapped to the P21 surface (Fig. 5.11). This image confirms that the registration makes sense (i.e., regions are matching up fairly well between the time points). The eigenvectors that correspond to the first and second principal strains are shown in Fig. 5.13. The parameters used for registration are listed in Table 5.2. The "Extra Fine" mesh was used, which consists of 9068 elements. The total run time in

COMSOL was 4.1 hours. The combination of both pre- and post-processing time in Matlab was on the same order of time, though not meticulously tracked.

3.2.2 Registration of P21 to P28 cortical surface: Right hemisphere, Kit A.2

The surface matching functions for P21 and P28 are shown in Fig. 5.14(a-d). An initial correspondence was determined between the surfaces. The dilatation ratio before relaxation is shown in Fig. 5.15(a,b). After relaxation, many of the initial distortions are removed (Fig. 5.14c,d) and the surface matching functions become more aligned (Fig. 5.14e-h). Plotting the dilatation ratio over a smaller range allows for smaller differences to be seen (Fig. 5.16). The average amount of areal expansion is less than from P14 to P21 (1.45), and the standard deviation is much lower (0.085). Values of the dilatation ratio from the medial wall were not included when computing the average and standard deviation.

Using CARET software, a number of regions (based on geometry) were selected on the P21 surface and mapped to the P28 surface (Fig. 5.17). The eigenvectors that correspond to the first and second principal strains are shown in Fig. 5.18. The parameters used for registration are listed in Table 5.2. The "Extra Fine" mesh was used, which consists of 9068 elements. The total run time in COMSOL was 5.4 hours. Both pre- and post-processing time in Matlab was on the same order of time.

3.2.3 Registration of P14 to P21 cortical surface: Right hemisphere, Kit A.1

The surface matching functions for P14 and P21 are shown in Fig. 5.19(a-d). An initial correspondence was determined between the surfaces. Fig. 5.20(a,b) shows the

dilatation ratio from the initial correspondence. After relaxation, many of the initial distortions are removed (Fig. 5.20c,d) and the surface matching functions become more aligned (Fig. 5.19e-h). Plotting the areal expansion using a more appropriate value range provides insight into local growth between P14 and P21 (Fig. 5.21). The average areal expansion is 1.85 with a standard deviation of 0.22. Values of the dilatation ratio from the medial wall were not included when computing the average and standard deviation. Lower growth is seen in the allocortex (approximately no growth). The largest amount of growth is seen around the CS/SPS sulcus, which lies on the medial side of the brain.

The eigenvectors that correspond to the first and second principal strains are shown in Fig. 5.22. The parameters used for registration are listed in Table 5.2. The "Extra Fine" mesh was used, which consists of 9068 elements. The total run time in COMSOL was 5.4 hours. The combination of both pre- and post-processing time in Matlab was on the same order of time, though not meticulously tracked.

3.2.4 Registration of P21 to P28 cortical surface: Right hemisphere, Kit A.1

The surface matching functions for P21 and P28 are shown in Fig. 5.23(a-d). An initial correspondence was determined between the surfaces. The dilatation ratio before relaxation is shown in Fig. 5.24(a,b). After relaxation, many of the initial distortions are removed (Fig. 5.24c,d) and the surface matching functions become more aligned (Fig. 5.23e-h). Plotting the dilatation ratio over a smaller range allows for smaller differences to be seen (Fig. 5.25). The average amount of areal expansion is less than from P14 to P21 (1.48), and the standard deviation is decreased (0.12). Values of the dilatation ratio

from the medial wall were not included when computing the average and standard deviation.

The eigenvectors that correspond to the first and second principal strains are shown in Fig. 5.26. The parameters used for registration are listed in Table 5.2. The "Extra Fine" mesh was used, which consists of 9068 elements. The total run time in COMSOL was 4.1 hours. Both pre- and post-processing time in Matlab was on the same order of time.

4. Discussion

4.1 Initial correspondence for registration

An initial correspondence between surfaces is required to apply the LACROSS algorithm. We used a series of manually selected landmark points on both the YAS and OAS to guide the initial registration. While using mean curvature to drive the initial correspondence would simplify the overall process from a user's standpoint, we were unable to derive and implement a completely satisfactory scheme. However, the use of landmarks to guide the initial correspondence makes sense with the future goal of identifying fixed biologically relevant landmarks that can be tracked over time (e.g., vasculature).

4.2 Surface matching terms

The surface matching terms are functions of mean curvature and distance from the medial wall. Mean curvature is an attractive measure of surface shape that is continuous and smooth over the cortical surfaces. The feature that curvature identifies on the surface

of the brain is whether a surface point lies on a gyrus, sulcus or somewhere else. By selecting mean curvature as a driving force for registration, we are setting a preference for surface points to remain in similar regions of mean curvature over time.

In order to ensure that the maximum and minimum values of the matching term were the same for both the YAS and OAS, a low threshold was applied to the calculated values of mean curvature. A smoothing filter was applied after thresholding so that the matching terms were smooth and that the base of the sulci and crown of the gyri had the largest magnitude.

4.3 Growth results

The average growth between P14 to P21 is greater than from P21 to P28 , and from P14 to P21, the regional variation in growth is higher. Between both sets of time points, very little growth occurs in the allocortex (inferior to the RhS sulcus and part of the SS/SPS sulcus. The allocortex develops earlier than the rest of the cortex, identified by low FA values earlier in development (Kroenke et al., 2009). From P14 to P21, the region of largest growth is seen around the CS/SpS sulcus. This sulcus contains a very smooth region at P14 that has developed into a deep fold by P21. As shown by the major eigenvector of the strain tensor, the maximum principal direction of growth generally occurs across gyri. Between P21 and P28, growth is closer to regionally uniform over the surface.

5. Conclusions

We acquired a series of anatomical MRI volumes from three ferret kits at different stages of development. Using previously developed methods, the images were segmented, and the segmented images were used to generate triangular mesh representations of the cortical surface. Surface area, curvature and sulcal depth were calculated and compared as a function of postnatal age. Using the approach described in Chapters III and IV, three of the cortical surfaces from a single kit were registered to one another. The deformation gradient tensor was estimated between the registered surfaces, the determinant of which provided a measure of growth. Regional and temporal variations in growth were seen in the cortex during this period.

Figure 5.1: MRI-compatible device to hold the head of the ferret still during an experiment. A mixture of isoflourane and oxygen is supplied through the nose cone. A receive surface-coil is placed above the head. The transmit coil (not shown) surrounds the entire setup when placed in the magnet. Image reproduced with permission of author (Barnette et al., 2009).
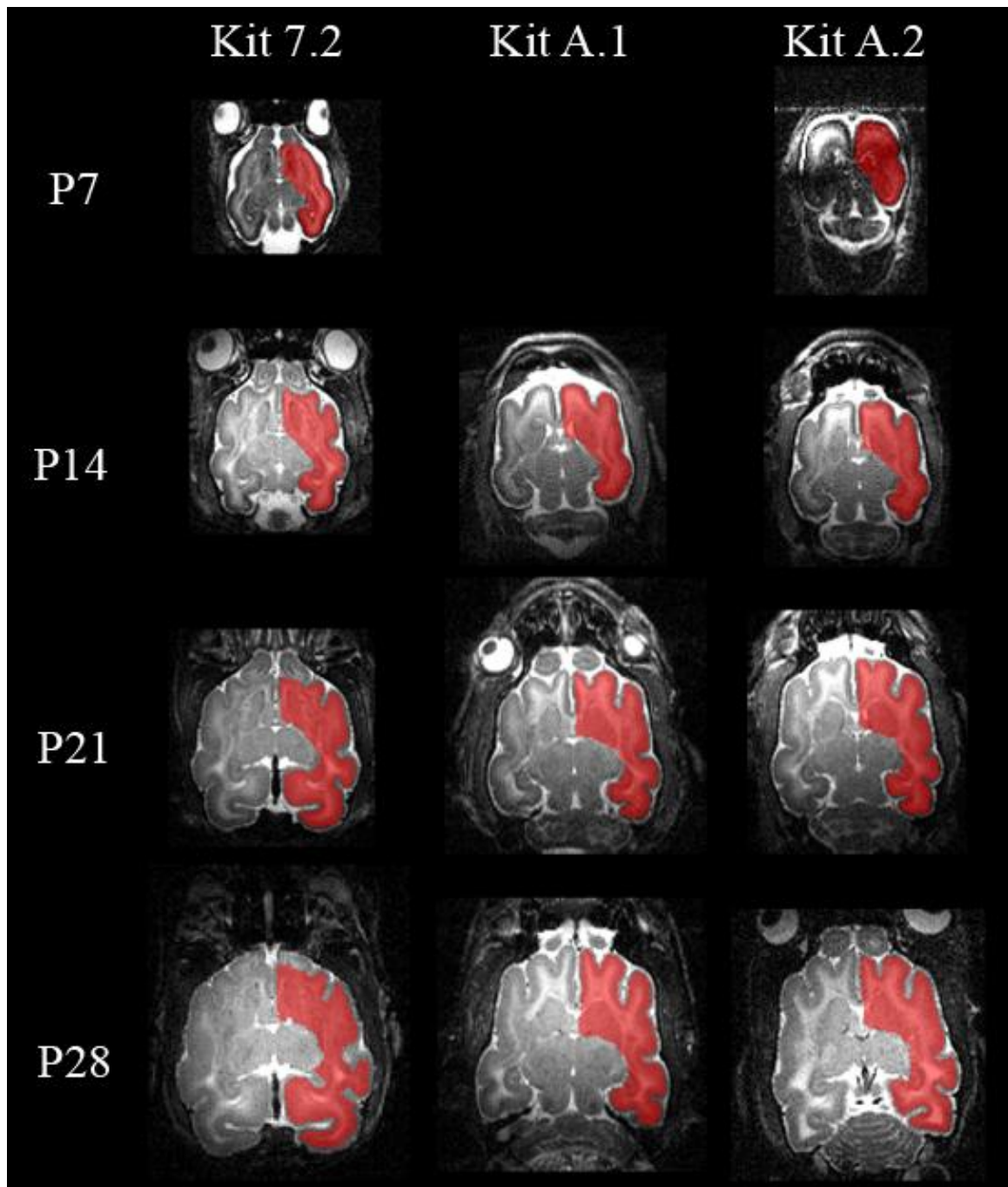
Figure 5.2: A single slice of a T2-weighted image acquired at seven day intervals from postnatal day 7 to 28 for each of the three ferret kits. CSF appears bright, GM appears dark gray and WM appears light gray. Each of the image volumes was segmented, and the segmentation for the right hemisphere is shown in red. Segmentation was done

manually using CARET software. The acquired image at P7 for Kit A.1 was not usable due to problems with the coil.
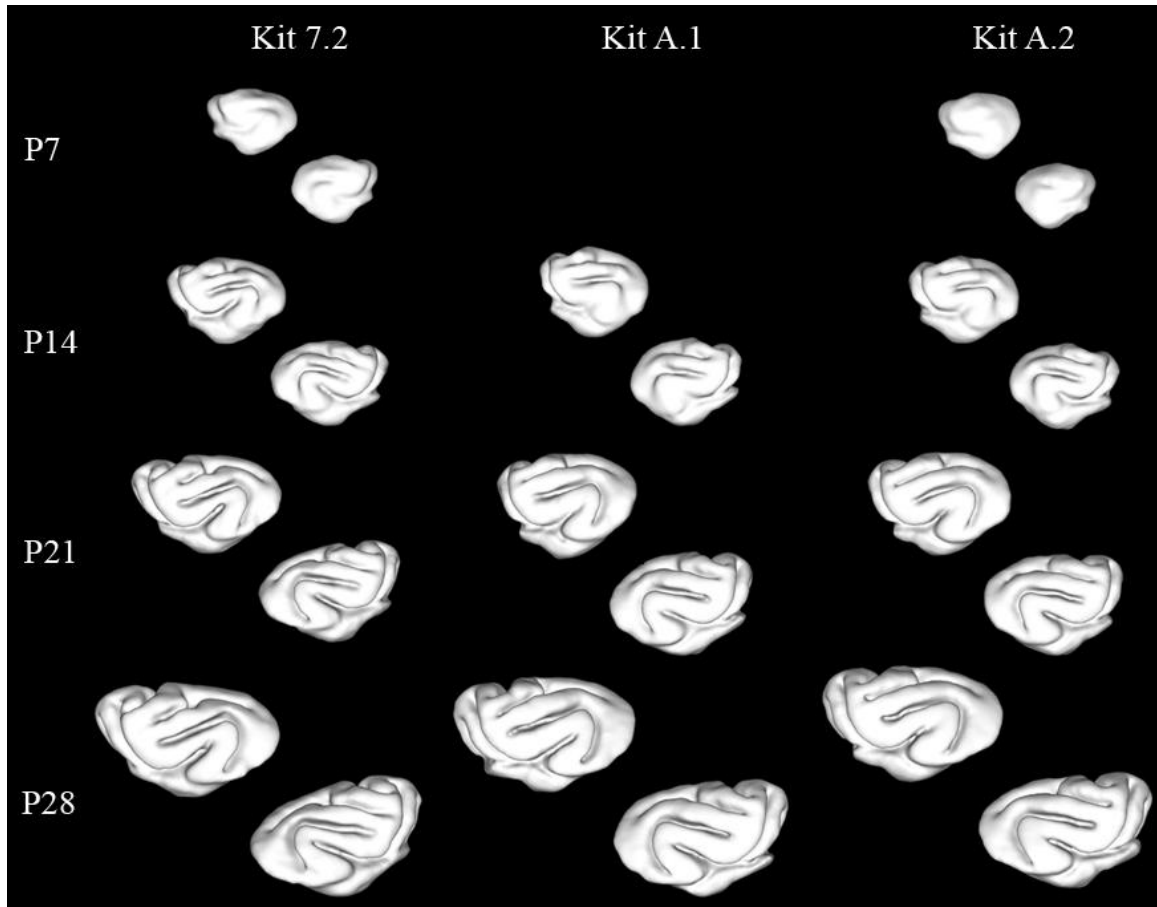
Figure 5.3: Surface mesh models were generated from the segmentation volumes using CARET software for both the right and left hemispheres. Surface representations consisted of 5200-28100 vertices and 10500-57000 triangular faces. All surfaces are to scale with respect to one another.
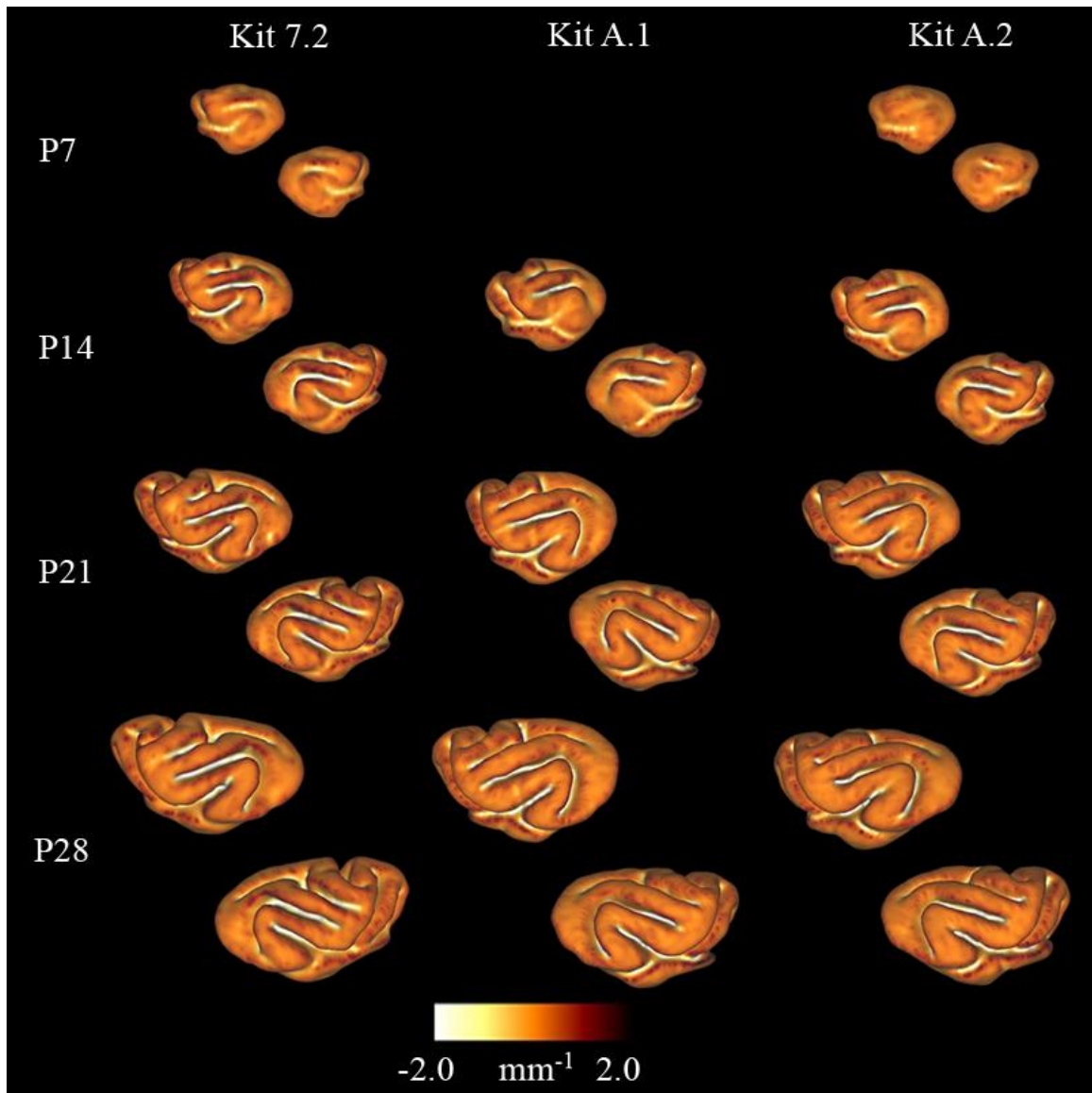
Figure 5.4: Using the approach described in Chapter II, first and second principal curvature values were estimated at each surface point. Mean curvature was calculated in Eq. V.4 for both the left and right hemispheres. Mean curvature provides a good measure of shape to help drive intra-subject registration.
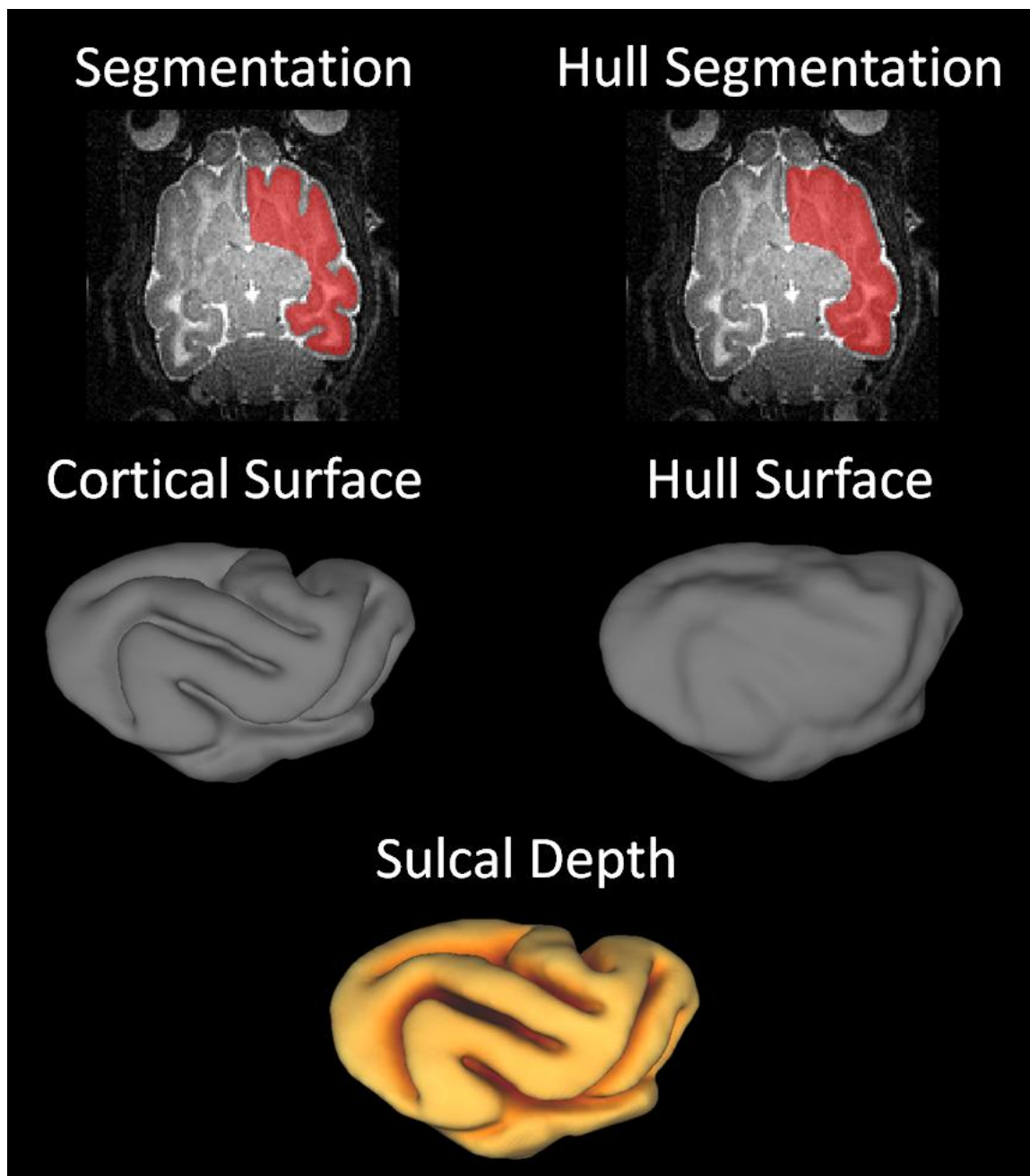
Figure 5.5: A cortical surface and a convex hull are generated from segmentation volumes in CARET. The hull segmentation is generated from the cortical segmentation volume. Sulcal depth measures the distance from the cortical surface to the cerebral hull for each point on the surface.
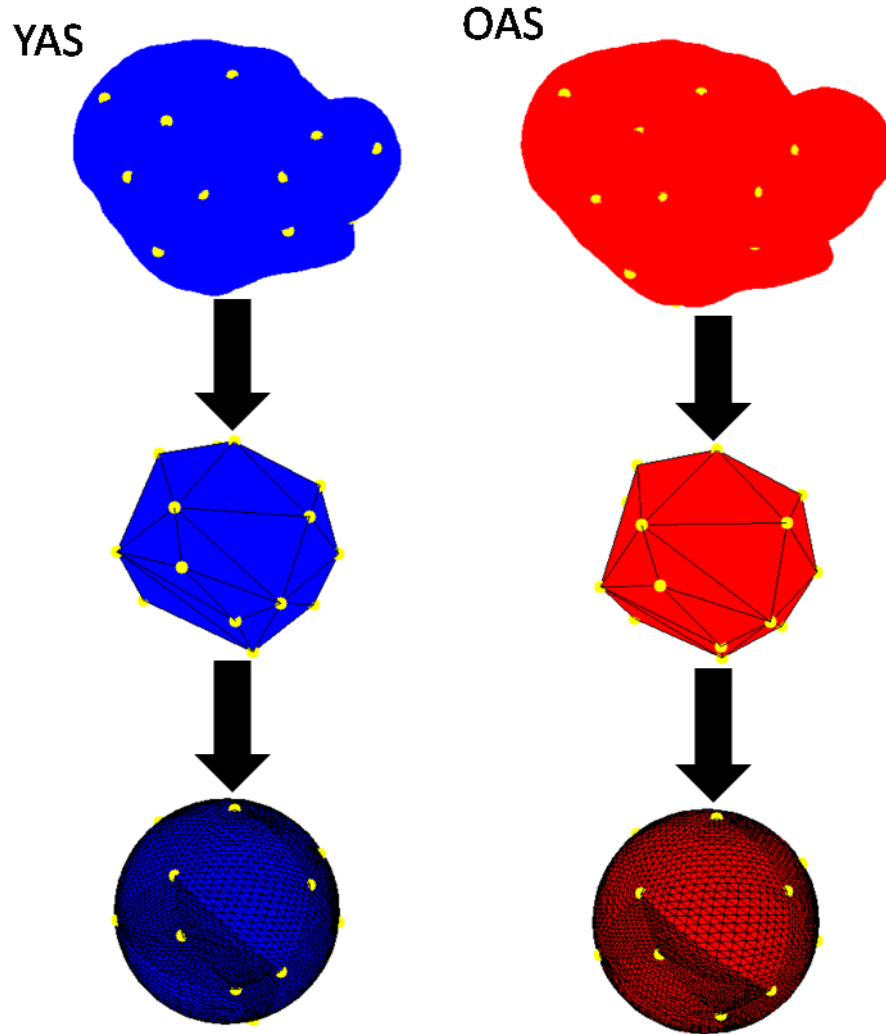
Figure 5.6: Determination of an initial correspondence between arbitrary surfaces: Using mean curvature as a guide, a series of approximately 20 landmark points is identified on both YAS and OAS. These landmark points define a very coarse correspondence between YAS and OAS. The landmark points are projected onto YSS and OSS, and a triangular mesh is created. The mesh is very coarse, containing 20 vertices and 36 faces. A custom function *refine_mesh* is used to create a denser landmark surface. The function works by calculating the midpoint on each line segment, turning each face into four new faces. This process is repeated multiple times until the mesh contains approximately 4900 vertices

and 9700 faces. The refined meshes act as a scaffolding to describe the shape of the YAS using the vertices from OAS.
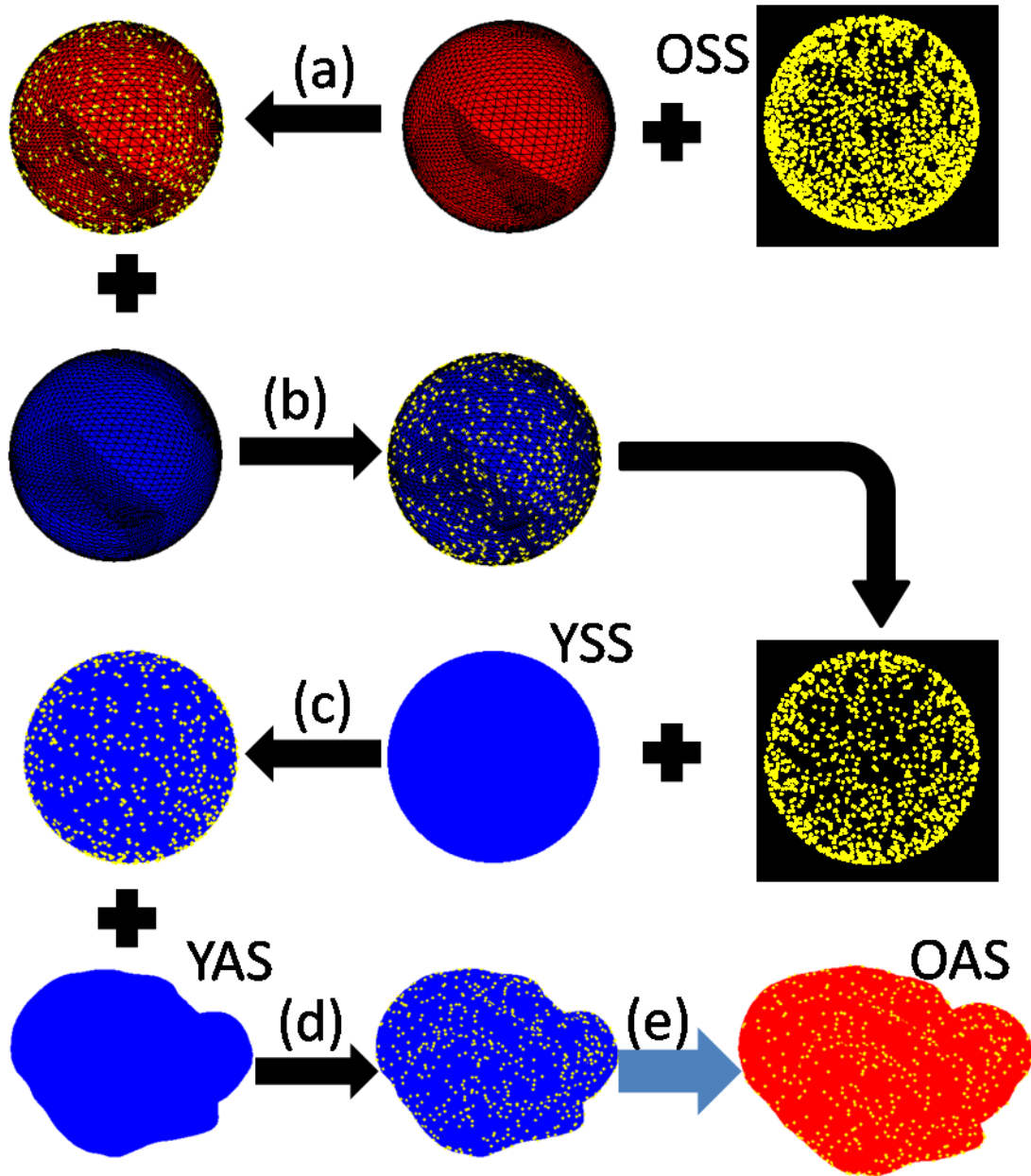
Figure 5.7: In order to apply the LACROSS registration approach, an initial correspondence between YAS and OAS is required. The goal is to take the points that describe OAS (The points from OSS are used, as a one-to-one relationship already exists

with OAS). The dense meshes created from the identified landmark points are used to create a correspondence between YAS and OAS. (a) The points from OSS are projected onto the older dense landmark mesh. (b) The correspondence between the older and younger dense meshes is known, so the projected coordinate location with respect to the younger dense landmark mesh is also known. (c) Those coordinates are then projected onto the YSS. (d) A correspondence is known between YSS and YAS, so the relative location of the projected coordinates on YAS is also known. (e) YAS is described in terms of the coordinates of OAS, so the correspondence between YAS and OAS is known.
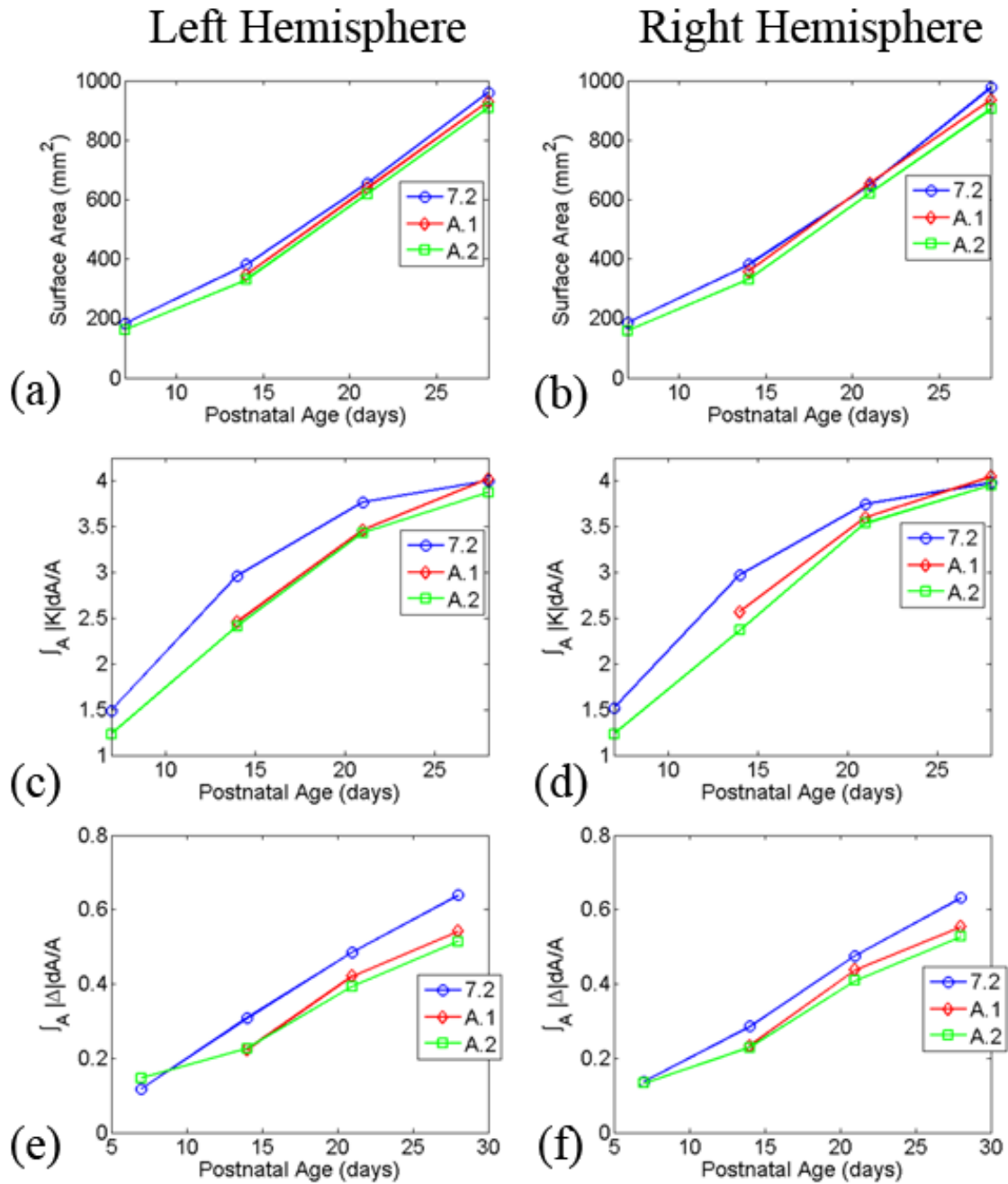
Figure 5.8: Global measures of shape were calculated for all of the surfaces. (a,b) Surface area as a function of postnatal age in days. (c,d) Normalized average mean curvature as a function of postnatal age in days. (e,f) Normalized average sulcal depth as a function of postnatal age in days.

Figure 5.9: The surface matching terms are used to drive the registration process. They are based on mean curvature and distance from the medial wall. (a,b) The surface matching term f_1 plotted on the P14 cortical surface. (c,d) The surface matching term f_2 plotted on the P21 cortical surface (e,f) The difference between f_1 and f_2 for the initial correspondence. (g,h) Difference between f_1 and f_2 after relaxation.

Figure 5.10: Local growth is calculated from the determinant of the deformation gradient tensor between the surfaces. (a,b) Areal expansion after an initial correspondence is determined. (c,d) Areal expansion after the relaxation algorithm is applied. Note that the color scale is the same for both. (e) The rate of change of the total energy function decreases as the algorithm proceeds

Figure 5.11: The dilatation ratio between the registered P14 and P21 cortical surfaces. The range of the color scale is reduced from the previous image to highlight differences in growth between regions.
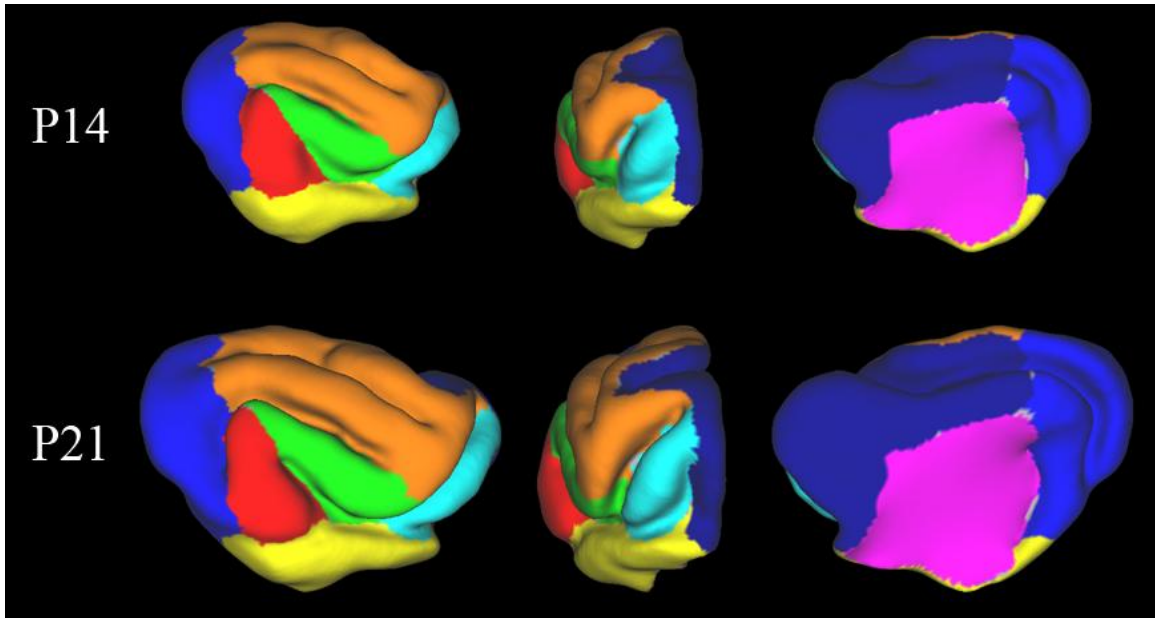
Figure 5.12: Using CARET software, a series of patches were drawn on the P14 cortical surface and mapped to the P21 cortical surface. The regions were selected based on anatomical features (e.g., sulcal and gyral boundaries, the medial wall).
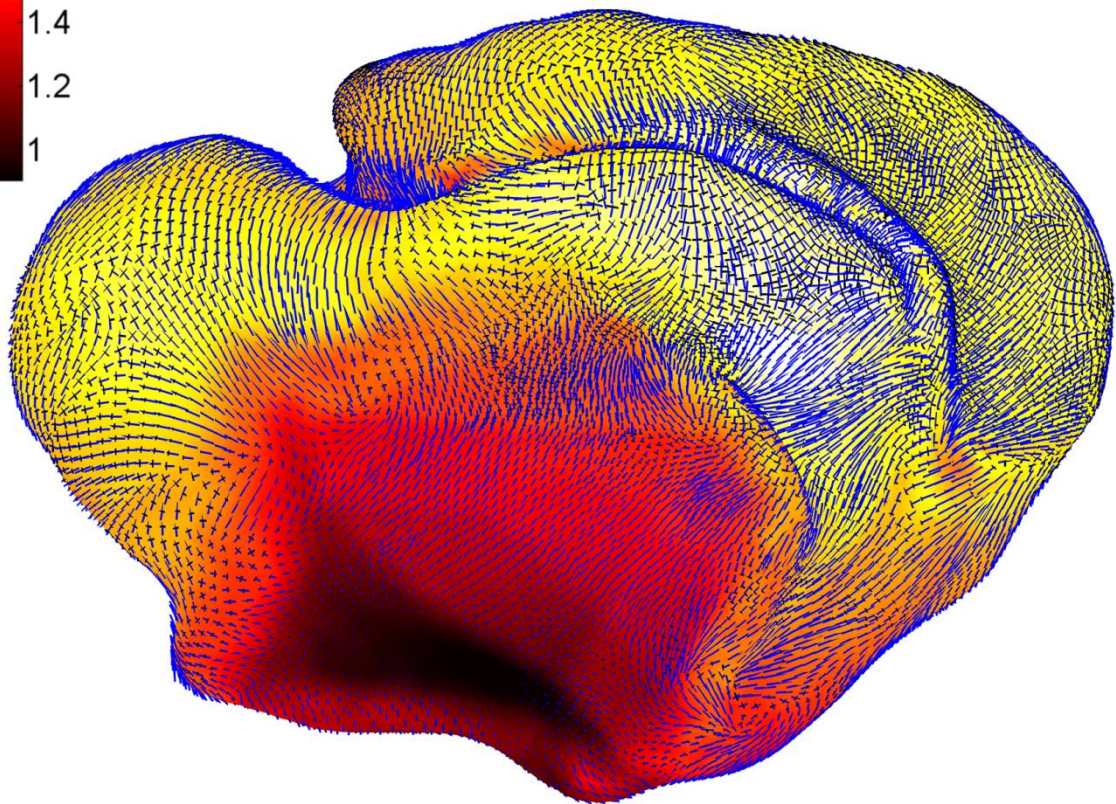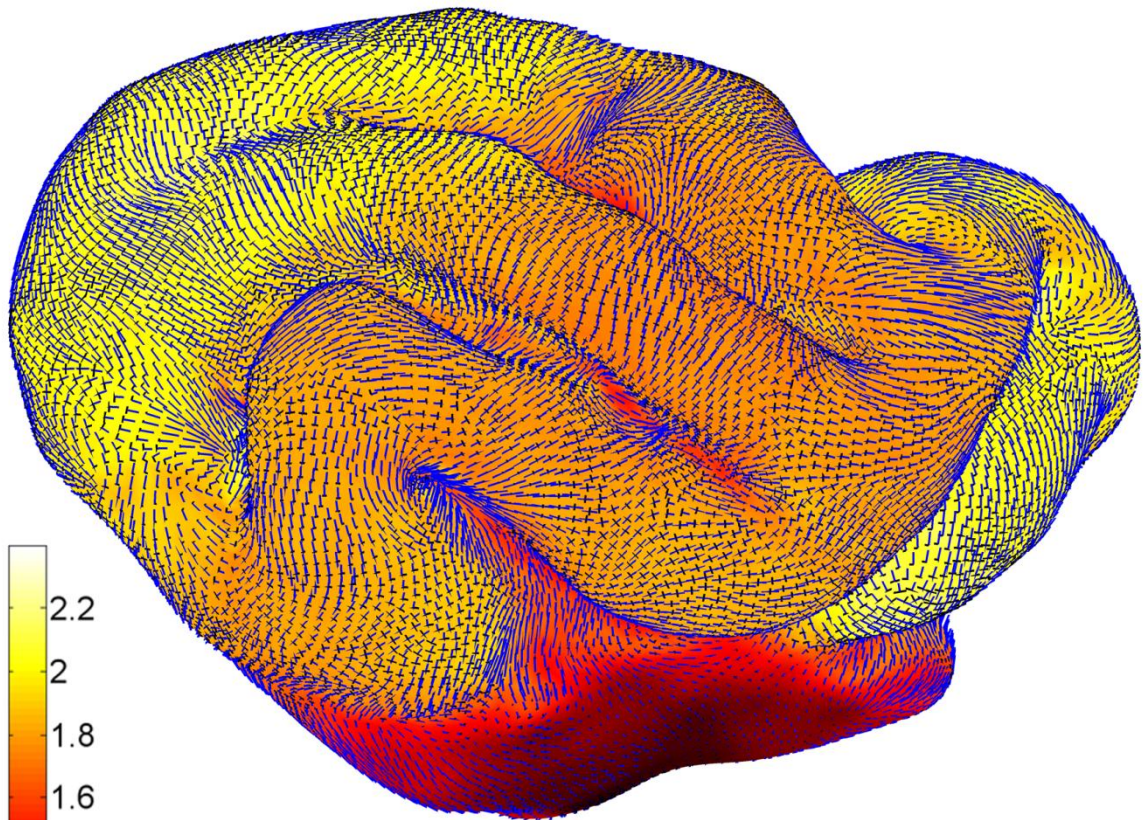
Figure 5.13: This figure shows the eigenvectors that correspond to the first (blue) and second (black) principal strains with the dilatation ratio as an underlay. The vector lengths are scaled by their respective principal strain. Second principal strain is always less than or equal to first principal strain. A principal strain value equal to zero corresponds to no growth in that direction.
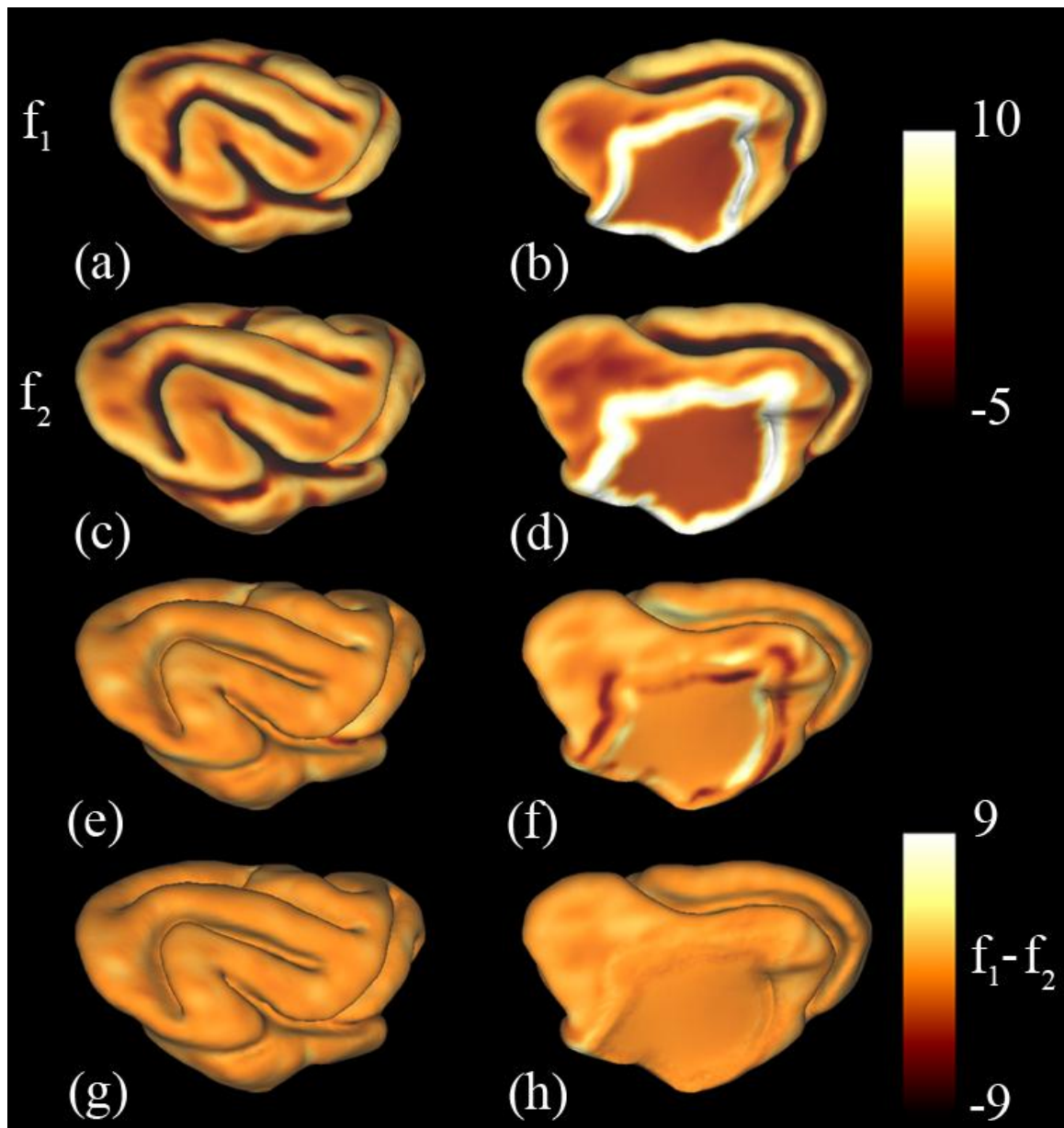
Figure 5.14: Surface matching terms used to drive the registration process for P21 to P28 brains. They are based on mean curvature and distance from the medial wall. (a,b) The surface matching term f_1 plotted on the P21 cortical surface. (c,d) The surface matching term f_2 plotted on the P28 cortical surface (e,f) The difference between f_1 and f_2 for the initial correspondence. (g,h) The difference between f_1 and f_2 after relaxation.
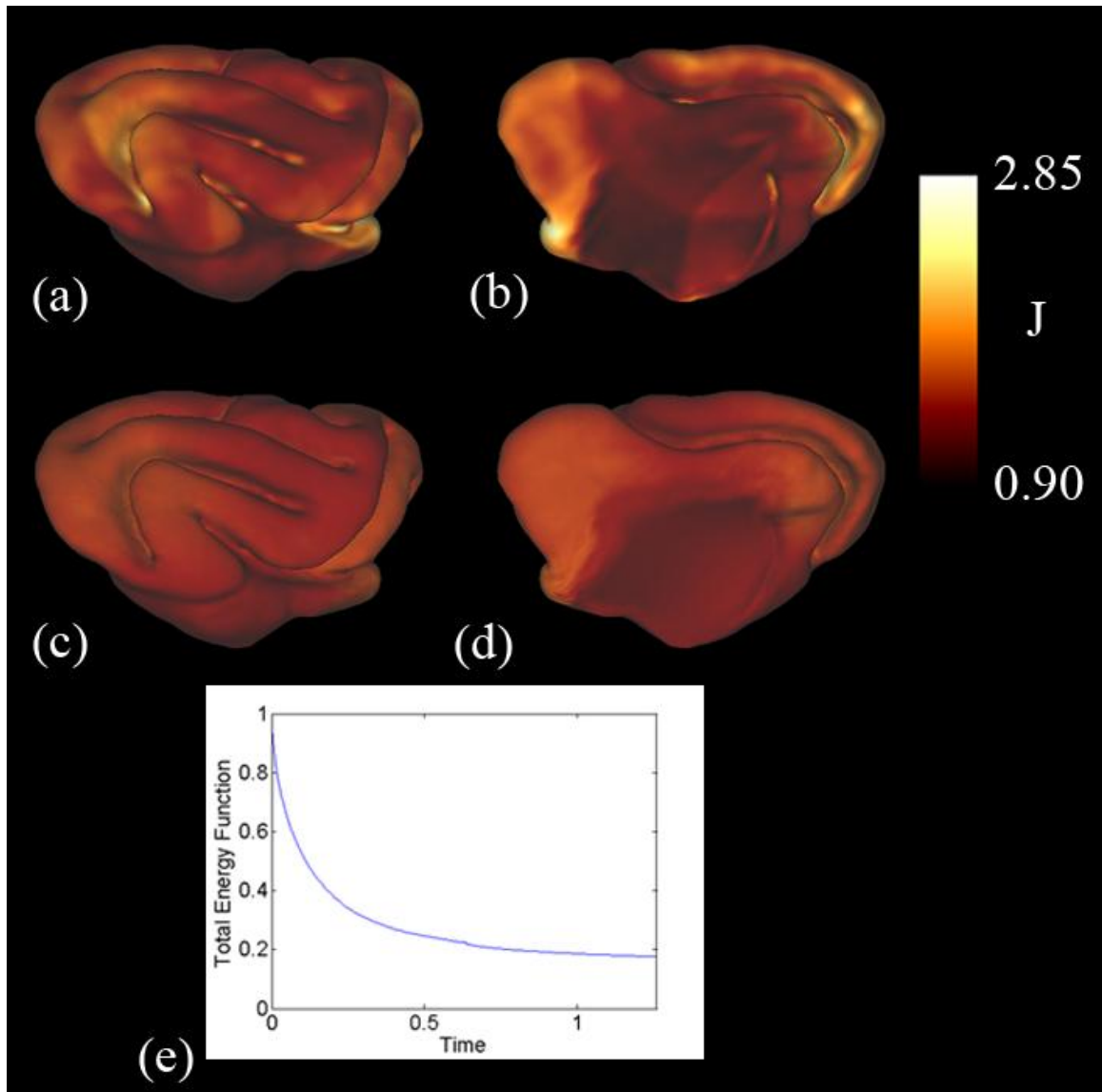
Figure 5.15: Local growth is calculated from the determinant of the deformation gradient tensor between the surfaces. (a,b) Areal expansion after an initial correspondence is determined. (c,d) Areal expansion after the relaxation algorithm is applied. Note that the color scale is the same for both. (e) As the algorithm proceeds, the rate of change of the total energy function decreases.
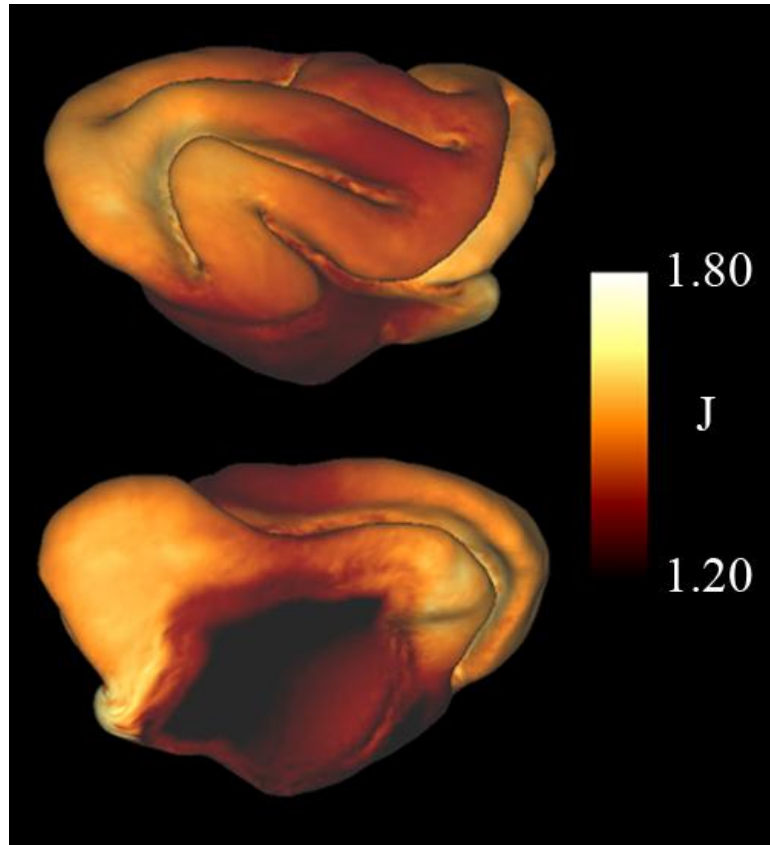
Figure 5.16: The dilatation ratio between the registered P21 and P28 cortical surfaces. The range of the color scale is less than the previous image to highlight differences between regions.
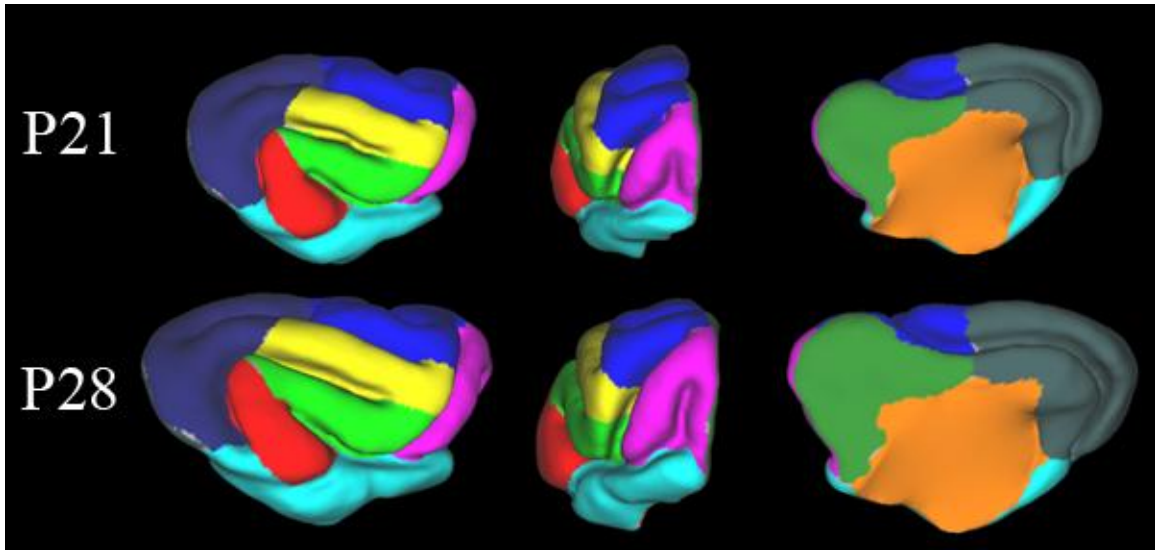
Figure 5.17: Using CARET software, a series of patches were drawn on the P21 cortical surface and mapped to the P28 cortical surface. The regions were selected based on anatomical features (e.g., sulcal and gyral boundaries, the medial wall).
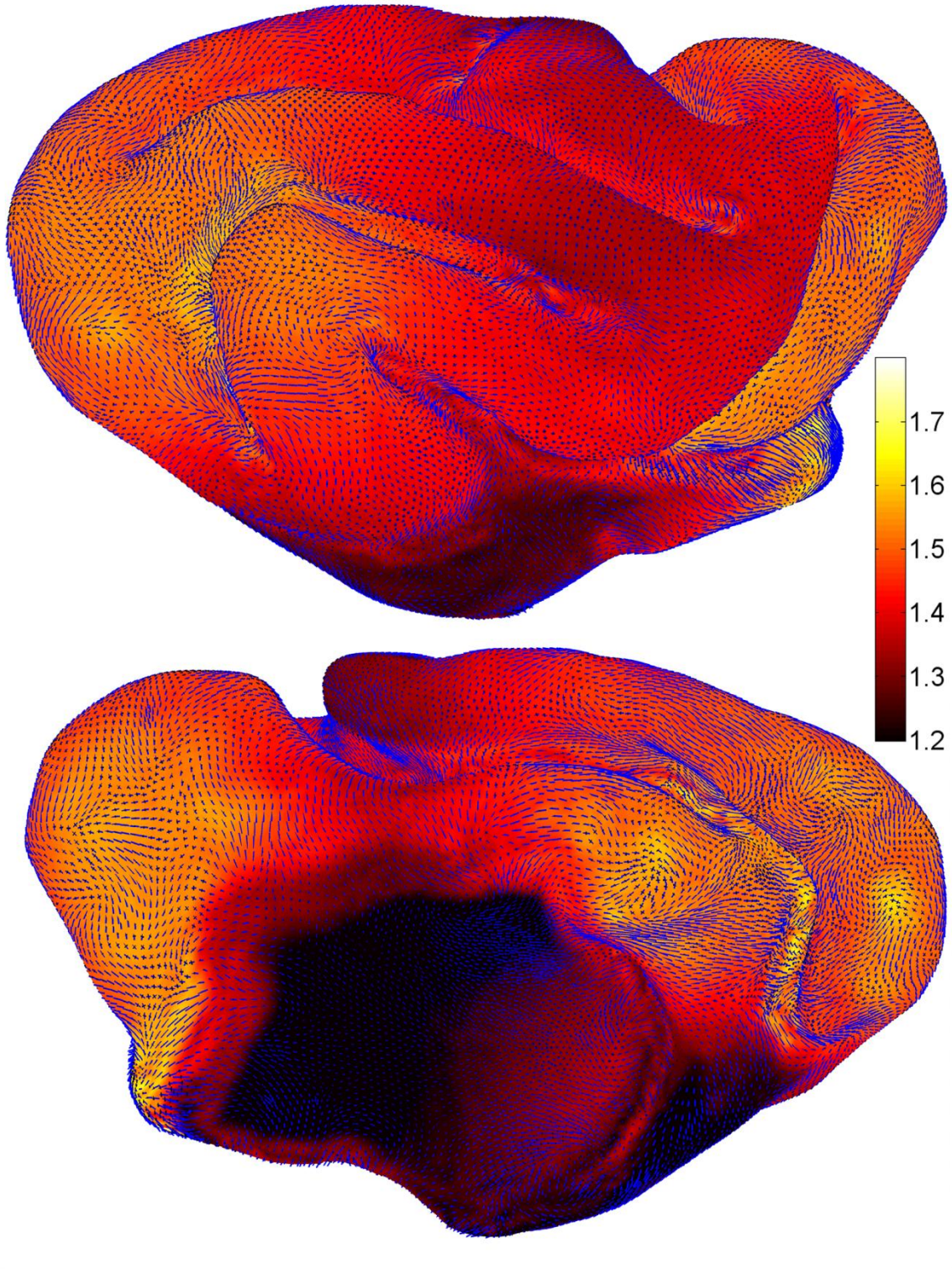
Figure 5.18: This figure shows the eigenvectors that correspond to the first (blue) and second (black) principal strains with the dilatation ratio as an underlay. The vector

lengths are scaled by their respective principal strain. Second principal strain is always less than or equal to first principal strain. A principal strain value equal to zero corresponds to no growth in that direction. The range of the dilatation ratio is from 1.2 to 1.8, which is the same as Fig. 5.16.
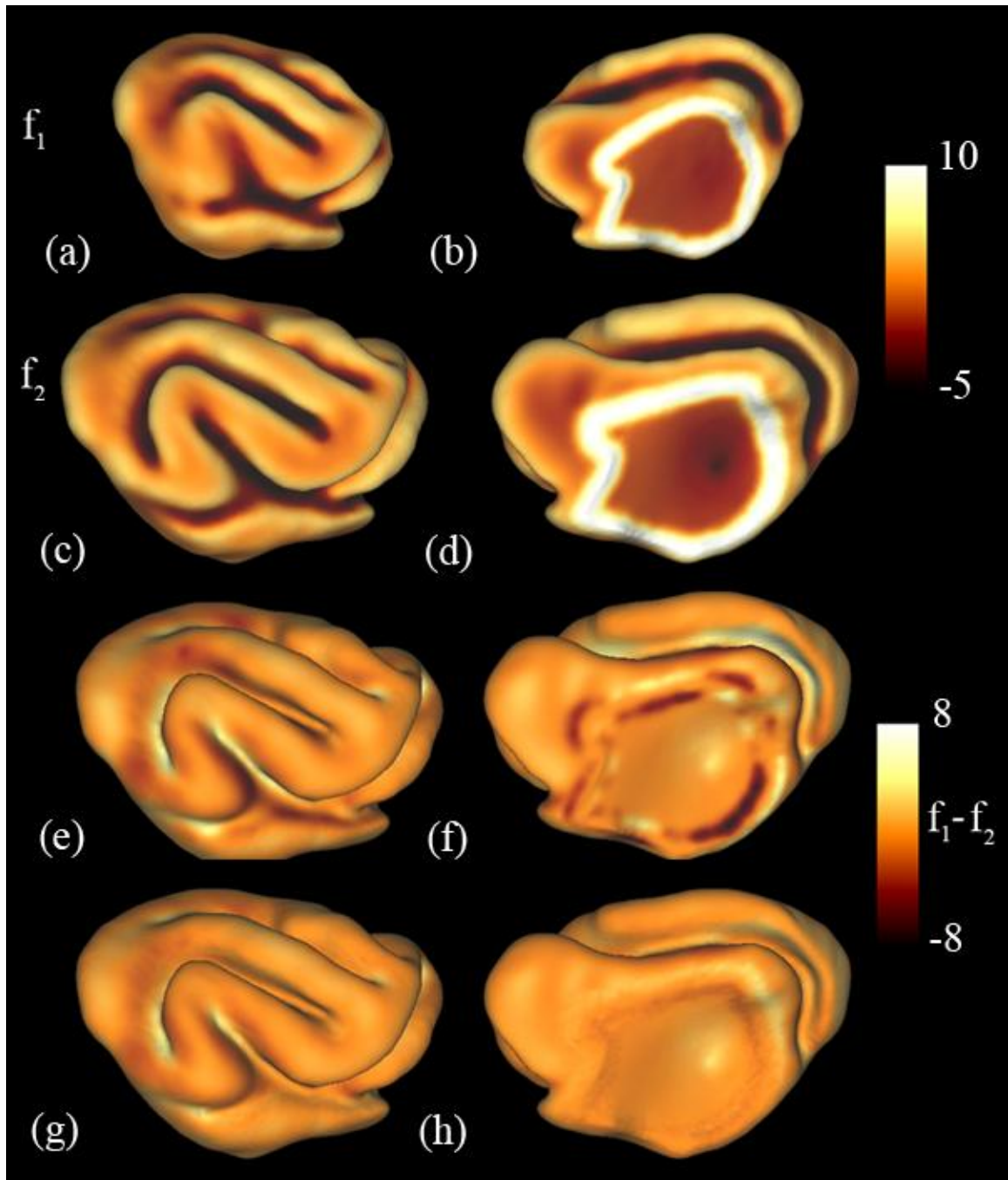
Figure 5.19: The surface matching terms are used to drive the registration process. They are based on mean curvature and distance from the medial wall. (a,b) The surface matching term f_1 plotted on the P14 cortical surface. (c,d) The surface matching term f_2 plotted on the P21 cortical surface (e,f) The difference between f_1 and f_2 for the initial correspondence. (g,h) Difference between f_1 and f_2 after relaxation.
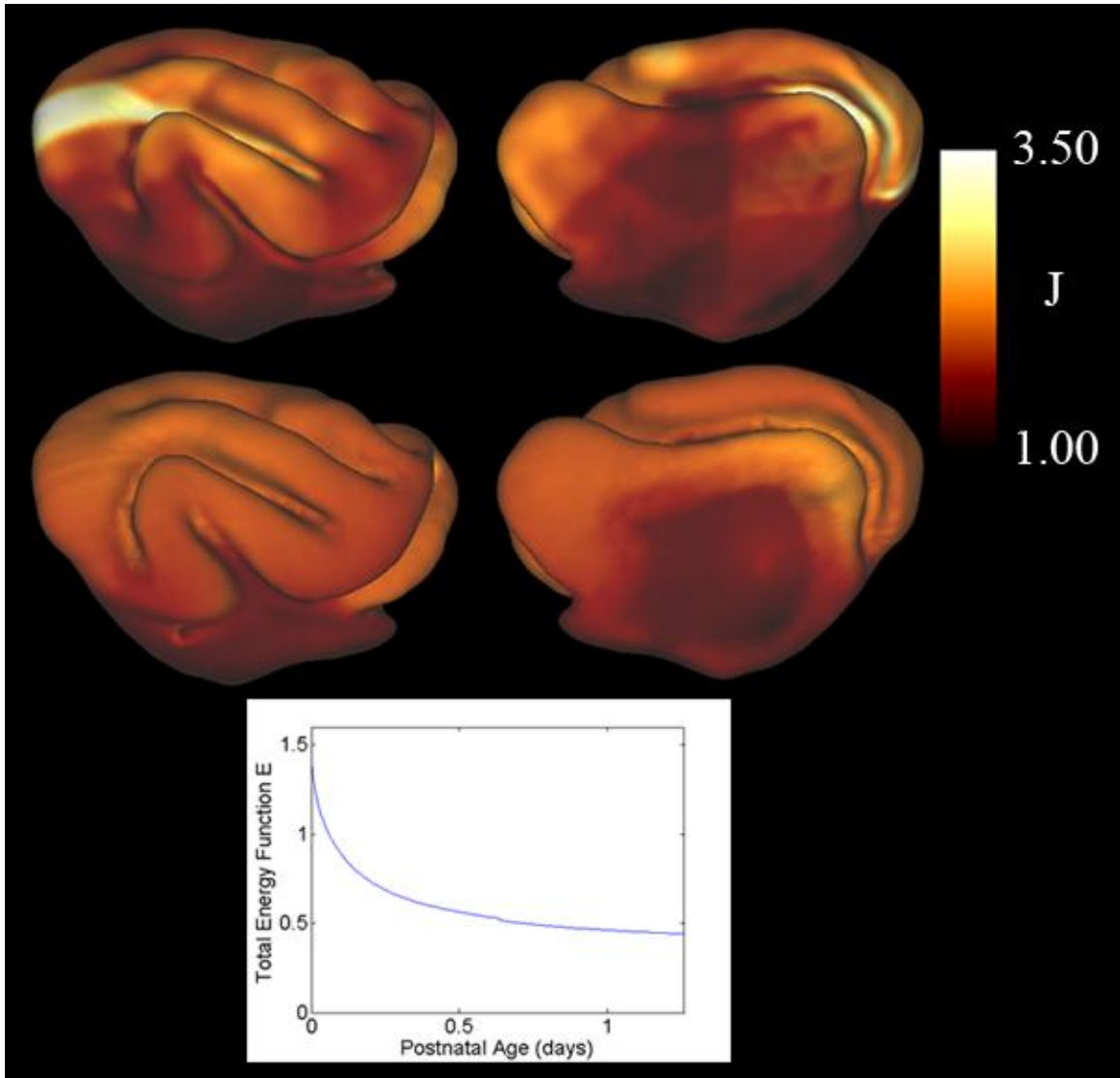
Figure 5.20: Local growth is calculated from the determinant of the deformation gradient tensor between the surfaces. (a,b) Areal expansion after an initial correspondence is determined. (c,d) Areal expansion after the relaxation algorithm is applied. Note that the color scale is the same for both. (e) The rate of change of the total energy function decreases as the algorithm proceeds
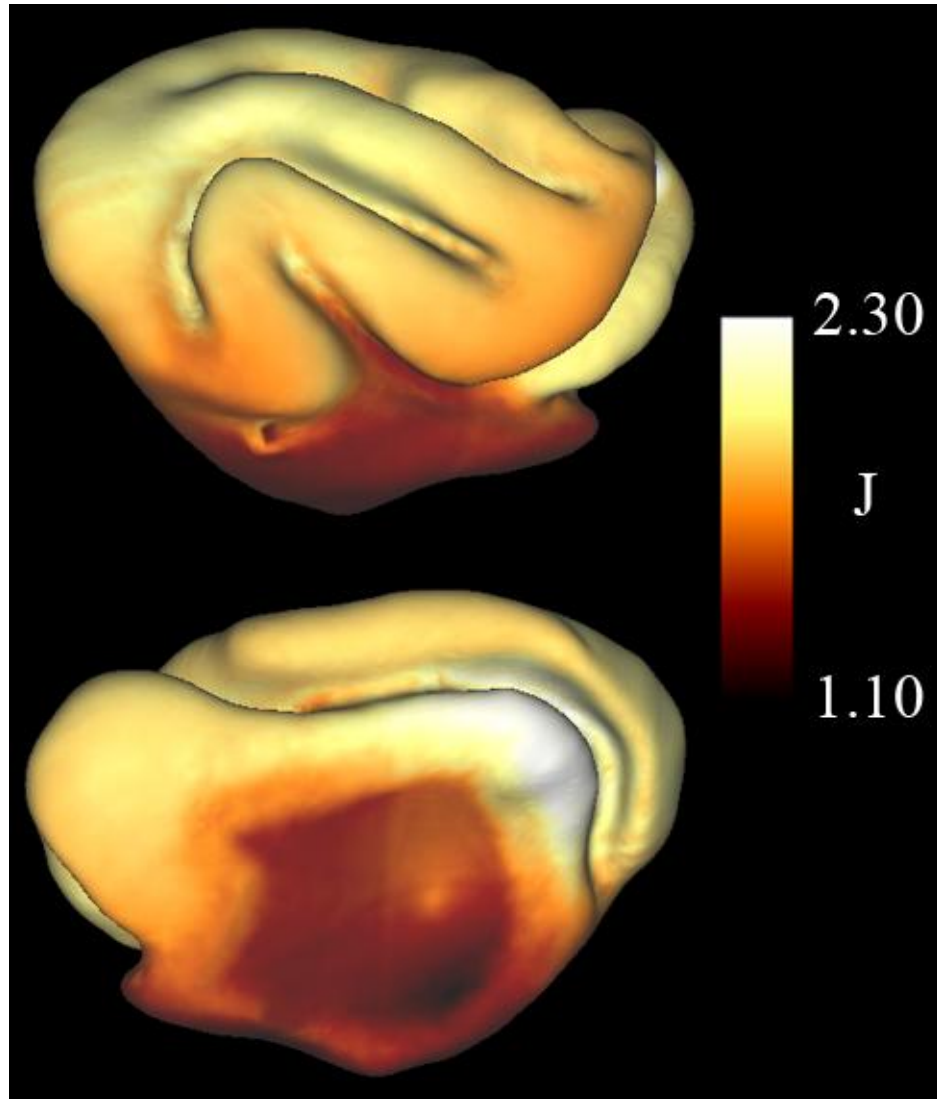
Figure 5.21: The dilatation ratio between the registered P14 and P21 cortical surfaces. The range of the color scale is reduced from the previous image to highlight differences in growth between regions.

157
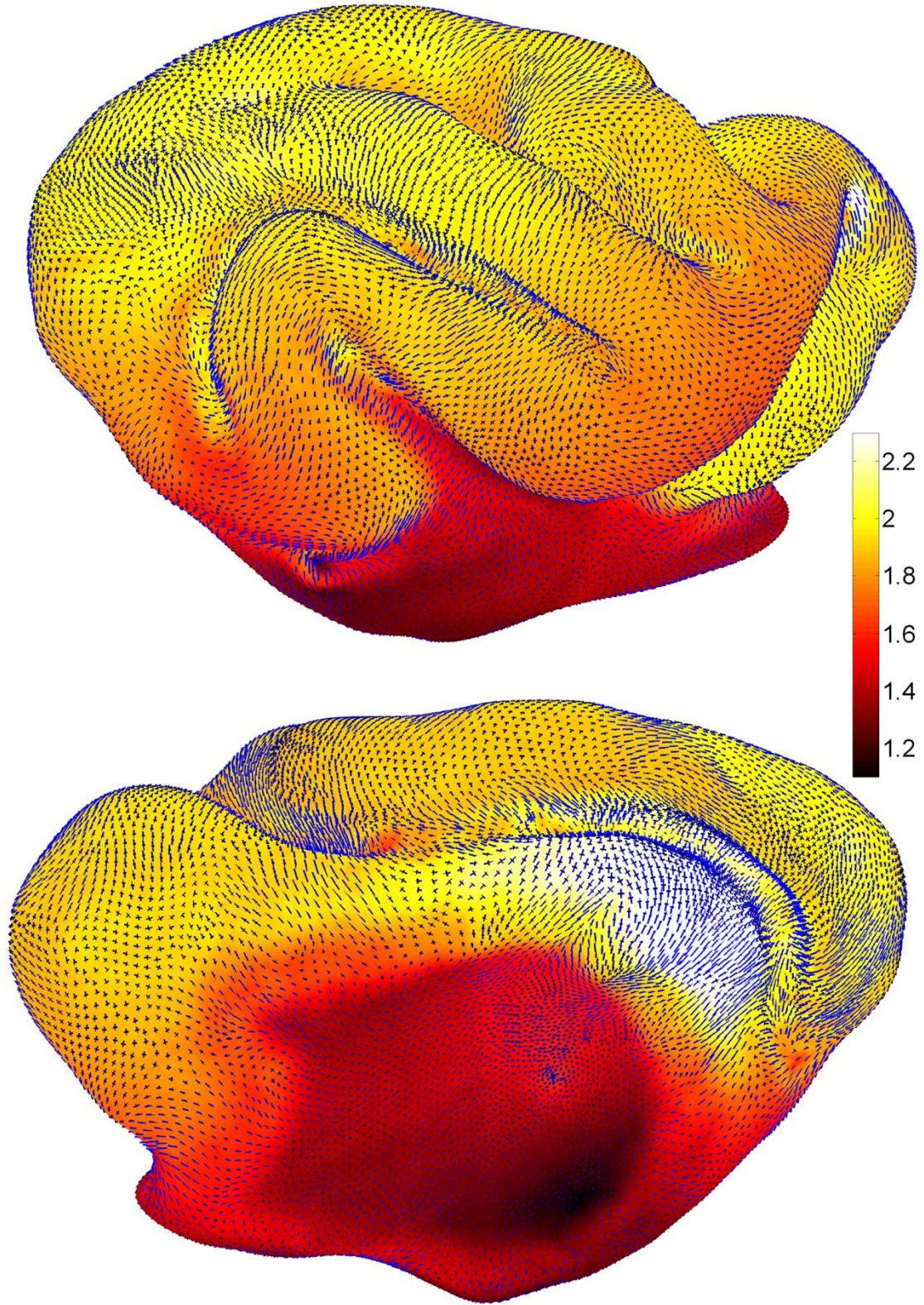
Figure 5.22: This figure shows the eigenvectors that correspond to the first (blue) and second (black) principal strains with the dilatation ratio as an underlay. The vector lengths are scaled by their respective principal strain. Second principal strain is always less than or equal to first principal strain.
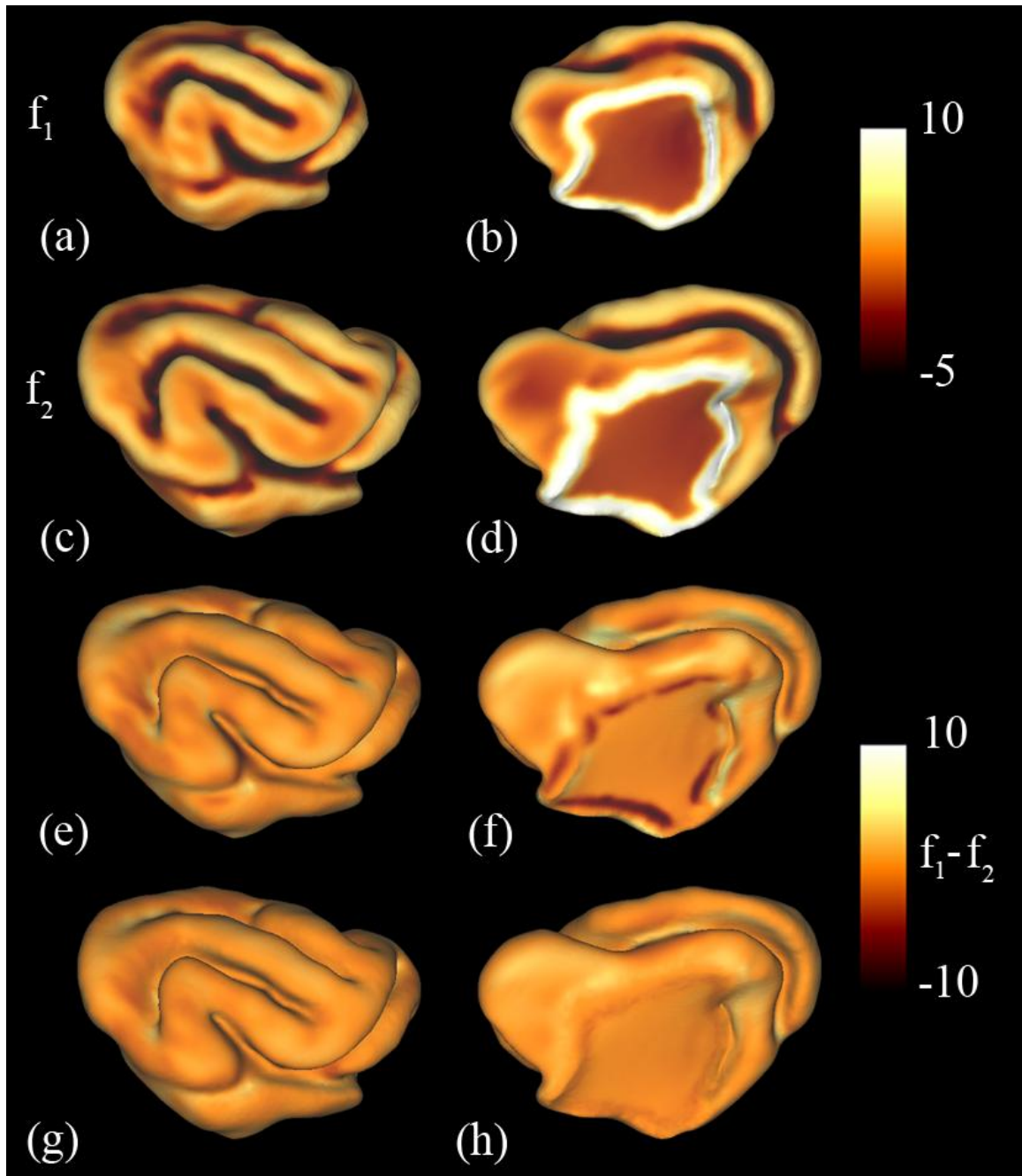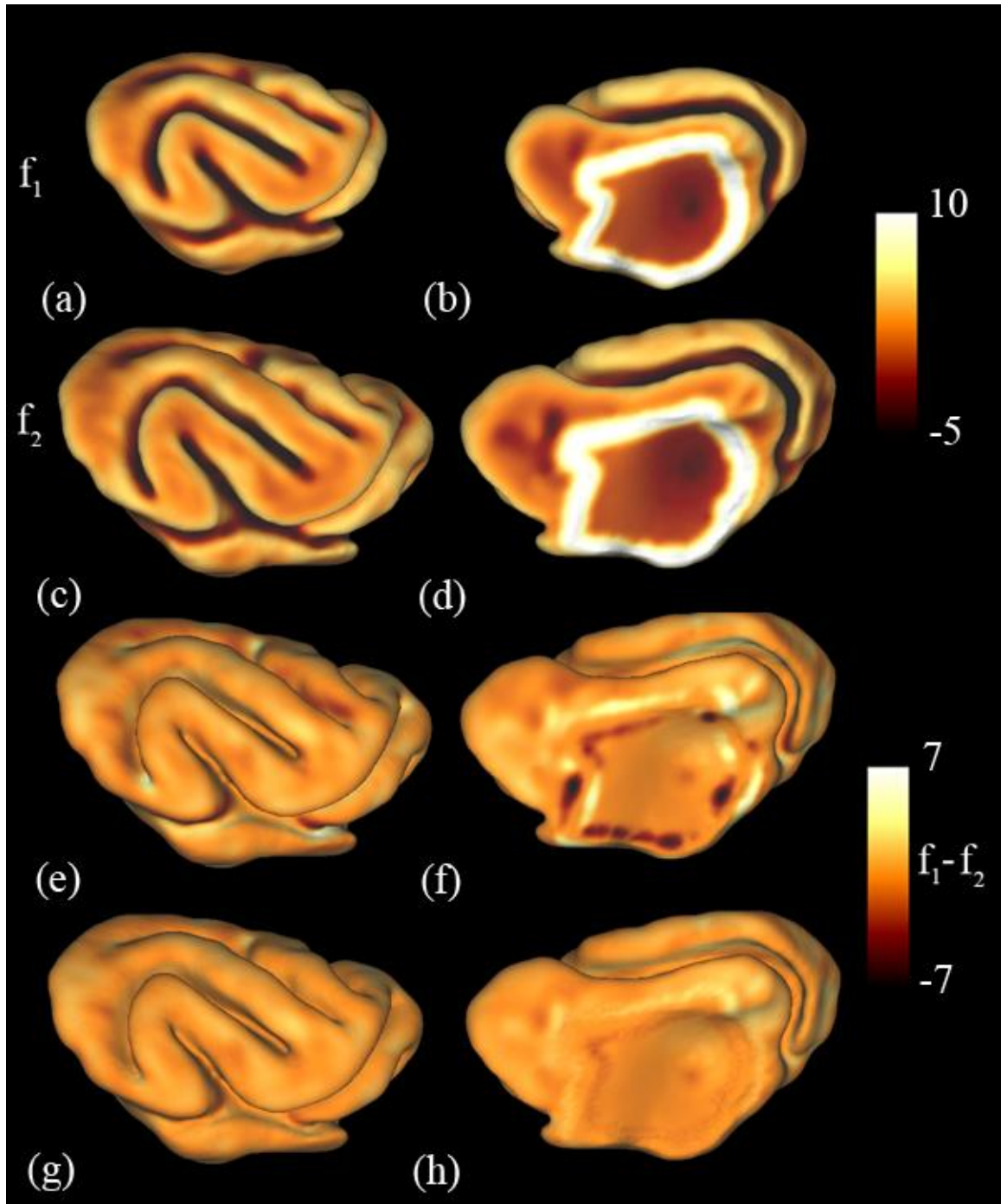
Figure 5.23: Surface matching terms used to drive the registration process for P21 to P28 brains. They are based on mean curvature and distance from the medial wall. (a,b) The surface matching term f_1 plotted on the P21 cortical surface. (c,d) The surface matching

term f_2 plotted on the P28 cortical surface (e,f) The difference between f_1 and f_2 for the initial correspondence. (g,h) The difference between f_1 and f_2 after relaxation.
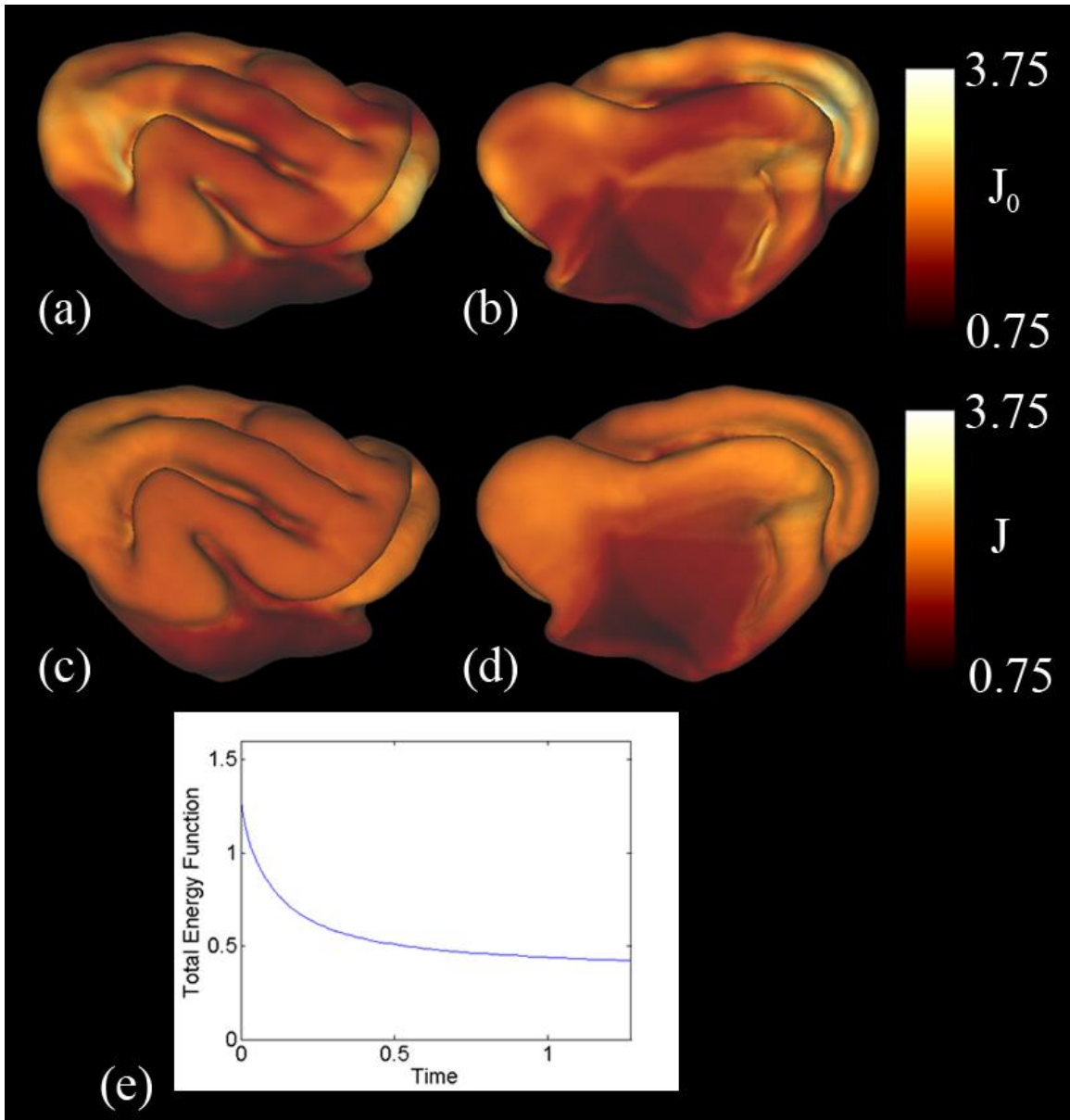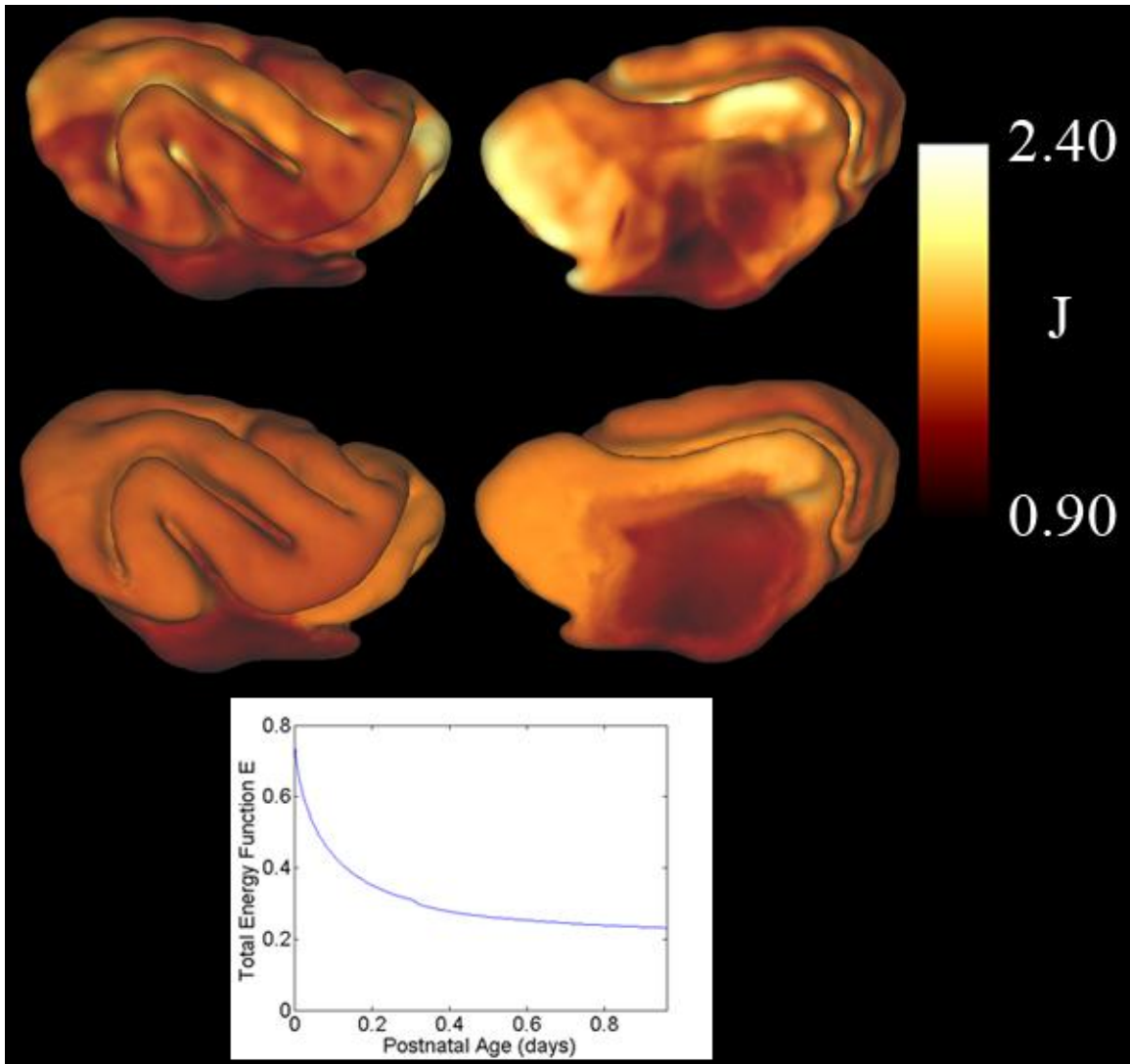

Figure 5.24: Local growth is calculated from the determinant of the deformation gradient tensor between the surfaces. (a,b) Areal expansion after an initial correspondence is determined. (c,d) Areal expansion after the relaxation algorithm is applied. Note that the color scale is the same for both. (e) As the algorithm proceeds, the rate of change of the total energy function decreases.
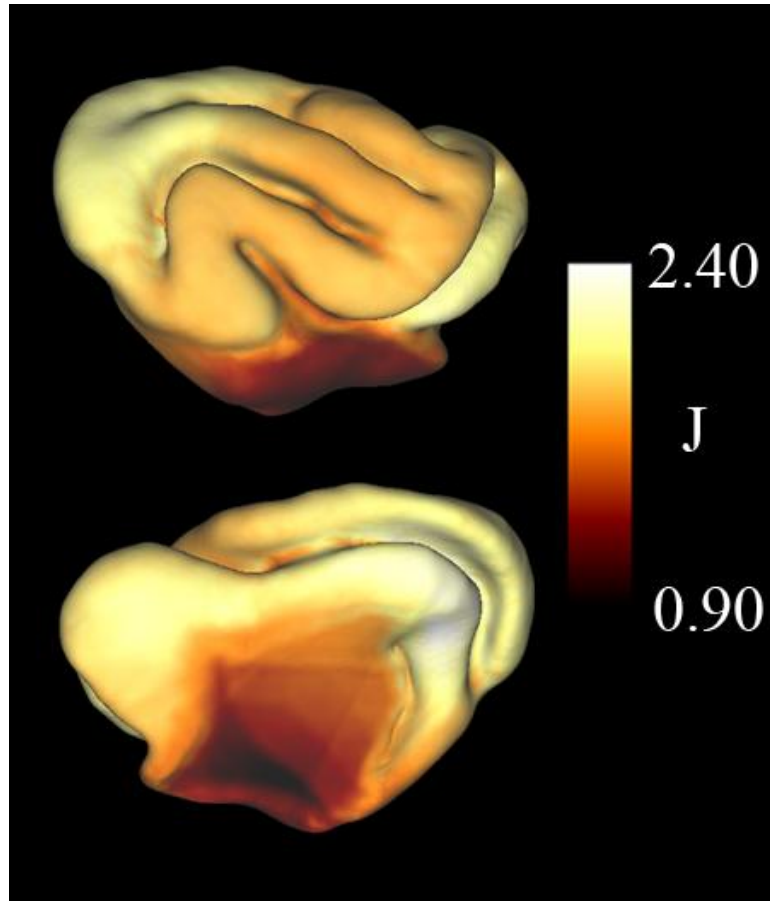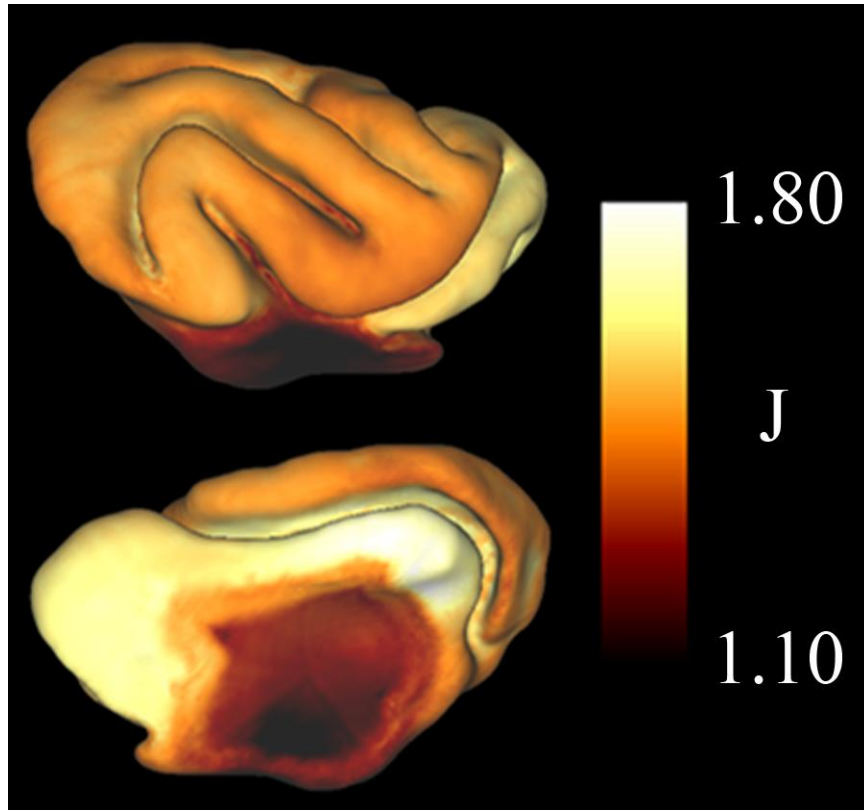
Figure 5.25: The dilatation ratio between the registered P21 and P28 cortical surfaces. The range of the color scale is less than the previous image to highlight differences between regions.
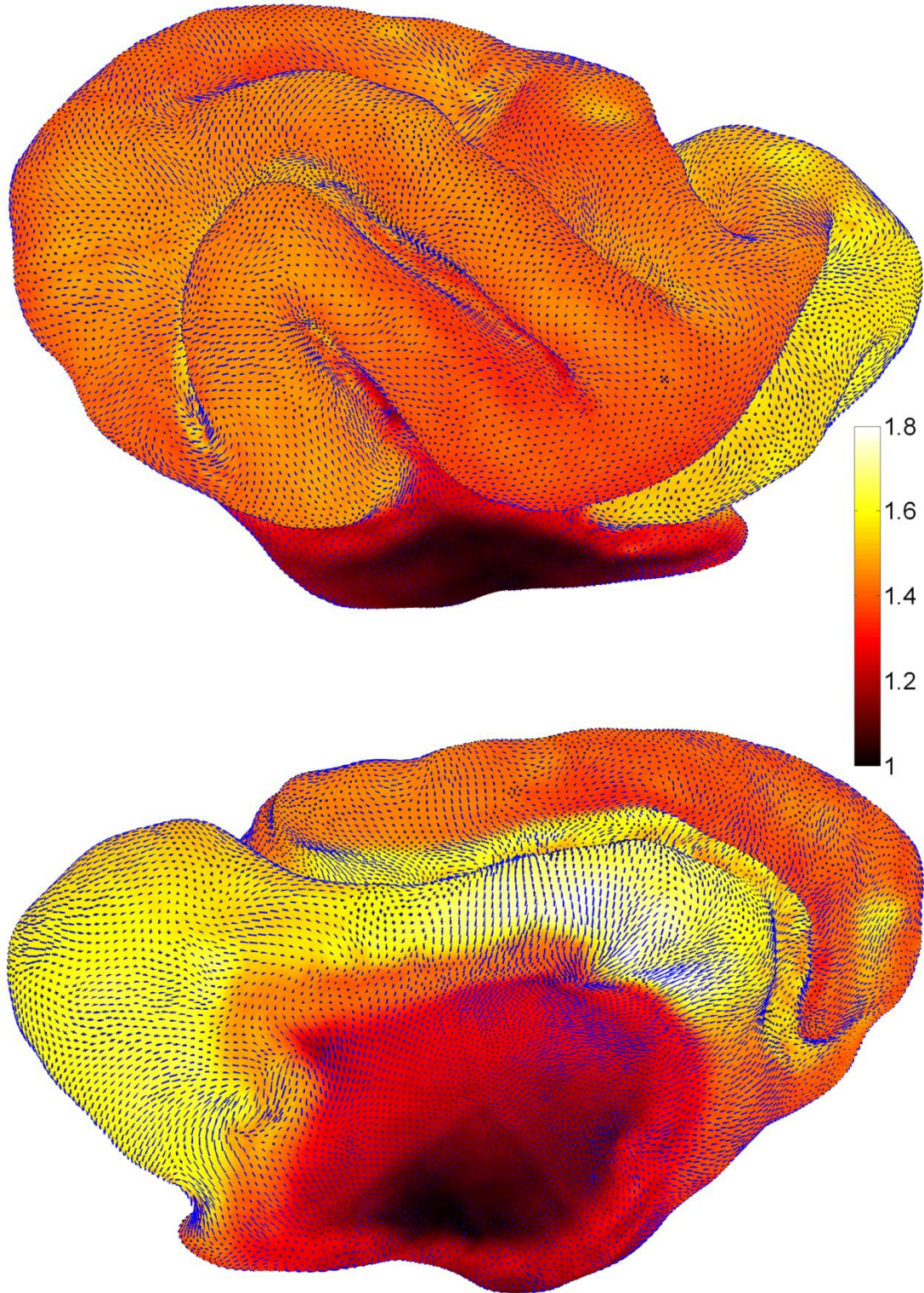
Figure 5.26: This figure shows the eigenvectors that correspond to the first (blue) and second (black) principal strains with the dilatation ratio as an underlay. The vector

lengths are scaled by their respective principal strain. Second principal strain is always less than or equal to first principal strain. A principal strain value equal to zero corresponds to no growth in that direction.

Table 5.1: Postnatal age in days at time of scan; echo time (TE) in ms; pulse repetition time (TR) in s; number of averages acquired (nt); resolution of voxels in microns

Age (days)	TE (ms)	TR (s)	nt	Resolution (µm)
7	80	4	10	250x250x250
14	60	4	8	250x250x250
21	58	4.1-4.4	4	250x250x250
28	55	4.2-4.4	4	250x250x250

Table 5.2: Parameters used for registration of P14 to P21 and P21 to P28 for Kits A.1 and A.2. μ is the shear modulus, κ is the bulk modulus, ν is the coefficient of damping and t is the time parameter entered in COMSOL.

	Iteration	μ (Pa)	κ (Pa)	ν (N-s/m^3)	No. Elements	t	time (hours)
A.2 P14=>P21	1	0.1	10	100	9068	0--1.27	4.1
A.2 P21=>P28	1	0.1	10	100	9068	0--0.63	3.1
	2	0.1	10	100	9086	0.63--1.26	2.3
A.1 P14=>P21	1	0.1	10	100	9068	0—0.63	2.9
	2	0.1	10	100	9068	0.63--1.26	2.5
A.1 P21=>P28	1	0.1	10	100	9068	0--0.63	2.1
	2	0.1	10	100	9086	0.63--1.26	2.0

Chapter VI. Summary and Future Work

1. Summary of work

We have developed a series of analytical tools that can be applied to calculate growth of the cerebral cortex during development of the brain. We have applied these tools to mesh surfaces of the cortex created images of ferret brains that were acquired using MRI. Our results show that the rate of growth in the brain varies between regions and over time for a single ferret kit.

Here, we list the specific aims of this dissertation along with how the aims have been fulfilled.

1. Acquire magnetic resonance images of ferret brains at different postnatal ages. In Chapter V we presented the images acquired from ferrets at different stages of development.

2. Create cortical surface representations from images. Also in Chapter V we showed the surfaces that were generated by manual segmentation of the acquired MRI volumes.

3. Develop analysis tools to estimate Lagrangian strain and curvature. In Chapter II we described a novel method to estimate Lagrangian strain between two multi-valued surfaces that have a one-to-one correspondence. This was accomplished by locally fitting patches of the surface using a second order polynomial function, and calculating derivatives analytically. In the same way we were able to estimate the curvature tensor at each surface point.

4. Develop and validate an intra-subject surface registration algorithm. In Chapter III we develop the theoretical framework for an intra-subject surface registration algorithm, and in Chapter IV we validate the approach by applying it to a series of test cases with known

growth patterns. Starting from an initial rough estimate of correspondence, the algorithm refines the correspondence so that distortions between anatomical surfaces are minimized and surface matching features are aligned. The finite element method is used to solve a partial differential equation on a parameterized shape (sphere). Even though the solution is obtained on a sphere, distortions are minimized with respect to the anatomical surfaces. *5. Apply analysis tools to cortical surfaces to measure spatial variations in growth and deformation during development*. In Chapter V we registered cortical surfaces from three time points in a single kit to one another. Areal expansion and Lagrangian strain were calculated between the registered surfaces, which show regional variations in growth over time. Larger variations in growth are seen between P14 to P21 versus P21 to P28.

2. Limitations

Currently we use mean curvature to drive registration, implying that curvature values remain similar during development. While this assumption is plausible, using biologic landmarks that are identifiable over time would provide stronger evidence to support our results. A possible method to identify such landmarks is presented in the next section.

Image resolution limits how well the cortex can be identified. Effects from "partial voluming" sometimes make it difficult to delineate the boundary between CSF and GM. Partial voluming occurs due to the discretization of a continuous structure.

When a polynomial is fit to local patches of the surface, local curvature plays a role is the accuracy of the fit. If the curvature is too high relative to the number of points used to describe the surface in the local patch, then the residual errors may be high, which

affects the accuracy of the estimated derivatives. In the developing ferret brain, the base of a sulcus often contains high curvature values relative to the number of points used to describe them. As such, larger residual errors are seen in these regions, which may affect the accuracy of the registration and growth estimates in these areas.

While our results show differential growth between regions of the cortex, we have only applied our approach to surfaces from two kits. Applying our method to the remainder of the surfaces we generated will provide a more complete picture and greater confidence in our results.

3. Future work

3.1 Imaging of neurovasculature

The identification of small blood vessels within the cortex at different points in development could provide a solid set of landmarks to use to drive longitudinal registration and calculate local growth in the cortex. Using MRI and advanced image reconstruction tools, small blood vessels within the cortex have been identified previously in the visual cortex of a cat (Bolan et al., 2006). We have acquired a series of T_2^*-weighted images in live ferrets at multiple time points. A gradient echo pulse sequence was used to acquire the images. Venograms at two time points are shown in Fig. 6.1. No contrast agents were used to acquire the images. Blood vessels are clearly identifiable at both time points. More work is needed to determine the optimal sequence, parameters and post-processing methods to acquire the best possible images. Also, younger time points (P7-P14) need to be imaged to determine if blood vessels can be

detected when they are considerably smaller. If vessels are not visible, the use of a contrast agent may be necessary.

3.2 Registration of P14 to P21 cortical surface: Landmarks

Here, we repeat the registration of the P14 to P21 cortical surfaces using landmark points in place of mean curvature to drive the registration (Fig. 6.2). The landmark points are the same ones used to obtain an initial correspondence between the surfaces. As the registration proceeds, distortions are relaxed away (Fig. 6.3c,d). The surface matching terms become more aligned along the medial wall and remain in place at the landmark points. (Fig. 6.2g,h). The use of landmarks as a driving force provides a similar pattern of growth as when mean curvature is used (Fig. 6.4). Even though the selected landmarks are not biologically significant, this case is significant because it demonstrates that the algorithm can be effectively applied to analyze growth when constrained by landmark points.

4. Conclusion

Objective, quantitative measurements of regional patterns of growth during cortical folding are essential for understanding gyrification. The methods and data described in this thesis are important steps toward this goal.
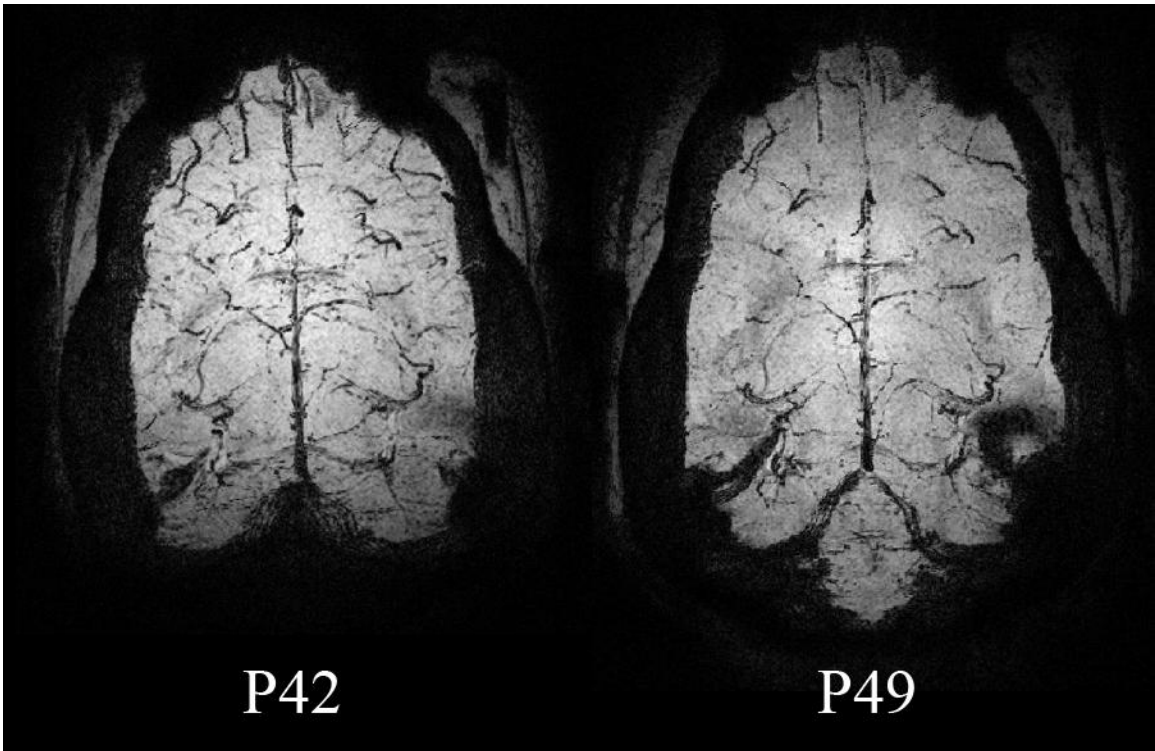
Figure 6.1: Venograms created from gradient echo images acquired on a single ferret at P42 and P49. Blood vessels appear dark within the brain. Image courtesy of Dr. Yulin Chang.
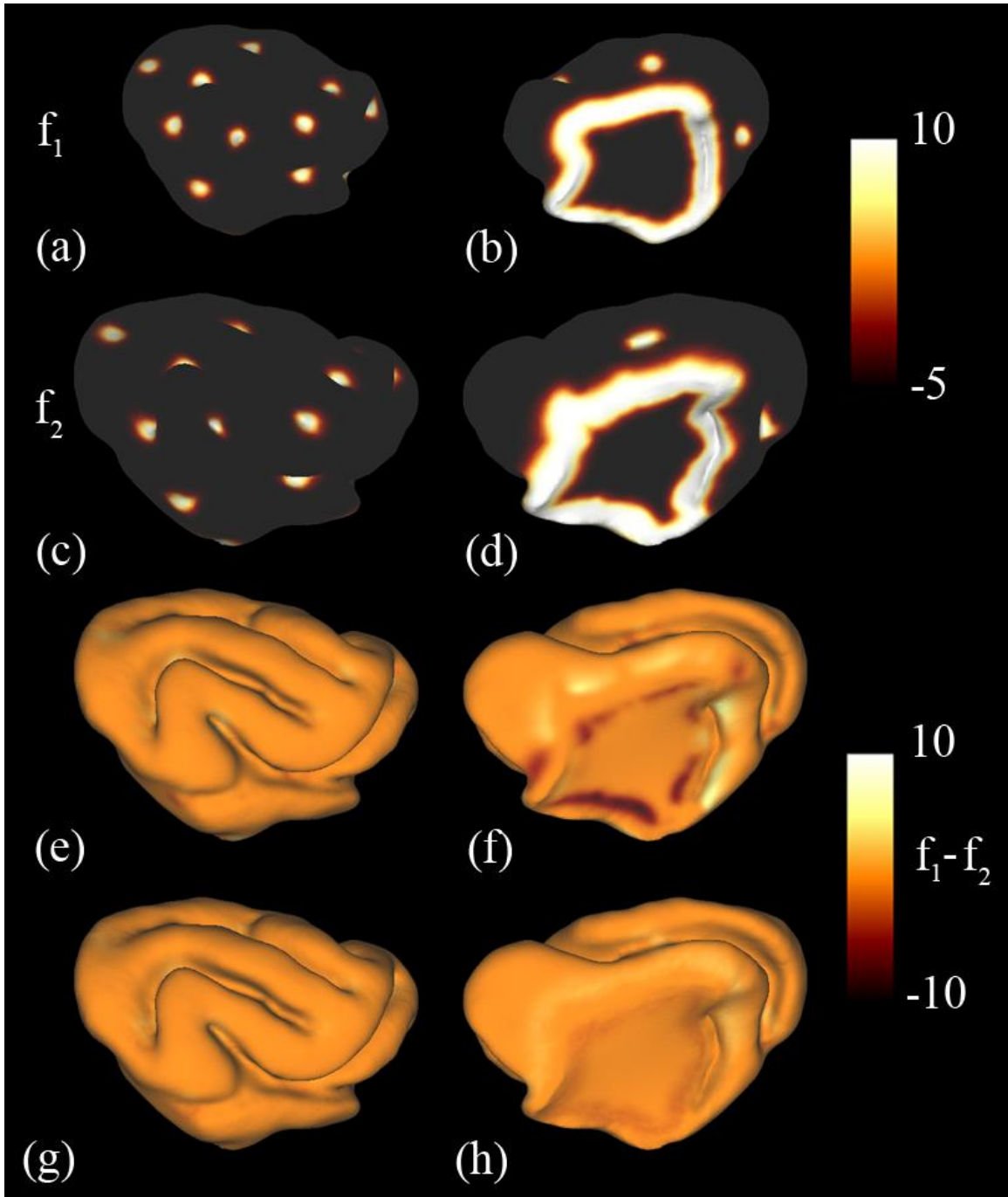
Figure 6.2: The surface matching terms are used to drive the registration process. They are based on mean curvature and distance from the medial wall. (a,b) The surface matching term f_1 plotted on the P14 cortical surface. (c,d) The surface matching term f_2

plotted on the P21 cortical surface (e,f) The difference between f_1 and f_2 for the initial correspondence. (g,h) Difference between f_1 and f_2 after relaxation.
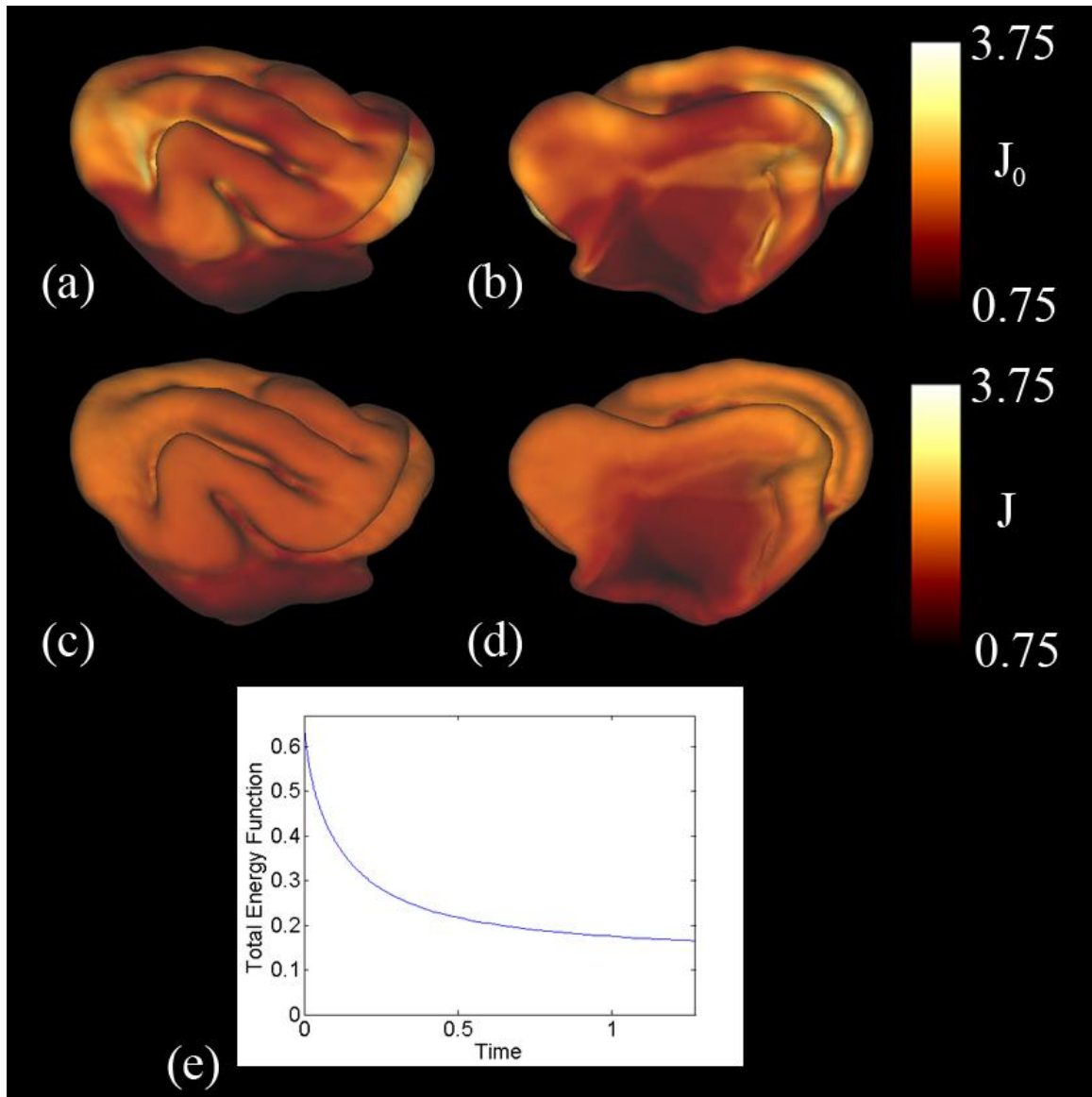
Figure 6.3: Local growth is calculated from the determinant of the deformation gradient tensor between the surfaces. (a,b) Areal expansion after an initial correspondence is determined. (c,d) Areal expansion after the relaxation algorithm is applied. Note that the

color scale is the same for both. (e) The rate of change of the total energy function
decreases as the algorithm proceeds
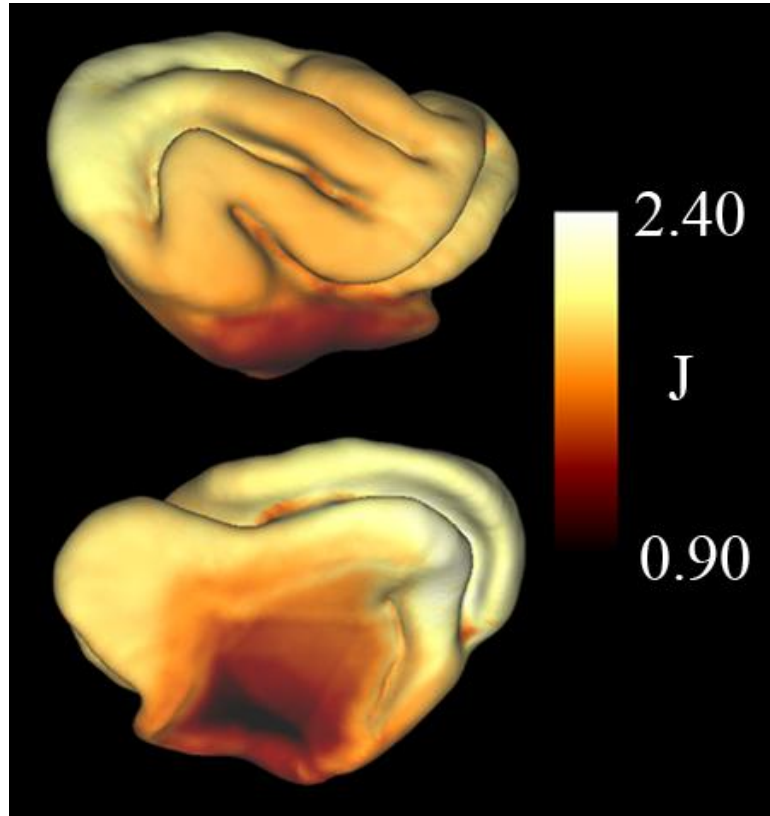
Figure 6.4: The dilatation ratio between the registered P14 and P21 cortical surfaces. The
range of the color scale is reduced from the previous image to highlight differences in
growth between regions.

References

Alford, P. W., and L. A. Taber. 2003. Regional epicardial strain in the embryonic chick heart during the early looping stages. Journal of Biomechanics 36:1135-1141.

Audette, M. A., F. P. Ferrie, and T. M. Peters. 2000. An algorithmic overview of surface registration techniques for medical imaging. Medical Image Analysis 4:201-217.

Axelrad, E. L. 1987. Theory of flexible shells. Amsterdam: North-Holland.

Barnette, A R, J J Neil, C D Kroenke, J L Griffith, A A Epstein, P V Bayly, A K Knutsen, and T E Inder. 2009. Characterization of brain development in the ferret via MRI. Pediatric Research 66 (1):80-84.

Barron, D H. 1950. An experimental analysis of some factors involved in the development of the fissure pattern of the cerebral cortex. Journal of Experimental Zoology 113:553-581.

Basser, P J, and D K Jones. 2002. Diffusion-tensor MRI: theory, experimental design and data analysis - a technical review. NMR in Biomedicine 15:456-467.

Batchelor, P G, A D Castellano Smith, D L Hill, DJ Hawkes, T C Cox, and A F Dean. 2002. Measures of folding applied to the development of human fetal brain. IEEE Transactions on Medical Imaging 21:953-965.

Bayly, P.V., T. S. Cohen, E. P. Leister, D. Ajo, E. C. Leuthardt, and G. M. Genin. 2005. Deformation of the human brain induced by mild acceleration. Journal of Neurotrauma 22 (8):845-856.

Beg, M.F., M.I. Miller, A. Trouve, and L. Younes. 2005. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. International Journal of Computer Vision 61 (2):139-157.

Bizley, J. K., Nodal, F. R., Nelken, I., King, A. J. 2005 Functional organization of ferret auditory cortex. Cerebral Cortex 15:1637-1653.

Bizley, J. K., Nodal, F. R., Bajo, V. M., Nelken, I., King, A. J. 2007.Physiological and anatomical evidence for multisensory interactions in auditory cortex. Cerebral Cortex 17:2172-2189.

Bonnet, J., and R. D. Wood. 2008. Nonlinear continuum mechanics for finite element analysis. 2nd ed. Cambridge: Cambridge University Press.

Bystron, I, C Blakemore, and P Rakic. 2008. Development of the human cerebral cortex: Bounder Committee revisited Nature Reviews Neuroscience 9:110-122.

Chada, S, P Lamoureux, R E Buxbaum, and S R Heidemann. 1997. Cytomechanics of neurite outgrowth from chick brain neurons. Journal of Cell Science 110:1179-1186.

Chi, J G, E C Dooling, and F H Gilles. 1977. Gyral development of the human brain. Annals of Neurology 1:86-93.

Christensen, G.E., and H.J. Johnson. 2001. Consistent Image Registration. IEEE Transactions on Medical Imaging 20 (7):568-582.

Christensen, G.E., R.D. Rabbit, and M.I. Miller. 1994. 3D brain mapping using a deformable neuroanatomy. Physics in Medicine and Biology 39:609-618.

Christensen, G.E., R.D. Rabbit, and M.I. Miller. 1996. Deformable templates using large deformation kinematics. IEEE Transactions on Image Processing 5 (10):1435-1447.

Csernansky, J G, S K Gillespie, D L Dierker, A Anticevic, L Wang, D M Barch, and D C Van Essen. 2008. Symmetric abnormalities in sulcal patterning in schizophrenia. NeuroImage 43:440-446.

Dale, A.M., B Fischl, and M.I. Sereno. 1999. Cortical surface-based analysis. I. Segmentation and surface reconstruction. NeuroImage 9:179-194.

Davatzikos, C. 1997. Spatial transformation and registration of brain images using elastically deformable models. Computer Vision and Image Understanding 66 (2):207-222.

Davies, J. A. 2005. Mechanisms of morphogenesis: the creation of biological form. San Diego: Elsevier.

deIpolyi, A R, P Mukherjee, K Gill, R G Henry, S C Partridge, S Veeraraghavan, H Jin, Y Lu, S P Miller, and D M Ferriero. 2005. Comparing microstructural and macrostructural development of the cerebral cortex in premature newborns: diffusion tensor imaging versus cortical gyration. NeuroImage 27:579-586.

Drury, H.A., D.C. Van Essen, M. Corbetta, and A.Z. Snyder. 1999. Surface-based analysis of the human cerebral cortex. In Brain Warping, edited by A. W. Toga. New York: Academic.

Dubois, J, M Benders, A Cachia, F Lazeyras, L R Ha-Vinh, S V Sizonenko, C Borradori-Tolsa, J F Mangin, and P S Huppi. 2008. Mapping the early cortical folding process in the preterm newborn brain. Cerebral Cortex 18:1444-1454.

Elston, G. N., Benavides-Piccione, R., DeFelipe, J. 2001. The pyramidal cell in cognition: a comparative study in human and monkey. Journal of Neuroscience 21:RC163

Filas, B.A., I. R. Efimov, and L. A. Taber. 2007. Optical coherence tomography as a toll for measuring morphogenetic deformation of the looping heart. The Anatomy Record 290:1057-1068.

Filas, B.A., A.K. Knutsen, P.V. Bayly, and L.A. Taber. 2008. A new method for measuring deformation of folding surfaces during morphogenesis. Journal of Biomechanical Engineering 130:061010-1-9.

Fischl, B, M.I. Sereno, and A.M. Dale. 1999. Cortical surface-based analysis. NeuroImage 9:195-207.

Fischl, B., M.I. Sereno, R.B.H. Tootell, and A.M. Dale. 1999. High-resolution intersubject averaging and a coordinate system for the cortical surface. Human Brain Mapping 8:272-284.

Fornito, A, G S Malhi, J Lagopoulos, B Ivanovski, S J Wood, D Velakoulis, M M Saling, P D McGorry, C Pantelis, and M Yucel. 2007. In vivo evidence for early neurodevelopmental anomaly of the anterior cingulate cortex in bipolar disorder. Acta Psychiatrica Scandinavica 116:467-472.

Gholipour, A, N Kehtarnavaz, R Briggs, M Devous, and K Gopinath. 2007. Brain functional localization: a survey of image registration techniques. IEEE Transactions on Medical Imaging 26 (4):427-451.

Gilbert, S. F. 2003. Developmental Biology. 7th ed. Sunderland, MA: Sinauer Associates.

Glaunes, J., M. Vaillant, and M.I. Miller. 2004. Landmark matching via large deformation diffeomorphisms on the sphere. Journal of Mathematical Imaging and Vision 20:179-200.

Hack, M, and H G Taylor. 2000. Perinatal brain injury in preterm infants and later neurobehavioral function. Journal of the American Medical Association 284 (15):1973-1974.

Hashima, A. R., A. A. Young, A. D. McCulloch, and L. K. Waldman. 1993. Nonhomogeneous analysis of epicardial strain distributions during acute myocardial ischemia in the dog. Journal of Biomechanics 26:19-35.

Haug, H. 1987. Brain sizes, surfaces, and neuronal sizes of the cortex cerebri: a sterological investigation of man and his variability and a comparison with some mammals (primates, whales, marsupials, insectivores, and one elephant). American Journal of Anatomy 180:126-142.

Hill, J.E., D. Dierker, J. Neil, T.E. Inder, A.K. Knutsen, J. Harwell, T. Coalson, and D.C. Van Essen. 2010. A surface-based analysis of hemispheric asymmetries and folding of cerebral cortex in term-born human infants. The Journal of Neuroscience 30 (6):2268-2276.

Hoffman, M A. 1989. On the evolution and gemoetry of the brain in mammals. Progress in Neurobiology 32:137-158.

Holden, M. 2008. A review of geometric transformations for nonrigid body registration. IEEE Transactions on Medical Imaging 27 (1):111-128.

Holzapfel, G. A. 2000. Nonlinear solid mechanics: a continuum approach for engineering. West Sussox, England: John Wiley and Sons.

Hornak, J. P. 1996-2010. The basics of MRI. http://www.cis.rit.edu/htbooks/mri/

Huttenlocher, P. R.1990. Morphometric study of human cerebral cortex development. Neuropsychologia 28(6):517-527

Huttenlocher, P. R. and Dbholkar, A. S. 1997. Regional differences in synaptogenesis in human cerebral cortex. Journal of Comparative Neurology 387:167-178

Huang H., Yamamoto, A., Hossain, M. A., Younes, L., Mori, S. 2008. Quantitative cortical mapping of fractional anisotropy in developing rat brains. Journal of Neuroscience 28:1427-1433.

Inder, T E, S K Warfield, H Wange, P S Huppi, and J J Volpe. 2005. Abnormal cerebral structure is present at term in premature infants. Pediatrics 115 (2):286-294.

Jackson, C A, J D Peduzzi, and T L Hickey. 1989. Visual cortex development in the ferret. I. Genesis and migration of visual cortical neurons. Journal of Neuroscience 9:1242-1253.

Johnson, H.J., and G.E. Christensen. 2002. Consisten landmark and intensity-based image registration. IEEE Transactions on Medical Imaging 21 (5):450-461.

Joshi, A., R. Leahy, A W Toga, and D. Shattuck. 2009. A framework for brain registration via simultaneous surface and volume flow. Paper read at IPMI 2009, LNCS 5636.

Joshi, S., and M.I. Miller. 2000. Landmark matching via large deformation diffeomorphisms. IEEE Transactions on Image Processing 9 (8):1357-1370.

Jovanov-Milosevic, N., Benjak, V., Kostovic, I. 2006 Transient cellular structures in developing corpus callosum of the human brain. Collegium Antropologicum 30(2):375-81

Kindberg, K., M. Karlsson, N. B. Jr. Ingels, and J. C. Criscione. 2007. Nonhomogeneous strain from sparse marker arrays for analysis of transmural myocardial mechanics. Journal of Biomechanical Engineering 129:603-610.

Klein, A., J. Andersson, B. A. Ardekani, J. Ashburner, B. Avants, M. Chiang, G.E. Christensen, D. L. Collins, J. Gee, P. Hellier, J. H. Song, M. Jenkinson, C. Lepage, D. Rueckert, P M Thompson, T. Vercauteren, R. P. Woods, J. J. Mann, and R. V. Parsey. 2009. Evaluation of 14 nonlinear deformation algoirthms applied to human brain MRI registration. NeuroImage 46:786-802.

Kostovic, I, and Jovanov-Milosevic, N. 2006. The development of cerebral connections during the first 20-45 weeks gestation. Seminars in Fetal and Neonatal Medicine (11).

Kostovic, I. and Judas, M. 2002. Correlation between the sequential ingrowth of afferents and transient patterns of cortical lamination in preterm infants. The Anatomical Record 267:1-6

Kreyszig, E. 2006. Advanced engineering mathematics. Hoboken, NJ: John Wiley and Sons.

Kroenke, C D, E N Taber, L A Leigland, A K Knutsen, and P V Bayly. 2009. Regional patterns of cerebral cortical differentiation determined by diffusion tensor MRI. Cerebral Cortex 19:2916-2929.

Kroenke, C D, D.C. Van Essen, T E Inder, S Rees, G L Bretthorst, and J J Neil. 2007. Microstructural changes of the baboon cerebral cortex during gestational development reflected in magnetic resonance imaging diffusion anisotropy. Journal of Neuroscience 27:12506-12515.

Lamoureux, P, R E Buxbaum, and S R Heidemann. 1989. Direct evidence that growth cones pull. Nature 340:159-162.

Leclerc, S. S., Rice, F. L., Kykes, R. W., Pourmoghadam, K., Gomez, C. M. 1993.Electrophysiological examination of the representation of the face in the suprasylvian gyrus of the ferret: a correlative study with cytoarchitecture. Somatosensory and Motor Research 10(2):133-159.

Litke, N., M. Droske, M Rumpf, and P. Schroder. 2005. An image processing approach to surface matching. Paper read at Eurographics Symposium on Geometry Processing, at Vienna, Austria.

Lui, T, D. Shen, and C. Davatzikos. 2004. Deformable registration of cortical structures via hybrid volumetric and surface warping. NeuroImage 22:1790-1801.

Magnotta, V. A., N. C. Andreasen, and et al. 1999. Quantitative measurement of gyrification in the human brain: changes with aging. Cortex 9:151-160.

Manger, P. R., Masiello, I., Innocenti, G. M. 2002. Areal organization of the posterior parietal cortex of the ferret (Mustela putorius). Cerebral Cortex 12:1280-1297.

Manger, P. R., Nakamura, H., Valentiniene, S., Innocenti, G. M. 2004.Visual areas in the lateral temporal cortex of the ferret (Mustela putorius). Cerebral Cortex 14:676-689.

McSherry, G M. 1984. Mapping of cortical histogenesis in the ferret. Journal of Embryology and Experimental Morphology 81:237-252.

McSherry, G M, and I H M Smart. 1986. Cell production gradients in the developing ferret isocortex. Journal of Anatomy 144:1-14.

Miller, M.I., G.E. Christensen, Y. Amit, and U. Grenander. 1993. Mathematical textbook of deformable neuroanatomies. Proceedings on the National Acadamies of Science 90:11944-11948.

Mori, S, and J Zhang. 2006. Principles of diffusion tensor imaging and its applications to basic neuroscience research. Neuron 51:527-539.

Morton, K. W. 2005. Numerical solution of partial differential equations: an introduction. 2 ed. New York: Cambridge University Press.

Neal, J, M Takahashi, M Silva, G Tiao, C A Walsh, and V L Sheen. 2007. Insights into the gyrification of developing ferret brain by magnetic resonance imaging. Journal of Anatomy 210:66-77.

Noctor, S C, N J Scholnicoff, and S L Juliano. 1997. Histogenesis of ferret somatosensory cortex. Journal of Comparative Neurology 387:179-193.

Novozhilov, V. V. 1970. Thin shell theory. Groningen, Netherlands: Wolters-Noordhoff.

Ogden, R. W. 1984. Non-linear elastic deformations. Mineola, NY: Dover Publications.

Oguz, I., J. Cates, T. Fletcher, R. Whitaker, D. Cool, S. Aylward, and M. Styner. 2008. Cortical correspondence using entropy-based particle systems and local features. Proceedings of the 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro:1637-1640.

Osher, S., and J. Sethian. 1988. Fronts propagation with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. Journal of Computational Physics 79 (1):12-49.

Pang, T, R Atefy, and V Sheen. 2008. Malformations of cortical development. The Neurologist 14:181-191.

Petrou, S, T Sach, and L Davidson. 2000. The long-term costs of preterm birth and low birth weight: results of a systematic review. Child: Care, Health and Development 27 (2):97-115.

Postelinicu, G., L. Zollei, and B Fischl. 2009. Combined volumetric and surface registration. IEEE Transactions on Medical Imaging 28 (4):508-522.

Qui, A., M. Albert, L. Younes, and M.I. Miller. 2009. Time sequence diffeomorphic metric mapping and parallel transport track time-dependent shape changes. NeuroImage 45:S51-S60.

Rakic, P. 1988. Specification of cerebral cortical areas. Science 241:170-176.

Rakic, P. 2000. Radial unit hypothesis of neocortical expansion. Novartis Foundation Symposium 228:30-52.

Rakic, P. 2006. A century of progress in corticoneurogenesis: from silver impregnation to genetic engineering. Cerebral Cortex 16:Suppl: i3-17.

Ramasubramamian, A., K. S. Latacha, J. M. Benjamin, D. A. Voronov, A. Ravi, and L. A. Taber. 2006. Computational model for early cardiac looping. Annals of Biomedical Engineering 34:1655-1669.

Ramasubramamian, A., and L. A. Taber. 2008. Computational modeling of morpohogenesis regulated by mechanical feedback. Biomechanics and Modeling in Mechanobiology 7 (2):77-91.

Ramsay, A. M., Meredith, M. A. 2004. Multiple sensory afferents to ferret pseudosylvian sulcal cortex. Neuroreport 15(3):461-465.

Rice, F. L., Gomez, C. M., Leclerc, S. S., Dykes, R. W., Moon, J. S., Pourmoghadam, K. 1993 Cytoarchitecture of the ferret suprasylvian gyrus correlated with areas containing multiunit responses elicited by stimulation of the face. Somatosensory and Motor Research 10(2):161-188.

Richman, D P, R M Stewart, J W Hutchinson, and V S Caviness. 1975. Mechanical model of brain convolutional development. Science 189 (18-21).

Saad, Y., and M.H. Schultz. 1986. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM Journal on Scientific and Statistical Computing 7:856-869.

Shen, D., and C. Davatzikos. 2002. HAMMER: Hierarchical attribute matching mechanism for elastic registration. IEEE Transactions on Medical Imaging 21 (11):1421-1439.

Shen, D., and C. Davatzikos. 2004. Measuring temporal morphological changes robustly in brain MR images via 4-dimensional template warping. NeuroImage 21:1508-1517.

Shi, Y., P.M. Thompson, I. Dinov, Osher, S., and A.W. Toga. 2007. Direct cortical mapping via solving partial differential equations on implicit surfaces. Medical Image Analysis 11:207-223.

Smart, I H M, and G M McSherry. 1986b. Gyrus formation in the cerebral cortex in the ferret. I. Description of the external changes. Journal of Anatomy 146:141-152.

Smart, I H, and G M McSherry. 1986a. Gyrus formation in the cerebral cortex of the ferret. II. Description of the internal histological changes. Journal of Anatomy 147:27-43.

Snyman, J. 2005. Practical mathematical optimization: an intro to basic optimization theory and classical and new gradient-based algorithms. US: Springer.

Soufan, A. T., B. G. van den, P. D. Moerland, M. M. Massink, M. J. van den Hoff, A. F. Moorman, and J. M. Ruijter. 2007. Three-dimensional measurement and visualization of morphogenesis applied to cardiac embryology. Journal of Microscopy 225:269-274.

Soufan, A. T., B. G. Van Den, J. M. Ruijter, P. A. de Boer, M. J. van den Hoff, and A. F. Moorman. 2006. Regionalized sequence of myocardial cell growth and proliferation characterizes early chamber formation. Circulation Research 99:545-552.

Stjernqvist, K, and N W Svenningsen. 1999. Ten-year follow-up of children born before 29 gestational weeks: health, cognitive development, behaviour and school achievement. Acta Peadiatrica 88:557-562.

Szabo, B.A. , and I Babuska. 1991. Finite element analysis. New York: Wiley.

Taber, L.A. 2004. Nonlinear theory of elasticity: Applications in biomechanics. Singapore: World Scientific.

Taber, L. A., H. Sun, E. B. Clark, and B. B. Keller. 1994. Epicardial strains in embryonic chick ventricle at stages 16 through 24. Circulation Research 75:896-903.

Tarui, T, T Takahashi, R S Nowakowski, and V S Caviness. 1999. Sequence of neuron origin and neocortical laminar fate: relation to cell cycle of origin in the developing murine cerebral wall. Journal of Neuroscience 19:10357-10371.

Terzopoulos, D., J. Platt, A. Barr, and K. Fleischer. 1987. Elastically deformable models. Computer Graphics 21 (4):205-214.

Thompson, P M, A D Lee, R A Dutton, J A Geaga, K M Hayashi, M A Eckert, U Bellugi, A M Galaburda, J R Korenberg, D L Mills, A W Toga, and A L Reiss. 2005. Abnormal cortical complexity and thickness profiles mapped in Williams Syndrome. The Journal of Neuroscience 25 (16):4146-4158.

Thompson, P.M., and A.W. Toga. 1996. A surface-based technique for warping three-dimensional images of the brain. IEEE Transactions on Medical Imaging 15 (4):402-417.

Tobita, K., and B. B. Keller. 2000. Maturation of end-systolic stress-strain relations in chick embryonic myocardium. Americal Journal of Physiology 279:H216-H224.

Travis, K., Ford, K., Jacobs, B. 2005 Regional dendritic variation in neonatal human cortex: a quantitative Golgi study. Developmental Neuroscience 27:277-287

Vaillant, M., and J. Glaunes. 2005. Surface matching via currents. Paper read at Information Processing for Medical Imaging, at CO.

Van Essen, D.C. 1997. A tension-based theory of morphogenesis and compact wiring in the central nervous system. Nature 385:313-317.

Van Essen, D.C., D Dierker, A Z Snyder, M E Raichle, A L Reiss, and J Korenberg. 2006. Symmetry of cortical folding abnormalities in Williams Syndrome revealed by surface-based analyses. The Journal of Neuroscience 26 (20):5470-5483.

Van Essen, D.C., H.A. Drury, J. Dickson, J. Harwell, D. Hanlon, and C.H Anderson. 2001. An integrated softweard suite for surface-based analyses of cerebral cortex. Journal of the American Medical Informatics 8 (5):443-459.

Van Essen, D.C., H.A. Drury, S. Joshi, and M.I. Miller. 1998. Functional and structural mapping of human cerebral cortex: Solutions are in the surfaces. Proceedings on the National Acadamies of Science 95:788-795.

Voets, N L, M G Hough, G Douaud, P M Matthews, A James, L Winmill, P Webster, and S Smith. 2008. Evidence for abnormalities of cortical development in adolescent-onset schizophrenia. NeuroImage 43:665-675.

Voigt, T. 1989. Development of glial cells in the cerebral wall of ferrets: direct tracing of their transformation from radial glia into astrocytes. Journal of Comparative Neurology 289:74-88.

Weisenfeld, N. I., and S K Warfield. 2009. Automatic segmentation of newborn brain MRI. NeuroImage 47:564-572.

Welker, W. 1990. Why does cerebral cortex fissure and fold? A review of determinants of gyri and sulci. in Cerebral Cortex (eds E. G. Jones & A. Peters):3-136 (Plenum).

Wisco, J J, G Kuperberg, D Manoach, B T Quinn, E Busa, B Fischl, S Heckers, and A G Sorensen. 2007. Abnormal cortical folding patterns within Broca's area in schizophrenia: Evidence from structural MRI. Schizophrenia Research 94:317-327.

Wolosin, S M, M E Richardson, J G Hennessey, M B Denckla, and S H Mostofsky. 2009. Abnormal cerebral cortex structure in children with ADHD. Human Brain Mapping 30:175-184.

Woods, R. P. 2003. Characterizing volume and surface deformations in an atlas framework: theory, applications, and implementation. NeuroImage 18:769-788.

Xu, G, P Bayly, and L Taber. 2009. Residual stress in the adult mouse brain. Biomechanics and Modeling in Mechanobiology 8:253-262.

Xu, G, A K Knutsen, K Dikranian, C D Kroenke, P V Bayly, and L A Taber. 2010. Axons pull on the brain, but tension does not drive cortical folding. Journal of Biomechanical Engineering. (*in press*)

Xue, H., D. Shen, B. Karacali, J. Stern, D. Rottenberg, and C. Davatzikos. 2006. Simulating deformations of MR brain images for validation of atlas-based segmentation and registration algorithms. NeuroImage 33:855-866.

Xue, H., L. Srinivasan, S. Jiang, M. Rutherford, A. D. Edwards, D. Rueckert, and J. Hajnal. 2007. Longitudinal cortical registration for developing neonates. Paper read at MICCAI 2007, Part II, LNCS.

Yanovsky, I., C. L. Gyuader, A. Leow, P M Thompson, and L. Vese. 2008. Nonlinear elastic registration with unbiased regularization in three dimensions. The MIDAS Journal 549:56-67.

Yeo, B., M. Sabuncu, T. Vercauteren, N. Ayache, B Fischl, and P. Golland. 2008. Spherical demons: fast surface registration. Paper read at MICCAI 2008, Part I, LNCS 5241.

Zamir, E. A., A. Czirok, A. Cui, C. D. Little, and B. J. Rongish. 2006. Mesodermal cell displacements during avian gastulation are due to both individual cell-autonomous and convective tissue movements. Proceedings on the National Acadamies of Science 103:19806-19811.

Zamir, E. A., A. Czirok, B. J. Rongish, and C. D. Little. 2005. A digital image-based method for computational tissue fate mapping during early avian morphogenesis. Annals of Biomedical Engineering 33:854-865.

Zerhouni, E. A., D. M. Parish, W. J. Rogers, A. Yang, and E. P. Shapiro. 1988. Human heart tagging with MR imaging - a method for noninvasive assessment of myocardial motion. Radiology 169:59-63.

Zhang, Y., M. Brady, and S. Smith. 2001. Segmentation of brain MR images through a hidden Markov random field model and the expectation maximization algorithm. IEEE Transactions on Medical Imaging 20 (1):45-57.

Appendix

1. LIGASE[3]

LIGASE is a semi-automated seed-based segmentation algorithm that identifies that boundary of the GM and WM in preterm human infants. The approach is based on the idea that all of the WM is connected and should be fairly similar in intensity with gradients in intensity existing between different neighboring tissue types. Another assumption is that the intensities of the WM voxels are well represented by an asymmetric Gaussian function.

Using a histogram of intensities from the image, the user identifies an approximate peak of the WM and then takes an "educated guess" as to where the function decays to 0 (i.e., where the likelihood of an intensity being WM decreases to 0) (Fig. A.1). These three values are used to create an asymmetric Gaussian function that describes the likelihood that a voxel is WM based entirely of its intensity. Because determining the range of the function is guess work, in practice, we have the user selected a range of values for both the max and min, and then run the algorithm using each of these values. Using the Matlab function *gradient*, the gradient of the image is calculated, from which the magnitude is calculated. The magnitude of the gradient will be large near the boundary of tissue types and small within the WM.

To begin, a seed point is selected by the user that is known to be WM. The algorithm looks at each of the neighboring voxels in turn and asks three questions: 1. what is the probability based on intensity that this voxel is WM? 2. Is the magnitude of the gradient below the cutoff value? 3. Is the pair-wise difference in intensity between the

---

[3] LIGASE was co-developed with Jason Hill, and is published in Hill et al., 2010.

voxel and the seed voxel bellow the cutoff value? Based on the answer to each question, the algorithm classifies the voxel as one of three options; not WM, WM but not a voxel to "search from", and WM and a voxel to "search from". If a voxel is added to the list of those to "search from", then that voxels neighbors are checked to see whether or not they are WM.

The cutoff values were determined empirically and vary depending on the probability based on intensity that the voxel is WM. So, a voxel that is very likely to be WM will have higher cutoff values than a voxel that is not likely to be WM. As I noted earlier, a range of values are selected by the user to determine the max and min of the Gaussian function. The algorithm is run using each of these values, so the output of LIGASE is an array of segmentation volumes. I wrote a graphical user interface (GUI) in Matlab that allows the user to look at each of the segmentation volumes and select which of the parameters work best. The user is also able to pick and choose between parts of each of the segmentation volumes to create the final segmentation volume. Here, the Matlab function *roipoly* is utilized. Header (.hdr) and image (.img) files are created from the final segmentation volume, which can then be imported to a number of software packages, including CARET, for any necessary manual editing. LIGASE is not perfect, so the segmentation volume does require some manually editing. However, LIGASE reduces the amount of time required to complete a segmentation volume considerably.

Fig. A.4 show the results of LIGASE applied to a number of acquired images at different stages of development. On average, LIGASE takes around 20-60 seconds to run through using a single set of parameters. The code is freely available online at (http://brainvis.wustl.edu/~LIGASE) and is provided in section 3.1.
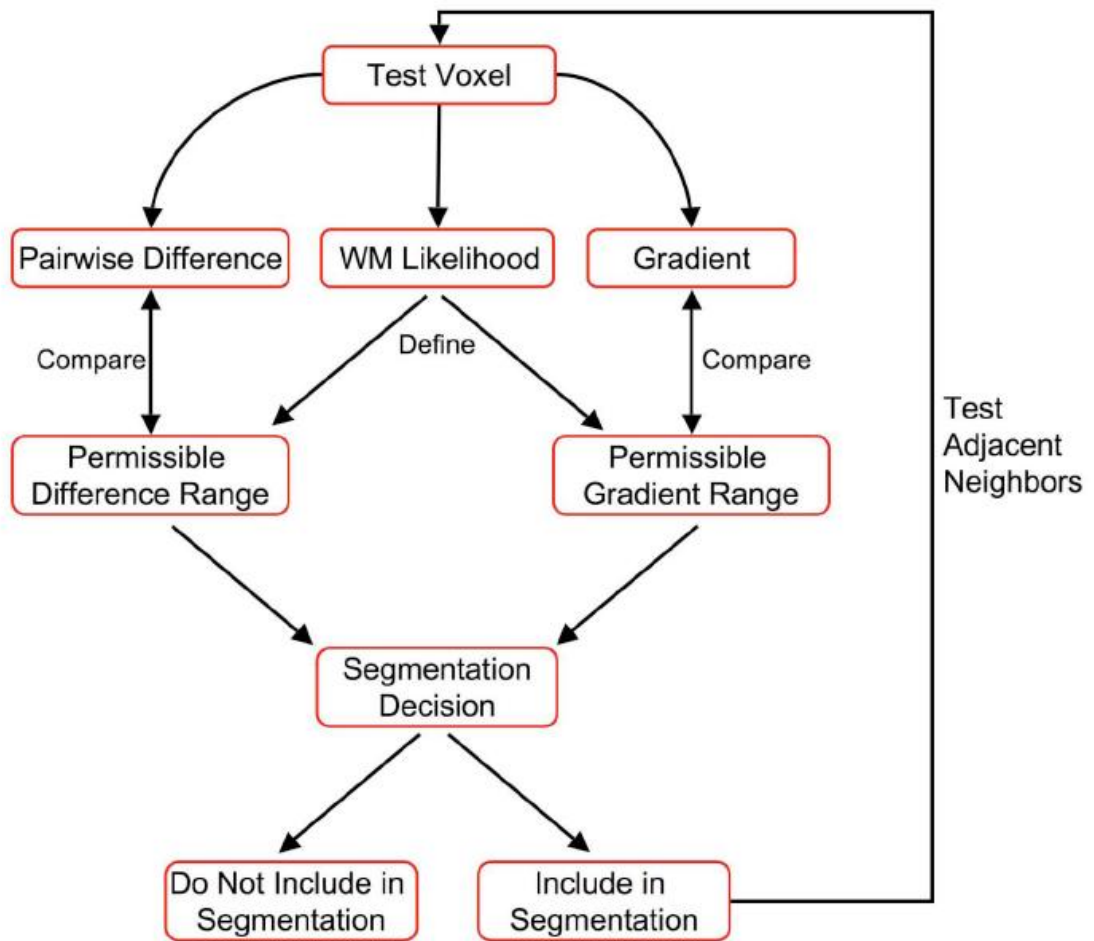
Figure A.1: LIGASE algorithm: Starting with a voxel that is known to be WM, all neighboring voxels are examined in turn (test voxel). The likelihood that a test voxel is WM determines the permissible range of values for the pairwise difference and magnitude of the gradient. If these values are within the allowed range the test voxel is included in the segmentation. This figure is reprinted with permission (Hill et al., 2010).
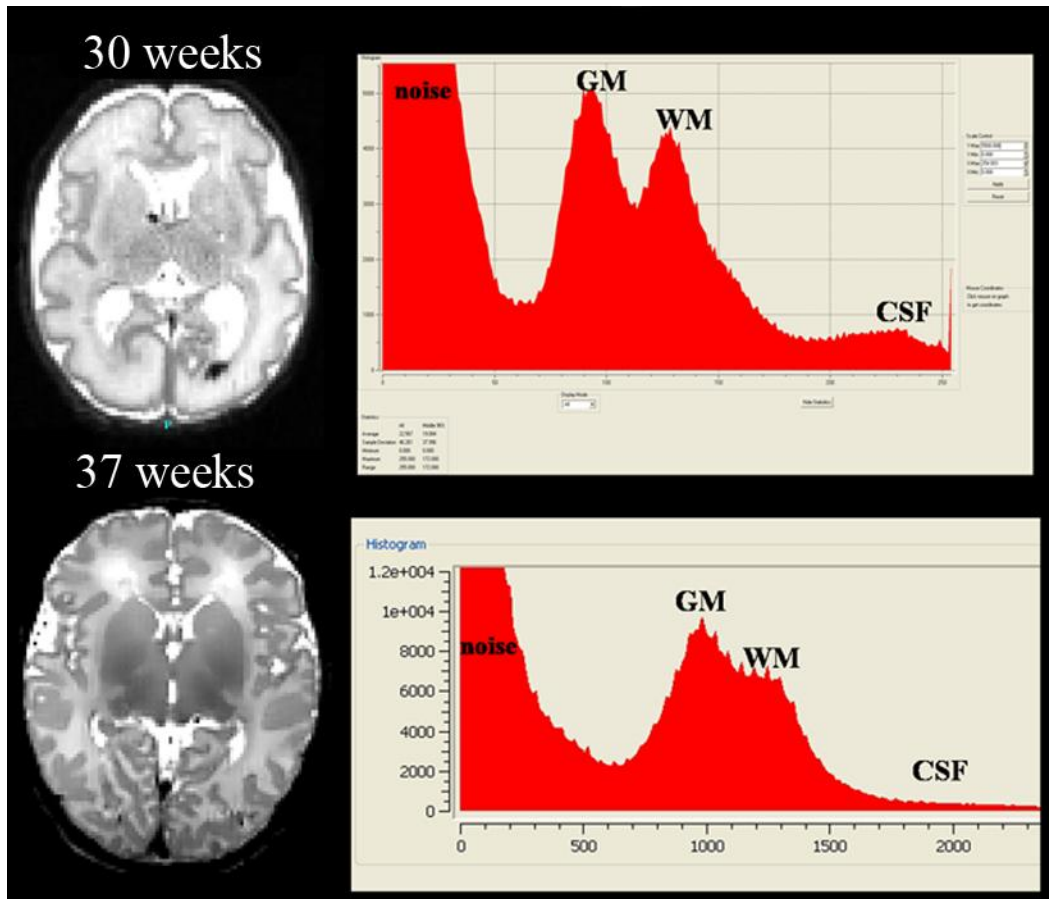
Figure A.2: Histograms of T2-weighted MRI acquired at 30 and 37 weeks GA: While some overlap exists, GM and WM peaks are clearly separated at 30 weeks GA. Closer to term equivalent the contrast decreases between GM and WM due to myelination of the WM.
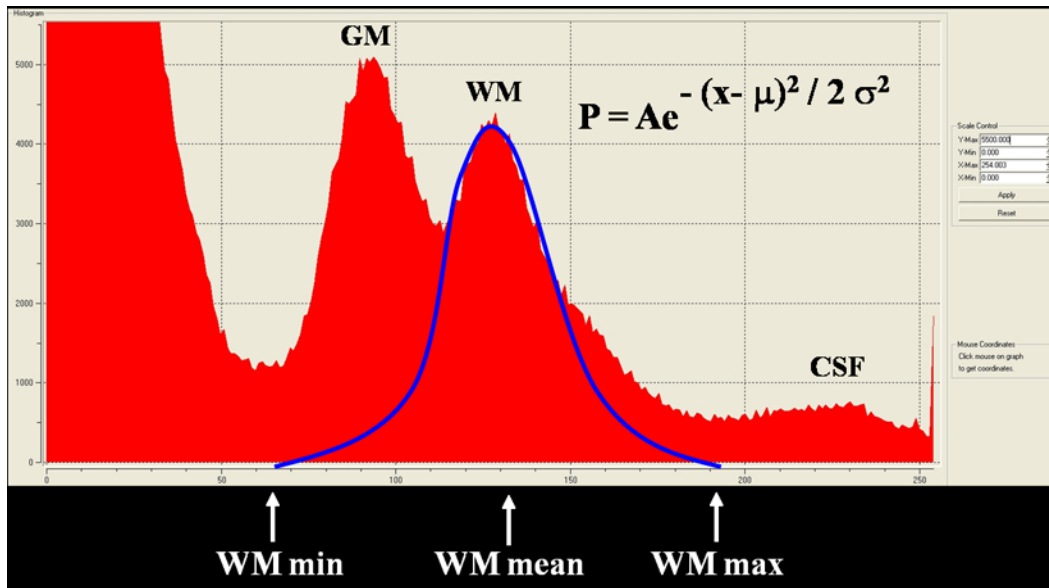
Figure A.3: The user identifies the intensity range of the WM. A Gaussian function is used to describe how likely a voxel is WM for a given intensity. In practice, the user identifies a range of possible minimum and maximum values, and the algorithm is run with each combination.
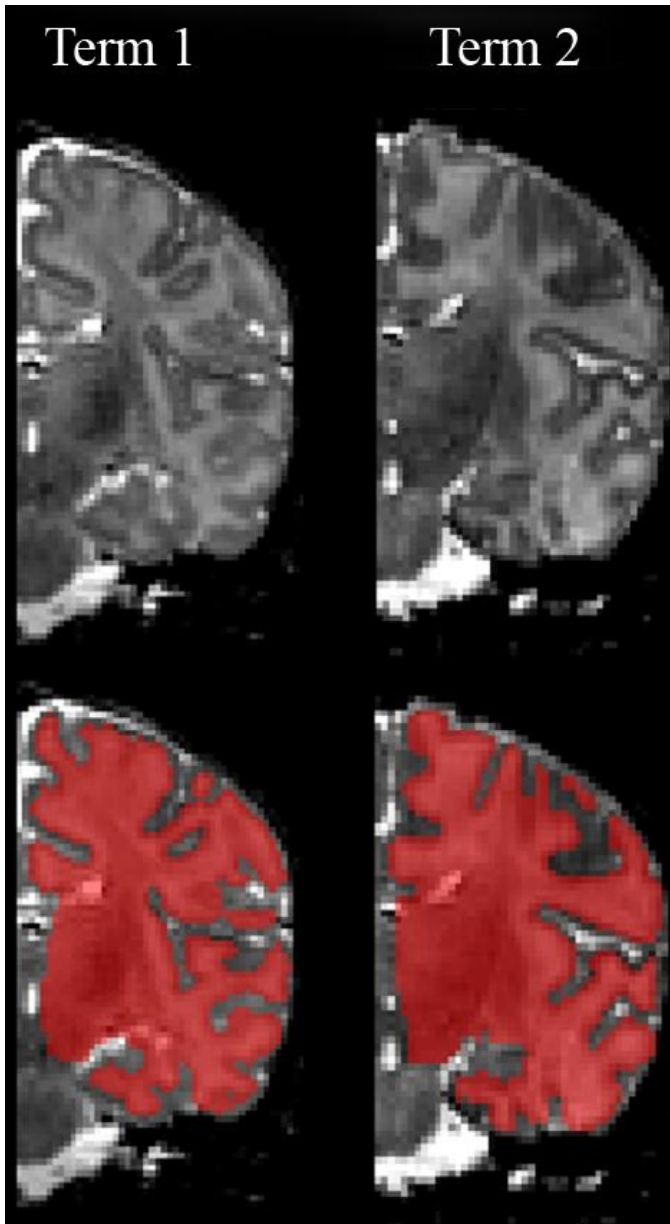
Figure A.4: Final segmentations of two infants at term equivalent: LIGASE was applied to segment the anatomical images. Afterwards the segmentation volumes were manually edited, and then dilated by one voxel so that the boundary of the segmentation volume was within the cortex. This image is reprint with permission (Hill et al., 2010).

2. Matlab code

2.1 *LIGASE*

```matlab
function seg4D=LIGASE(anat_mat,initial_pt)

% Written by: Andrew Knutsen, Jason Hill
% Developed: Spring 2006
% Updated: July 2009


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                    Description                    %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% We have developed an algorithm to segment images of premature through
% term infants obtained using MRI. The goal of the algorithm is to
% locate the grey matter (GM) and white matter (WM) boundary.  Briefly, the
% algorithm works by starting with a single voxel that the user identifies
% as WM. Then the program uses intensity information (WM probably based
% off of histogram, local gradient, piece-wise intensity difference) to
% determine whether the neighboring voxels are also WM. It then searches
% again from those voxels identified as WM, and continues to grow from
% there until no neighbors are identified as WM. Note that some variables
% within this function are specific to each anatomy volume and therefore
% must be adjusted each time a new volume is to be segmented. This
% algorithm assumes that the intensities of the anatomy volume have been
% scaled from 0 to 255.

% A more complete description of the method is available in:

% Jason Hill, Donna Dierker, Jeffrey Neil, Terrie Inder, Andy Knutsen, John
% Harwell, and David Van Essen. "A surface-based analysis of hemispheric
% asymmetries and folding of cerebral cortex in term-born human infants."


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                    Variables                    %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Input Variables
% anat_mat - Anatomy volume to be segmented
% initial_pt - Voxel within anat_mat that lies within WM

% Output Variables
% seg4D - Array of segmentation volumes generated by this algorithm



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%       Variables that should be changed with each use       %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Mean of the distribution based on histogram. Value should be between 0
% and 255
```

```matlab
wm_mean = 120;

% User estimated lower and upper bounds of where the left hand side of the
% distribution approaches 0 based on a histogram of intensities. _low is
% the lower bound estimate, _high is the upper bound estimate and _step
% determines number of estimates. Values should be whole numbers between
% 0 and 255.

wm_min_low = 70;
wm_min_high = 100;
wm_min_step = 5;

% User estimated lower and upper bounds of where the right hand side of the
% distribution approaches 0 based on a histogram of intensities. _low is
% the lower bound estimate, _high is the upper bound estimate and _step
% determines number of estimates. Values should be whole numbers between
% 0 and 255.

wm_max_low = 145;
wm_max_high = 155;
wm_max_step = 10;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                Begin Code               %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

wm_mean = wm_mean/255;

wm_min_low = wm_min_low/255;
wm_min_high = wm_min_high/255;
wm_min_step = wm_min_step/255;

wm_max_low = wm_max_low/255;
wm_max_high = wm_max_high/255;
wm_max_step = wm_max_step/255;

num_wm_min_vals = round((wm_min_high - wm_min_low)/wm_min_step) + 1;
num_wm_max_vals = round((wm_max_high - wm_max_low)/wm_max_step) + 1;

% Rescale volume intensities from 0->1
anat_mat = anat_mat/max(max(max(anat_mat)));
slice = anat_mat;

[imax jmax kmax]=size(slice);

lmax = num_wm_min_vals*num_wm_max_vals;

% Initialize variables
seg4D = zeros(imax, jmax, kmax, lmax);
il = 1;
wm_min = wm_min_low;

for m = 1:num_wm_min_vals
  wm_max = wm_max_low;
    for n = 1:num_wm_max_vals
```

```matlab
        % Calculate standard deviations for both sides of probability function (for assymetric distribution
they are different).
        sig1=sqrt(-(wm_min-wm_mean)^2/log(.1/.9)/2);
        sig2=sqrt(-(wm_max-wm_mean)^2/log(.1/.9)/2);

        % Probability at 1 standard deviation (P1) and 2 (P2)
        P1 = 0.5459;
        P2 = 0.1218;

        % Define global cutoff values
        difference_cutoff = .12*(wm_max - wm_min); % cutoff value for local gradient
        grad_mag_cutoff = 0.05; % cutoff value for global gradient

        % Initialize variables for this iteration
        seg = zeros(imax, jmax, kmax);

        x = zeros(imax*jmax*kmax,1);
        y = zeros(imax*jmax*kmax,1);
        z = zeros(imax*jmax*kmax,1);

        x(1)=initial_pt(2);
        y(1)=initial_pt(1);
        z(1)=initial_pt(3);

        i=1;
        counter = 1;

        % Calculate the gradient of the volume anat_mat, and scale from 0->1
        [fx fy fz] = gradient(slice,1);
        grad_mag = sqrt(fx.^2+fy.^2+fz.^2);
        grad_mag = grad_mag/max(max(max(grad_mag)));

        % Set initial point equal to 1
        seg(x(i),y(i),z(i))=1;

        tic
        % Start at voxel [x(i) y(i) z(i)]
        while i<=counter

            % Check whether voxel [x(i) y(i) z(i)] is on the edge of the image volume
            arangemin=min(1,x(i)-1);
            brangemin=min(1,y(i)-1);
            crangemin=min(1,z(i)-1);

            arangemax=min(1,imax-x(i));
            brangemax=min(1,jmax-y(i));
            crangemax=min(1,kmax-z(i));

            % Look at all neighbors of voxel [x(i) y(i) z(i)] and test whether neighbors should be included in
segmentation
            for a=-arangemin:arangemax;
              for b=-brangemin:brangemax;
                for c=-crangemin:crangemax;
                  if seg(x(i)+a,y(i)+b,z(i)+c)>0

                      % Voxel has already been identified as WM
```

193

```matlab
        else

            % Calculate intensity difference between voxel [x(i) y(i) z(i)] and its neighbor [x(i)+a
y(i)+b z(i)+c]

            diff=abs(slice(x(i),y(i),z(i))-slice(x(i)+a,y(i)+b,z(i)+c));

            % Gradient value at voxel [x(i)+a y(i)+b z(i)+c]
            grad=grad_mag(x(i)+a,y(i)+b,z(i)+c);

            % Check to see whether voxel neighbor intensity is above or below wm_mean
            if slice(x(i)+a,y(i)+b,z(i)+c)<=wm_mean % intensity < mean

                % Calculate probably value P using sig1
                P=.9*exp(-(slice(x(i)+a,y(i)+b,z(i)+c)-wm_mean)^2/(2*sig1^2));

                if P<=P2 % More than 2 standard deviations from the mean
                    if diff < difference_cutoff*.50
                        if grad < grad_mag_cutoff*.5

                            % Include voxel in segmentation
                            seg(x(i)+a,y(i)+b,z(i)+c)=1;
                        else

                            % Include voxel in segmentation
                            seg(x(i)+a,y(i)+b,z(i)+c)=1;
                        end
                    else

                        % Do not include in segmentation volume
                    end

                elseif P>P2 && P<=P1 % Between 1 and 2 standard deviations from the mean
                    if diff < difference_cutoff*.75
                        if grad < grad_mag_cutoff*.5

                            % Include voxel in segmentation
                            seg(x(i)+a,y(i)+b,z(i)+c)=1;
                        else

                            % Include voxel in segmentation
                            seg(x(i)+a,y(i)+b,z(i)+c)=1;
                        end
                    else

                        % Do not include in segmentation volume
                    end
                elseif P>P1 % Within 1 standard deviation of the mean
                    if diff < difference_cutoff*1.5
                        if grad < grad_mag_cutoff*1.5

                            % Include voxel in segmentation
                            seg(x(i)+a,y(i)+b,z(i)+c)=1;

                            % Add to list of voxels to search from
                            counter = counter+1;
```

194

```matlab
                            x(counter)=x(i)+a;
                            y(counter)=y(i)+b;
                            z(counter)=z(i)+c;
                        else

                            % Include voxel in segmentation
                            seg(x(i)+a,y(i)+b,z(i)+c)=1;
                        end

                    else

                        % Include voxel in segmentation
                        seg(x(i)+a,y(i)+b,z(i)+c)=1;
                    end
                end
        elseif slice(x(i)+a,y(i)+b,z(i)+c)>wm_mean % intensity > mean

            % Calculate probably value P using sig2
            P=.9*exp(-(slice(x(i)+a,y(i)+b,z(i)+c)-wm_mean)^2/(2*sig2^2));

            if P<=P2 % More than 2 standard deviations from the mean
                if diff < difference_cutoff
                    if grad < grad_mag_cutoff

                        % Include voxel in segmentation
                        seg(x(i)+a,y(i)+b,z(i)+c)=1;

                        % Add to list of voxels to search from
                        counter = counter+1;

                        x(counter)=x(i)+a;
                        y(counter)=y(i)+b;
                        z(counter)=z(i)+c;
                    else

                        % Include voxel in segmentation
                        seg(x(i)+a,y(i)+b,z(i)+c)=1;
                    end
                else

                    % Include voxel in segmentation
                    seg(x(i)+a,y(i)+b,z(i)+c)=1;
                end
            elseif P>P2 && P<=P1 % Between 1 and 2 standard deviations from the mean
                if diff < difference_cutoff*1.50
                    if grad < grad_mag_cutoff*1.50
                        % Include voxel in segmentation
                        seg(x(i)+a,y(i)+b,z(i)+c)=1;

                        % Add to list of voxels to search from
                        counter = counter+1;

                        x(counter)=x(i)+a;
                        y(counter)=y(i)+b;
                        z(counter)=z(i)+c;
                    else
```

195

```matlab
                        % Include voxel in segmentation
                        seg(x(i)+a,y(i)+b,z(i)+c)=1;
                    end
                else
                    % Include voxel in segmentation
                    seg(x(i)+a,y(i)+b,z(i)+c)=1;
                end
            elseif P>P1 % Within 1 standard deviation of the mean
                if diff < difference_cutoff*1.5
                    if grad < grad_mag_cutoff*1.5

                        % Include voxel in segmentation
                        seg(x(i)+a,y(i)+b,z(i)+c)=1;

                        % Add to list of voxels to search from
                        counter = counter+1;

                        x(counter)=x(i)+a;
                        y(counter)=y(i)+b;
                        z(counter)=z(i)+c;
                    else

                        % Include voxel in segmentation
                        seg(x(i)+a,y(i)+b,z(i)+c)=1;
                    end
                else

                    % Include voxel in segmentation
                    seg(x(i)+a,y(i)+b,z(i)+c)=1;
                end
            end
        end
    end
end
i=i+1;
end
toc

% Assign segmentation to 4D volume
seg4D(:,:,:,il) = seg;

% Increase index for next segmentation
il =il+1;

% Use next wm_max estimate
wm_max = wm_max + wm_max_step;
end
% Use next wm_min estimate
wm_min = wm_min + wm_min_step;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                    End Code                      %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

2.2 *strains*

```matlab
function [L,varargout]=strains(XYZ,xyz,NS,varargin)

% Written by Andrew K Knutsen, Dr. Phil Bayly

%%%%%%%%%%%%%%%%%%%%%%%%%%% Describe Code
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% [L,R,V,E] = strains(XYZ,xyz,NS,dist,n1,n2,ns,fv)
% varargin => [dist,n1,n1,ns,fv)
% varargout => [L,R,V,Ettn]

% Output Variables
% L - Principal Stretch Ratios W.R.T. Deformed And Undeformed Geometry (Lamda)
% R - Residuals: N,t1,t2,n.  And number points used in fit - num_pts
% V - Stretch Ratios In n1,n2,shear Directions
% E - Physical Lagrangian And Eulerian Principal Strains (E1,E2,e1,e2)

% Input Variables
% XYZ - Individual Surface Coordinates: [X Y Z]
% xyz - Atlas Surface Coordinates: [x y z]
% NS - Normal Vectors To Individual Surface: [NX NY NZ]
% dist - Radius Of Points To Be Used In Modeling [Default Value == 1]
% n1 - Deformed Curvature Vectors [n1x n1y n1z] To Calculate Stretch/Strain
% n2 - Deformed Curvature Vectors [n2x n2y n2z] To Calculate Stretch/Strain
% ns - Normal Vectors To Atlas Surface: [nx ny nz]
% fv - Structure For Plotting Strains - If On, 1st And 2nd Principal
%      Strains Will Be Plotted On Surface fv

%%%%%%%%%%%%%%%%%%%%%%%% Define Input Variables
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

n_in = nargin;

switch n_in
    case 3
        dist = 1;
        ns = [];
        n1 = [];
        n2 = [];
        vectors = 0;
        plotsurf = 0;
    case 4
        dist = cell2mat(varargin(1));
        ns = [];
        n1 = [];
        n2 = [];
        vectors = 0;
        plotsurf = 0;
    case 6
        dist = cell2mat(varargin(1));
        ns = [];
        n1 = cell2mat(varargin(2));
        n2 = cell2mat(varargin(3));
```

```matlab
            vectors = 1;
            plotsurf = 0;
        case 7
            dist = cell2mat(varargin(1));
            n1 = cell2mat(varargin(2));
            n2 = cell2mat(varargin(3));
            ns = cell2mat(varargin(4));
            vectors = 1;
            plotsurf = 0;
        case 8
            dist = cell2mat(varargin(1));
            n1 = cell2mat(varargin(2));
            n2 = cell2mat(varargin(3));
            ns = cell2mat(varargin(4));
            vectors = 1;
            plotsurf = 1;
            FV.vertices = XYZ;
            fv = cell2mat(varargin(5));
        otherwise
            disp('Wrong number of inputs')
            return
end

if isempty(ns)
    ns = ones(size(xyz));
end

if isempty(n1)
    vectors = 0;
end

%%%%%%%%%%%%%%%%%% Check Input Variable Orientation
%%%%%%%%%%%%%%%%%%%%%%%
% Force XYZ To Be N Rows By 3 Columns (X,Y,Z)
if size(XYZ,1) == 3
    XYZ = XYZ';
elseif size(XYZ,2) == 3

else
    error('XYZ must be matrix N rows by 3 columns: (X,Y,Z)')
end

% Force xyz To Be N Rows By 3 Columns (X,Y,Z)
if size(xyz,1) == 3
    xyz = xyz';
elseif size(xyz,2) == 3

else
    error('xyz must be matrix N rows by 3 columns: (x,y,z)')
end

% Force NS To Be N Rows By 3 Columns (NX,NY,NZ)
if size(NS,1) == 3
    NS = NS';
elseif size(NS,2) == 3
```

```matlab
else
    error('NS must be matrix N rows by 3 columns: (NX,NY,NZ)')
end

% Force ns To Be N Rows By 3 Columns (nx,ny,nz)
if size(ns,1) == 3
    ns = ns';
elseif size(ns,2) == 3

else
    error('ns must be matrix N rows by 3 columns: (nx,ny,nz)')
end

% Normalize Surface Normal Vectors
Mag = sqrt(NS(:,1).^2+ NS(:,2).^2+NS(:,3).^2);
NS = [NS(:,1)./Mag NS(:,2)./Mag NS(:,3)./Mag];
mag = sqrt(ns(:,1).^2+ns(:,2).^2+ns(:,3).^2);
if mag == 0
    mag = 1;
end
ns = [ns(:,1)./mag ns(:,2)./mag ns(:,3)./mag];

XS=XYZ(:,1);YS=XYZ(:,2);ZS=XYZ(:,3);

% Set Matrices To Assign Output
RN = zeros(length(XS),1);Rt1 = zeros(length(XS),1);
Rt2 = zeros(length(XS),1);Rn = zeros(length(XS),1);

num_pts = zeros(length(XS),1);

Lamda_1 = zeros(length(XS),1);Lamda_2 = zeros(length(XS),1);
lamda_1 = zeros(length(XS),1);lamda_2 = zeros(length(XS),1);
lamda_n1 = zeros(length(XS),1);lamda_n2 = zeros(length(XS),1);
gamma = zeros(length(XS),1);
E1 = zeros(length(XS),1);E2 = zeros(length(XS),1);
e1 = zeros(length(XS),1);e2 = zeros(length(XS),1);
NS_adj = zeros(length(XS),3);
U1 = zeros(length(XS),3);U2 = zeros(length(XS),3);

%%%%% Assign Normal And Tangential Coordinates - Individual Surface %%%%%

tic
for i=1:length(XS),

    % Make Unit Normal And Tangents,
    X0 = XYZ(i,1);Y0 = XYZ(i,2);Z0 = XYZ(i,3);
    x0 = xyz(i,1);y0 = xyz(i,2);z0 = xyz(i,3);

    e_N=NS(i,:)';

    imax = find(abs(e_N)==max(abs(e_N)),1,'first');
    switch imax
        case 1
            mag1=sqrt(e_N(3)^2+e_N(2)^2);
            if mag1==0,e_T1=[0;0;1];
            else e_T1 = [0;-e_N(3);e_N(2)]/mag1;
```

```matlab
            end
        case 2
            mag2=sqrt(e_N(3)^2+e_N(1)^2);
            if mag2==0,e_T1=[0;0;1];
            else e_T1 = [-e_N(3);0;e_N(1)]/mag2;
            end
        case 3
            mag3=sqrt(e_N(1)^2+e_N(2)^2);
            if mag3==0,e_T1=[1;0;0];
            else e_T1 = [-e_N(2);e_N(1);0]/mag3;
            end
end;
e_T2 = cross(e_N,e_T1);

% Reject Pts With Normals In Opposite Direction (Opposing Faces)
dir_chk = e_N'*NS';
e_n = ns(i,:)';
dir_chk2 = e_n'*ns';

% Find Points in the Neighborhood
DX=XS-X0;DY=YS-Y0;DZ=ZS-Z0;
DR = sqrt(DX.^2+DY.^2+DZ.^2)';

% Keep Remaining Points
keep_pts = find(dir_chk >= 0 & dir_chk2 >= 0 & DR <= dist);

if length(keep_pts) >= 8

    XX=XS(keep_pts);YY=YS(keep_pts);ZZ=ZS(keep_pts);

    uno = zeros(length(keep_pts),1);
    DXYZ = [XX-X0,YY-Y0,ZZ-Z0];

    % Tangential And Normal Components - Individual Surface
    T1 = (e_T1'*DXYZ')';T2 = (e_T2'*DXYZ')';N = (e_N'*DXYZ')';

    %%%%% Assign Normal And Tangential Coordinates - Atlas Surface %%%%%%

    % fit curves for deformed vertices
    x = xyz(keep_pts,1);y = xyz(keep_pts,2);z = xyz(keep_pts,3);

    dxyz = [x-x0,y-y0,z-z0]; % diff b/w def coords and pt of interest
    t1 = (e_T1'*dxyz')';t2 = (e_T2'*dxyz')';n = (e_N'*dxyz')';

    %%%%%%%%%%%%%%%%%%%%%%%%%%% Model Surface
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    % Fitting Polynomial
    alpha = [uno T1 T2 (1/2)*T1.^2 (1/2)*T2.^2 T1.*T2];

    % Coefficients For Fit N=N(T1,T2),t1=t1(T1,T2),t2=t2(T1,T2),n=n(T1,T2)
    a = alpha\N;b = alpha\t1;c = alpha\t2;d = alpha\n;

    % Modeled Data
    warning off all
    dN = alpha*a;dt1 = alpha*b;dt2 = alpha*c;dn = alpha*d;
```

```matlab
rN = dN - N;rt1 = dt1 - t1;rt2 = dt2 - t2;rn = dn - n;
RN(i,:) = sum(rN.^2)/sum(N.^2);Rt1(i,:) = sum(rt1.^2)/sum(t1.^2);
Rt2(i,:) = sum(rt2.^2)/sum(t2.^2);Rn(i,:) = sum(rn.^2)/sum(n.^2);

% Number Of Points Used In The Fit
num_pts(i,:) = length(keep_pts);

% compute derivatives N,t1,t2,n wrt T1 and T2 at (T10,T20)
dNdT1 = a(2);
dNdT2 = a(3);
dt1dT1 = b(2);
dt1dT2 = b(3);
dt2dT1 = c(2);
dt2dT2 = c(3);
dndT1 = d(2);
dndT2 = d(3);

% Describe Transformation Matrix
T_mat = [e_T1 e_T2 e_N]';

%%%%%%%%%%%%%%%%%%%%%%% Calculate Strain
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Calculate Base Vectors
G1 = [1,0,dNdT1];
G2 = [0,1,dNdT2];

% Calculate New Normal Vector From G1 x G2
G3 = cross(G1,G2);
G3 = G3./sqrt(dot(G3,G3));

% Output Normal Vectors From Modeled Surface
NS_adj(i,:) = inv(T_mat)*G3';

% Compute Contravariant Components For Undeformed Base Vectors
G = [G1' G2' G3'];

G_cont = inv(G)';

% Calculate Deformed Base Vectors
g1 = [dt1dT1 dt2dT1 dndT1];
g2 = [dt1dT2 dt2dT2 dndT2];

g3 = cross(g1,g2);
g3 = g3./sqrt(dot(g3,g3));

g = [g1' g2' g3'];

% Calculate Deformation Matrix
F = g*G_cont';

C = F'*F;
B = F*F';

if vectors == 1
```

```matlab
    % Normalize n1 And n2 At Vertex i
    N1 = n1(i,:)./sqrt(dot(n1(i,:),n1(i,:)));
    N2 = n2(i,:)./sqrt(dot(n2(i,:),n2(i,:)));

    % Transform N1,N2 Into Local Coordinate System
    N1_local = (T_mat*N1')';
    N2_local = (T_mat*N2')';

    % Calculate Stretch Ratio In Direction n1/n2 Curvature
    % Assumes N1 And N2 Are W.R.T Deformed Geometry
    lamda_n1(i) = (N1_local*inv(B)*N1_local').^-.5;
    lamda_n2(i) = (N2_local*inv(B)*N2_local').^-.5;

    gamma(i) = N1_local*(eye(3) - lamda_n1(i)*lamda_n2(i)*inv(B))*N2_local';

end

% Calculate Stretch Ratios/Direction W.R.T Deformed Surace
[Ud,omega] = eig(inv(B));

omega = diag(omega);

lamda = omega.^-.5;

%%%%%%%%%%% Find Max/Min Stretch Ratios %%%%%%%%%%%%

% Take Dot Product Of Normal Vector With Eigenvectors
ii = abs(g3*Ud);

% Gives In Plane Indices
rm_d = find(ii < max(ii));

lamda_1(i) = max(lamda(rm_d));
lamda_2(i) = min(lamda(rm_d));

% Calculate Stretch Ratios/Direction W.R.T. Undeformed Surface

[U,Omega] = eig(C);

Omega = diag(Omega);

Lamda = sqrt(Omega);

% Take Dot Product Of Normal Vector With Eigenvectors
mm = abs(G3*U);

% Gives In Plane Indices
rm_u = find(mm < max(mm));

Lamda_1(i) = max(Lamda(rm_u));
Lamda_2(i) = min(Lamda(rm_u));
%       Lamda_1(i) = max(abs(Lamda(rm_u)));
%       Lamda_2(i) = min(abs(Lamda(rm_u)));

% Calculate Lagrangian/Eularian Strain
```

```matlab
            e1(i) = 1/2*(1 - lamda_1(i)^-2);
            e2(i) = 1/2*(1 - lamda_2(i)^-2);

            E1(i) = 1/2*(Lamda_1(i)^2 - 1);
            E2(i) = 1/2*(Lamda_2(i)^2 - 1);

            % Vectors corresponding to E1 and E2
            if Lamda(rm_u(1))==Lamda(rm_u(2))
                U1(i,:) = inv(T_mat)*U(:,rm_u(1));
                U2(i,:) = inv(T_mat)*U(:,rm_u(2));
            else
                U1(i,:) = inv(T_mat)*U(:,Lamda==max(Lamda(rm_u)));
                U2(i,:) = inv(T_mat)*U(:,Lamda==min(Lamda(rm_u)));
            end

        else % Too Few Points For Fit

            % Set Output Values For Not Enough Points In Fit
            RN(i,:) = NaN;Rt1(i,:) = NaN;Rt2(i,:) = NaN;Rn(i,:) = NaN;
            num_pts(i,:) = 0;

            Lamda_1(i) = NaN;
            Lamda_2(i) = NaN;

            lamda_1(i) = NaN;
            lamda_2(i) = NaN;

            if vectors == 1
                lamda_n1(i) = NaN;
                lamda_n2(i) = NaN;
                gamma(i) = NaN;
            end

            NS_adj(i,:) = zeros(1,3);

            E1(i) = NaN;
            E2(i) = NaN;

            U1(i,:) = 0;
            U2(i,:) = 0;

            e1(i) = NaN;
            e2(i) = NaN;

        end

        % Progress Of Script
        if i==round(length(XS)/3)
            disp('script 33% finished')
        elseif i==round(length(XS)*2/3)
            disp('script 67% finished')
        elseif i==length(XS)
            disp('script finished')
        end
end
```

```matlab
%%%%%%%%%%%%%% Define Output Variables - Set varargout
%%%%%%%%%%%%%%%%%

% Create Structures For Output
L = struct('d1',lamda_1,'d2',lamda_2,'U1',Lamda_1,'U2',Lamda_2);
V = struct('n11',lamda_n1,'n22',lamda_n2,'n12',gamma);
E = struct('E1',E1,'E2',E2,'e1',e1,'e2',e2,'U1',U1,'U2',U2);
R = struct('N',RN,'t1',Rt1,'t2',Rt2,'n',Rn,'num_pts',num_pts,'NS_adj',NS_adj);

n_out = nargout;

switch n_out
    case 2
        varargout = {R};
    case 3
        varargout = {R,V};
    case 4
        varargout = {R,V,E};
end

toc

%%%%%%%%%%%%%%%%%%%%%%%%% Create Surface Plots
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if plotsurf == 1

    % Plot 1st Principal Strain
    figure
    subplot(1,2,1)
    patch(fv,'FaceVertexCData',E.E1,'FaceColor','interp','EdgeColor','none');
    title({'1st Principal Strain';['Distance = ',mat2str(dist)]});
    xlabel('x'),ylabel('y'),zlabel('z')
    view(3);daspect([1 1 1]),grid on
    h1 = gcf;set(h1,'color',[1 1 1]);
    caxis([floor(min(min([E.E1 E.E2]))),ceil(max(max([E.E1 E.E2])))]);colorbar

    % Plot 2nd Principal Strain
    subplot(1,2,2)
    patch(fv,'FaceVertexCData',E.E2,'FaceColor','interp','EdgeColor','none');
    title({'2nd Principal Strain';['Distance = ',mat2str(dist)]});
    xlabel('x'),ylabel('y'),zlabel('z')
    view(3);daspect([1 1 1]),grid on
    h2 = gcf;set(h2,'color',[1 1 1]);
    caxis([floor(min(min([E.E1 E.E2]))),ceil(max(max([E.E1 E.E2])))]);colorbar
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%
```

2.3 *interp_F0*

```matlab
function f=interp_F0(x,y,z,IX,IY)

load defgrad_F0

N=length(x);

X = sph.vertices(:,1);
Y = sph.vertices(:,2);
Z = sph.vertices(:,3);

ix = IX(1);
iy = IY(1);

f = zeros(1,N);

for it=1:N,
    dist = sqrt((x(it)-X).^2 + (y(it)-Y).^2 + (z(it)-Z).^2);
    [dist,isort]=sort(dist);
    if dist(1)==0,
        f(1,it) = F0(ix,iy,isort(1));
    else
        f(1,it) = (F0(ix,iy,isort(1))/dist(1) + F0(ix,iy,isort(2))/dist(2) + F0(ix,iy,isort(3))/dist(3)) / ...
            (1/dist(1)+1/dist(2)+1/dist(3));
    end
end
```

2.4 *project_to_surf*

```matlab
function [vert,face] = project_to_surf(v,fv)

% Written by Andy K Knutsen
% For use with LACROSS Registration Algorithm

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                 Description                  %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% This function projects coordinates in space onto a surface, and outputs
% the projected coordinates along with the corresponding face index.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                 Variables                    %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Input variables
% v - coordinates in space to be projected
% fv - surface (faces and vertices) that the coordinates v are to be
%   projected to

% Output variables
% vert - projected coordinates onto surface
```

```matlab
% face - face that each coordinate is projected onto

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%              Begin Code              %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

vert = zeros(size(v,1),3);
face = zeros(size(v,1),1);

XYZ = fv.vertices;

for ix = 1:size(v,1)
   xyz = v(ix,:);

   diff = [xyz(:,1)-fv.vertices(:,1) xyz(:,2)-fv.vertices(:,2) xyz(:,3)-fv.vertices(:,3)];
   dist = sqrt(dot(diff',diff')');

   [B,IX] = sort(dist);

   [irow,icol] = find(fv.faces == IX(1));

   ind = fv.faces(irow,:);

   iy = 1;
   while iy <= size(ind,1)

      % Create vectors of face boundaries
      v1 = XYZ(ind(iy,2),:) - XYZ(ind(iy,1),:);
      v2 = XYZ(ind(iy,3),:) - XYZ(ind(iy,1),:);

      v3 = XYZ(ind(iy,3),:) - XYZ(ind(iy,2),:);

      % Normal vector to face ind(iy,:)
      ns = cross(v1,v2);
      ns = ns/sqrt(dot(ns,ns));

      % Calculate component in normal direction
      w0 = xyz-XYZ(ind(iy,1),:);

      wn = dot(w0,ns)*ns;

      xf = XYZ(ind(iy,1),:) + w0 - wn;

      % Check for crossover
      u1 = xf - XYZ(ind(iy,1),:);
      u2 = xf - XYZ(ind(iy,2),:);
      u3 = xf - XYZ(ind(iy,3),:);

      C1 = sign(dot(cross(u1,v1),cross(u2,v3)));
      C2 = sign(dot(cross(u1,v1),cross(u3,-v2)));
      C3 = sign(dot(cross(u2,v3),cross(u3,-v2)));

      % If no crossover, the output results
      if C1 == 1 && C2 == 1 && C3 == 1
         vert(ix,:) = xf;
         face(ix) = irow(iy);
```

```matlab
        iy = length(ind);

    elseif iy == size(ind,1)
        vert(ix,:) = xf;
        face(ix) = irow(iy);
    end
    iy = iy + 1;
  end
end

end
```

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                End Code                 %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

2.5 *surf_to_surf*

```matlab
function V = surf_to_surf(FV,fv,vert,face)

% Written by Andy K Knutsen
% For use with LACROSS Registration Algorithm

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                Description              %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% This function projects coordinates in space onto a surface, and outputs
% the projected coordinates along with the corresponding face index.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                Variables               %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Input variables
% FV - surface with known coordinate - projecting from this surface
% fv - surface where coordinate location is desired - projecting to this
%   surface

% Output variables
% vert - projected coordinates onto surface
% face - face that each coordinate is projected onto

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                Begin Code               %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

V = zeros(size(vert,1),3);

for ix = 1:length(vert)

    % Vertex to be projected
    f = face(ix);
```

```matlab
    v0 = vert(ix,:);

    % Create vectors out of face boundaries
    x1 = FV.vertices(FV.faces(f,1),:);
    x2 = FV.vertices(FV.faces(f,2),:);
    x3 = FV.vertices(FV.faces(f,3),:);

    v1 = x2 - x1;
    v2 = x3 - x1;

    v3 = cross(v1,v2);

    M = [v1' v2' v3'];

    u1 = v0 - x1;

    coefs = inv(M)*u1';

    % Create vectors out of face boundaries for the surface to project to
    X1 = fv.vertices(fv.faces(f,1),:);
    X2 = fv.vertices(fv.faces(f,2),:);
    X3 = fv.vertices(fv.faces(f,3),:);

    V1 = X2 - X1;
    V2 = X3 - X1;

    % Let v1 and v2 form a basis on FV, and V1 and V2 form a basis on fv.
    % The coordinate on face f on fv is given by
    Xf = X1 + coefs(1)*V1 + coefs(2)*V2;

    V(ix,:) = Xf;
end

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                End Code                 %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

2.6 *refine_mesh*

```matlab
function [XYZ,FACES] = refine_mesh(V,F)

% Written by Andrew K Knutsen
% For use with LACROSS Registration Algorithm

xyz = zeros(length(F)*15,3);
faces = zeros(length(F)*16,3);

% Parse each face into 16 new faces
for ix = 1:length(F)

    x1 = V(F(ix,1),:);
```

```matlab
    x2 = V(F(ix,2),:);
    x3 = V(F(ix,3),:);

    x4 = x1 + (x2-x1)/2;
    x5 = x2 + (x3-x2)/2;
    x6 = x1 + (x3-x1)/2;

    x7 = x1 + (x4-x1)/2;
    x8 = x6 + (x4-x6)/2;
    x9 = x1 + (x6-x1)/2;

    x10 = x2 + (x4-x2)/2;
    x11 = x2 + (x5-x2)/2;
    x12 = x5 + (x4-x5)/2;

    x13 = x5 + (x6-x5)/2;
    x14 = x3 + (x5-x3)/2;
    x15 = x3 + (x6-x3)/2;

    v = [x1;x2;x3;x4;x5;x6;x7;x8;x9;x10;x11;x12;x13;x14;x15];

    [th ph r] = cart2sph(v(:,1),v(:,2),v(:,3));
    r = ones(size(r));

    [v(:,1) v(:,2) v(:,3)] = sph2cart(th,ph,r);

    f = [1 7 9;7 4 8;7 8 9;9 8 6;6 13 15;8 13 6;8 12 13;4 12 8;4 10 12; ...
        10 2 11;12 10 11;12 11 5;12 5 13;13 5 14;13 14 15;15 14 3] + (ix-1)*15;

    iy = (ix-1)*15+1;
    iz = ix*15;
    xyz(iy:iz,:) = v;

    iyy = (ix-1)*16+1;
    izz = ix*16;
    faces(iyy:izz,:) = f;
end

% XYZ = xyz;
% FACES = faces;

% Find unique coordinates, faces
[XYZ,IX] = unique(round(1000*xyz)/1000,'rows','first');
FACES = faces;

xyz = round(xyz*1000)/1000;

for ix = 1:length(XYZ),
  ir = find(XYZ(ix,1)==xyz(:,1) & XYZ(ix,2)==xyz(:,2) & XYZ(ix,3)==xyz(:,3));

  for iy = 1:length(ir)
    FACES(faces==ir(iy)) = ix;
  end
end
```

2.7 *smoothing_filter*

```matlab
function [c,IR] = smoothing_filter(FV,C,iter,stren,IR)

% Written by Andrew K Knutsen
% For use with LACROSS Registration Algorithm

%%% Begin Code %%%

F = FV.faces;
c = zeros(size(C));

if isempty(IR)
    IR = ones(length(C),30);
    for ix = 1:length(C)
        IR(ix,:) = ix;

        % Find connected vertices

        [irow,icol] = find(F == ix); %#ok<NASGU>

        ir = unique(F(irow,:));
        IR(ix,1:length(ir)) = ir;

        CC = C(ir);

        meanC = mean(C(ix)-CC);

        c(ix) = C(ix) - stren*meanC;
    end

    iter = iter - 1;

    while iter >= 1
        for ix = 1:length(C)

            C = c;

            ir = unique(IR(ix,:));

            CC = C(ir);

            meanC = mean(C(ix)-CC);

            c(ix) = C(ix) - stren*meanC;
        end
        iter = iter - 1;
    end
else
    for it = 1:iter
        for ix = 1:length(C)

            ir = unique(IR(ix,:));

            CC = C(ir);
```

```matlab
        meanC = mean(C(ix)-CC);

        c(ix) = C(ix) - stren*meanC;
    end
    C = c;
  end
end
```