Visualization and Analysis of Diffusion Tensor Fields

Gordon Kindlmann

UUCS-04-014

School of Computing
University of Utah
Salt Lake City, UT 84112 USA

September 28, 2004

Abstract

The power of medical imaging modalities to measure and characterize biological tissue is amplified by visualization and analysis methods that help researchers to see and understand the structures within their data. Diffusion tensor magnetic resonance imaging can measure microstructural properties of biological tissue, such as the coherent linear organization of white matter of the central nervous system, or the fibrous texture of muscle tissue. This dissertation describes new methods for visualizing and analyzing the salient structure of diffusion tensor datasets. Glyphs from superquadric surfaces and textures from reaction-diffusion systems facilitate inspection of data properties and trends. Fiber tractography based on vector-tensor multiplication allows major white matter pathways to be visualized. The generalization of direct volume rendering to tensor data allows large-scale structures to be shaded and rendered. Finally, a mathematical framework for analyzing the derivatives of tensor values, in terms of shape and orientation change, enables analytical shading in volume renderings, and a method of feature detection important for feature-preserving filtering of tensor fields. Together, the combination of methods enhances the ability of diffusion tensor imaging to provide insight into the local and global structure of biological tissue.

CONTENTS

ABSTRACT iv

LIST OF FIGURES ix

LIST OF TABLES xiv

ACKNOWLEDGMENTS xv

CHAPTERS

1. INTRODUCTION 1

1.1 Motivation 1
1.2 Overview 2
1.3 Contributions 4
1.4 Data and Methods 5

2. BACKGROUND 6

2.1 Mathematical Background 6
2.1.1 Tensor Algebra Basics 7
2.1.2 Matrix Representations 8
2.1.3 Trace, Determinant, Contraction, Norm 9
2.1.4 Eigenvalues and Eigenvectors 10
2.1.5 Tensors as Vectors 11
2.1.6 Third-order tensors 13
2.1.7 Group Action ψ, Tensor Shape, and Invariants 13
2.1.8 Measurement by Convolution 14
2.2 Diffusion Tensors and their Acquisition 18
2.2.1 MRI basics 18
2.2.2 Modeling Diffusion 19
2.2.3 Diffusion Tensor MRI 21
2.2.4 Diffusion in Nervous Tissue 22
2.3 Quantifying Tensor Shape 23
2.3.1 Invariants Towards Eigenvalues 23
2.3.2 Eigenvalue Wheel 25
2.3.3 Anatomical Significance of Tensor Shape 28
2.3.4 Barycentric Shape Metrics 31
2.3.5 Discussion 33

3. GLYPHS AND TEXTURES 34

3.1 Superquadric Tensor Glyphs 34
3.1.1 Eigenvector Calculation 34
3.1.2 Introduction and Related Work 35

3.1.3 Motivation . . . 37
3.1.4 Method . . . 38
3.1.5 Results . . . 41
3.1.6 Discussion . . . 46
3.2 Reaction-Diffusion Textures for Visualization . . . 47
3.2.1 Introduction and Related Work . . . 47
3.2.2 Turing's Reaction-Diffusion Formulation . . . 49
3.2.3 Tuning Textures with Tensors . . . 52
3.2.4 Discussion . . . 54

4. FIBER TRACKING WITH TENSORLINES . . . 56

4.1 Introduction . . . 56
4.2 Method . . . 59
4.3 Results . . . 62
4.4 Discussion . . . 63

5. INVARIANT GRADIENTS AND ROTATION TANGENTS . . . 67

5.1 Orthonormal Invariant Gradients . . . 69
5.2 Extended Definition of $\{\widehat{\boldsymbol{\nabla}}\mu_1, \widehat{\boldsymbol{\nabla}}\mu_2, \widehat{\boldsymbol{\nabla}}\alpha_3\}$ . . . 73
5.3 Measuring Shape Gradients in Tensor Fields . . . 74
5.4 Rotations and Tangents to SO_3 . . . 80
5.5 Derivatives of ψ: Rotation Tangents . . . 82
5.6 Space of Tensor Orientation . . . 85
5.7 Orthonormal Rotation Tangents . . . 87
5.8 Measuring Orientation Changes in Tensor Fields . . . 87
5.9 Discussion . . . 91

6. DIRECT VOLUME RENDERING . . . 92

6.1 Introduction and Related Work . . . 92
6.2 Barycentric Color and Opacity Maps . . . 93
6.3 Lit-Tensors . . . 93
6.4 Volume Rendering with Tensor Gradients . . . 99
6.5 Discussion . . . 105

7. ORTHOGONAL TRACT DELINEATION . . . 107

7.1 Analysis of Tensor Gradient Components . . . 108
7.2 Application to a Second Dataset . . . 114
7.3 Discussion . . . 115

8. CONCLUSIONS . . . 118

8.1 Future Work . . . 118

APPENDICES

A. INDEX NOTATION . . . 122

B. ORTHOGONALITY OF μ_1, μ_2, α_3 . . . 125

C. RAY-TRACING SUPERQUADRICS 128

D. COMPUTING INVARIANT GRADIENTS 131

REFERENCES 133

LIST OF FIGURES

1.1 Diffusion tensor components 2

1.2 Slice-based tensor visualizations 3

2.1 Convolving discretely sampled values with a continuous kernel generates a continuous signal. Note that the mathematical definition of convolution involves replacing the discrete sample values with scaled Dirac delta functions in the continuous domain. 15

2.2 Effect of kernel choice (left) on reconstruction result (right), for four different kernels. Hann windowed sinc (d) has a support ten samples wide. 16

2.3 Effect of derivative kernel choice (left) on derivative measurement result (right), for the same four kernels used in Figure 2.2. The original data values and value reconstruction results (convolving with the un-differentiated kernel) are shown in gray. 17

2.4 Effect of second derivative kernel choice (left) on second derivative measurement result (right), for two of the kernels used in Figure 2.2. 18

2.5 Diagram of cubic characteristic polynomial (in gray), its roots, and the invariants required for determining the roots with Equation 2.29. 25

2.6 Comparison of tensor shape diagrams. 27

2.7 Visualizations of degrees of freedom of tensor shape, with eigenvalue wheel (top) and superquadric glyphs (bottom). The eigenvalue wheel location, radius, and orientation correspond to tensor size, anisotropy, and anisotropy type, respectively. 29

2.8 Barycentric space of tensor shapes. 32

2.9 Plots of eigenvalue variance μ_2 and skewness α_3 in barycentric shape space. 32

3.1 Tensor shapes, with cuboids. 37

3.2 Tensor shapes, with cylinders. 38

3.3 Tensor shapes, with ellipsoids. 39

3.4 From some viewpoints, ellipsoids poorly convey tensor shape. 39

3.5 Superquadrics defined by Equation 3.2. The gray triangle in the background indicates the subset of the shape space employed by superquadric tensor glyphs. Edges indicate the tessellation resulting from uniform steps in ϕ and θ. 40

3.6 Parameterization change across the linear/planar seam, from $c_l > c_p$ to $c_l < c_p$ ($\gamma = 3$). 41

3.7 Tensor shapes, with superquadric glyphs, and three different values of edge sharpness parameter γ. 42

3.8 Superquadrics convey shape differences more reliably than ellipsoids ($\gamma = 3$). 43

3.9 Slice of DT-MRI dataset of brain visualized with ellipsoids (top) and superquadrics (bottom). 44

3.10 Three-dimensional region of DT-MRI dataset of brain visualized with ellipsoids (top) and superquadrics (bottom). 45

3.11 Slice of DT-MRI dataset of canine myocardium (left ventricle), using ellipsoids (left) and superquadrics (right). Colormap encodes fiber orientation relative to sampling plane. Figure courtesy of Dr. Daniel Ennis. 46

3.12 Results from Equation 3.10 on a 128×128 grid, using $\Delta t = 0.5$, $\Delta x = 1.0$, $d_a = 0.25$, $d_b = 0.0625$, $s = 0.0125$, and the specified values of α and β. The amount of morphogen a is shown in grayscale. In all cases, number of iterations is 20000. 51

3.13 Demonstration of lowering the Euler time step Δt. If Δt is too large, the system diverges (left-most image). If it converges, it tends to converge to the same texture, assuming that the number of iterations N is large enough. 51

3.14 Synthetic tensor dataset of a slice of torus, visualized by ellipse glyphs (a), and a reaction-diffusion texture (b). Thresholding the texture (c) confirms that the spots have the correct aspect ratio. 52

3.15 Portion of coronal slice of DT-MRI dataset, including the ventricles in the lower-left, visualized by ellipse glyphs (a), and a reaction-diffusion texture (b). 53

3.16 Reaction-diffusion texture-based visualization. 54

4.1 Barycentric histogram of a diffusion tensor MRI dataset. The coordinates correspond to the amount of linear, planar and spherical anisotropy in the tensor. The lack of clustering suggests considerable partial voluming is taking place throughout the volume. 57

4.2 Visualization of hyperstreamline (red) and tensorline (yellow) propagation though anisotropic (left, right and bottom) and isotropic regions (top middle) in synthetic data. The directions of the major eigenvectors are indicated with arrows. Note that the tensorline continues along its present course, despite encountering a region of isotropic diffusion. 58

4.3 Hyperstreamline advection through a nearly isotropic region. The S-shaped path is an artifact of the noise in the region and conveys misleading information about the direction of flow through the field. 58

4.4 Redistribution of uniform sample resulting from anisotropic diffusion tensor. Particles have a higher probability of being diffused in the $\mathbf{e}_1$ (major eigenvector) direction of the ellipsoid. 59

4.5 Redistribution of uniform sample resulting from nearly isotropic diffusion tensor. Probability of particle diffusion is approximately the same in every direction. 59

4.6 Remapping of unit vector through four different tensors. Each row gives the geometric interpretation of applying a different tensor (left matrix and left ellipsoid) to the same example vector (middle vector and middle circle) and the resultant transformed vector (right vector). The gray axis of the ellipses is the direction the major eigenvector and is the direction in which a hyperstreamline would be propagated; in contrast, the dark arrow on the right is the diffusion modulated direction. Our method uses a combination of these two terms to produce more stable propagation paths through isotropic regions. 60

4.7 Comparison of tensorlines (solid yellow) and hyperstreamline (core is dashed red, border is gray). Note the tensorlines continue with only minor fluctuations through the isotropic region in the center of the figure, whereas the hyperstreamline is diverted into an S-shaped path. 62

4.8 Comparison of behavior of standard $\mathbf{e}_1$-based tractography (a) and tensorlines (b) in the corpus callosum and corona radiata. At voxels with a significant planar component, the direction of $\mathbf{e}_1$ can change suddenly, while tensorlines continue on an anatomically plausible path. 64

4.9 Examples of major white matter tracts visualized by tensorlines. Note superquadric glyphs populating the cutting plane surfaces. 65

5.1 Schematic view of shape and orientation changes around a given tensor value $\mathbf{D}$. The orbit $SO_3(\mathbf{D})$ contains orientation changes, and shape changes are all orthogonal to the orbit. 67

5.2 Eigenvalues and invariants in 90×90 tensor synthetic image. The value of α_3 is essentially noise when μ_2 is at or near zero. 77

5.3 Display of synthetic tensor image with superquadric glyphs. Image was downsampled to 36×36 tensor samples for better visibility of individual glyphs. Viewpoint is off-axis for better depiction of the difference between linear and planar anisotropy, as well as glyph orientation. 78

5.4 Gradient magnitude in synthetic tensor image, measured in different ways. The grayscale values have been inverted (darker means higher gradient magnitude) for better display of features at low gradient magnitudes. 79

5.5 The space of orientation changes, as represented by $\mathbf{P}$, is reduced from three to two dimensions when eigenvalue skewness α_3 is at extremum, and is reduced to zero dimensions when eigenvalue variance μ_2 is zero. 86

5.6 Layout and invariants in 90×90 tensor synthetic image. The range of α_3 along vertical axis is $-1/\sqrt{2}$ to $1/\sqrt{2}$, as determined by linearly varying Θ from $\pi/3$ to 0. Rotation around eigenvectors occurs in three distinct regions. 88

5.7 Display of synthetic tensor image with superquadric glyphs. Image was downsampled to 30×30 tensor samples for better visibility of individual glyphs. 89

5.8 Gradient magnitudes in synthetic tensor image. Darker grays indicate higher values. Quantization ranges for (d), (e), and (f) are equal, to permit comparison of the different magnitudes by their gray values. 90

6.1 Examples of barycentric opacity maps and resulting volumes. 94

6.2 Examples of barycentric color maps and resulting renderings. 95

6.3 Sequence of volumes of differing anisotropy, rendered with lit-tensors. Anisotropy varies gradually between the nine volumes, going in scanline order. 98

6.4 Lit-tensor model shading on the whole brain, with the same opacity and light direction as first image in Figure 6.2. 99

6.5 Mixing between lit-tensor and opacity gradient shading, for a synthetic sphere (left), and a portion of brain data (right). Going from top to bottom, the contribution of lit-tensors is 1.00, 0.66, 0.33, and 0.00. 100

6.6 Volume-rendered isosurfaces at a range of FA values show basic 3D structure of white matter in a DTI brain scan. 102

6.7 Volume renderings of half a DT-MRI brain scan, with isovalue FA = 0.4, from a medial and slightly inferior view. 103

6.8 Volume renderings of half a DT-MRI brain scan, with isovalue FA = 0.5, from a lateral and slightly inferior view. 104

6.9 Renderings of anterior corpus callosum and cingulum bundle, showing the effect of different kernel choices 106

7.1 One coronal (left) slice and two sagittal (right) slices used for inspecting tensor gradient components, shown with the RGB colormap (a), and with anatomical regions of interest labeled (b). A: cingulum bundle (green) superior to corpus callosum (red), B: inferior longitudinal fasciculus (green) lateral to internal capsule (purple), C: cerebellar commissures (green) lateral to cortico-spinal tracts (blue), D: anterior commissure (red), E: trans-pontine tracts (red) amidst cortico-spinal tracts (blue), F: cingulum bundle (green and blue) superior to corpus callosum (red). 109

7.2 Tensor gradient magnitude $\|\nabla \mathbf{D}\|$ on the same dataset slices shown in Figure 7.1. 109

7.3 Images of overall magnitudes of shape change (a) and orientation change (b). Arrows point to boundary of ventricle, which is essentially absent in the orientation change image. 110

7.4 Magnitudes of changes in eigenvalue mean (a), variance (b), and skewness (c). The boundary between cingulum bundle and corpus callosum is indicated by arrows in the coronal (right-most) slice of (c), and by the indicated box (including zoom) in the second sagittal (left-most) slice of (c). 111

7.5 Continuous variation between linear, planar, and linear. 111

7.6 Magnitudes of orientation changes measured in DT-MRI slices. The boundary between cingulum bundles and corpus callosum is highlighted in the $|\nabla \phi_3|$ image in (c). 112

7.7 High values of $\sqrt{|\nabla \alpha_3|^2 + |\nabla \phi_3|^2}$ (a), multiplied by c_p (b), lead to a successful indicator of proximity between orthogonally oriented linear anisotropy (c). 113

7.8 Images of RGB colormap (a) and overall gradient magnitude (b) in one coronal (left) and one sagittal (right) slice. 115

7.9 Magnitudes of three shape and three orientation components of $\nabla\mathbf{D}$. 116

7.10 Second demonstration of the two factors in detecting proximal orthogonal tracts: $\sqrt{|\nabla\alpha_3|^2 + |\nabla\phi_3|^2}$ (a) detects the linear-planar-linear transition, while the c_p measure (b) highlights regions of planar anisotropy, resulting in a successful delineation (c) of the cingulum bundle and corpus callosum, as well as orthogonal tracts in the brainstem (indicated by arrows). 117

C.1 As α decreases, due to higher anisotropy or a higher setting of γ (Equation 3.6), root-finding by Newton-Raphson becomes more challenging for the function $q(t)$, but not so for $\log(1 + q(t))$. 129

LIST OF TABLES

2.1 Summary of elements of tensor algebra. 7
4.1 Constraints for tensorline propogation. 61

ACKNOWLEDGMENTS

I first met Chris Johnson on Monday, June 24, 1996, when we were both at the University of North Carolina at Chapel Hill for a site visit of the NSF Science and Technology Center in which Cornell and University of Utah were members. I was there to describe the project that was funding my master's thesis research at the time. During a break in the preparations, I gave Chris a demo of a very early version of my software for visualizing scatterplots of scalar volumes, to see the relationship between data value, gradient magnitude, and a second directional derivative (the kinds of histograms that can help one set a volume rendering transfer function). Chris was interested in what he saw. I'd like to think that my demo that day played some part in my acceptance to the PhD program at University of Utah nine months later. I am indebted to Chris for making me part of his tremendous research lab. The quality, scope, and variety of research within the Scientific Computing and Imaging (SCI) Institute is a testament to his vision and talent. With luck, as future professor and mentor I can pass on to my students some fraction of the experience, finesse, and insight that Chris so generously makes available to his students.

Numerous people have made my years at Utah more enjoyable, memorable, and productive. Besides being a great teacher, Peter Shirley's unique perspective has usefully grounded my understanding of diverse academic and political issues. Dean Brederson's irreverent wit, wide-ranging musical tastes, and mathematical insight had a formative role in my earlier years as a SCI student. Wilson Hsieh has been a valuable friend, and our countless conversations on all matters of music and academia are fondly remembered. I am immensely pleased to have worked with Joe Kniss in research that led to two oft-cited publications and some software of remarkable longevity. Trips to the coffee shop with Milan Ikits led to many enjoyable evenings of research discussions and software hacking. Ever since I first arrived in Utah, David Weinstein has been a great friend and collaborator. Our numerous conversations, brain-storming sessions, and late-night coding marathons are unquestionably the highlights of my graduate student years. Greg Jones has been an unending source of humor and wisdom, and the SCI group is stronger and more focused thanks to him. Over the years, my close friend Cristian Castillo-Davis has been an essential reference point, giving me perspective on my personal and social life, and inspiring me to extend my research interests and abilities into novel domains. Finally, Meghan Jobson has been an essential source of support and love during these last few years of wrapping up my research and dissertation.

None of the research presented in this dissertation would have gotten off the ground were it not for the time, experience, and help of Andrew Alexander, formerly of the Medical Imaging Research Laboratory at the University of Utah, and currently with the W. M. Keck Laboratory for Functional Brain Imaging and Behavior at the University of Wisconsin-Madison.

Thanks to Matthew Bane, Helen Hu, and Cameron Christensen for their help in preparing some of the material in Chapter 4. More than any other chapter, Chapter 5

was the product of fruitful discussions with those more mathematically sophisticated than myself. Principally responsible are Prof. James Damon (Department of Mathematics, University of North Carolina at Chapel Hill), Prof. Richard Price (Department of Physics, University of Texas at Brownsville), Prof. Jeffrey Weiss (Department of Bioengineering, University of Utah), Prof. Domingo Toledo (Department of Mathematics, University of Utah), and P. Thomas Fletcher (University of North Carolina at Chapel Hill).

The soundtrack for this dissertation was provided primarily by `www.somafm.com`. Support independent internet radio stations (by giving them money) and the independent musicians they broadcast (by buying their albums). This dissertation was fueled in part by Sage's Cafe and their amazing carrot butter. Early stages of the work were fueled in part by the Park Ivy Cafe, now closed, and their amazing blueberry-lime bundt cake.

This research is ultimately due to my parents, Peter and Marcia Kindlmann. I slowly realize the value of the gifts I received from them in my upbringing. A stimulating environment at home rewarded perception and curiosity. Their day-to-day example inspired my work ethic, and taught me that the most exacting standards for one's work should come from oneself. Most importantly, their loving and stable relationship infused the home I grew up in with an expansive peace and tranquility, to which I can return in moments of thoughtful concentration. Any novel ideas in my work arise from the space of quiet and careful reflection that they created within and around me.

Funding for this work was provided by the University of Utah Research Foundation PID 2107127, and NIH/NCRR grants P20 HL68566 and P41 RR12553.

CHAPTER 1

INTRODUCTION

1.1 Motivation

Scientific visualization combines the image synthesis methods of computer graphics with mathematical models of the physical world to create a visual framework for discovering, understanding, and solving scientific problems [159]. Magnetic resonance imaging (MRI) has become an important source of medical data due to its ability to noninvasively image organs and tissue. Techniques for interpreting MRI data have grown in complexity as MRI itself has become a more sophisticated and versatile indicator of biological structure and function. Within the last 10 years, *diffusion tensor* MRI has matured as way to measure the microstructural organization of living tissue, such as the white matter in the nervous system [15, 16]. Because each point in a three-dimensional diffusion tensor MRI scan is a symmetric 3×3 matrix, the data can not be interpreted simply by inspecting grayscale images of scans. This dissertation presents new scientific visualization methods for understanding and analyzing the salient structure within diffusion tensor datasets.

Diffusion is the transport of one material through another by the random microscopic motion of molecules due to thermal energy. Diffusion tensor MRI (DT-MRI) characterizes the manner in which the diffusion of water molecules is directionally constrained by the fine structure of the medium in which the molecules are moving. DT-MRI is thereby an indirect indicator of the fine-scale physical organization of tissue. The proper functioning of organs in the body depends on the integrity of their fine-scale organization. Different parts of the brain, for example, communicate through organized bundles of axons, termed *white matter* for the color of the fatty myelin covering around the axon which aids signal conduction [73]. The coherent organization of axons in white matter permits water to diffuse faster along the axon path than across it, which allows DT-MRI to assess the directional structure of the white matter in the brain [24]. In light of this unique measurement capability, there has been a recent surge of research in applying DT-MRI to problems in neuroanatomy [98, 75, 40], surgical planning [179, 81], as well as the detection and/or characterization of conditions such as stroke [173, 74] and neurodegenerative disease [87, 161].

To provide an initial feel for at what diffusion tensor data looks like, Figure 1.1 shows one slice through a DT-MRI scan, visualized with a matrix of images, each of which displays a single component of the matrix representation of the diffusion tensor. As noted by Laidlaw, this is a direct way of inspecting the tensor values, but it provides little insight into the underlying structure [110]. Two of the most popular methods of visualizing diffusion tensors, shown in Figure 1.2, are improvements over Figure 1.1, but they are still slice-based. Figure 1.2(a) shows in grayscale the level of *anisotropy*, or the extent to which diffusion is constrained in some directions more than others. The white matter deep within the brain is brighter than the gray matter at the surface. Figure 1.2(b) shows, with a simple spherical colormap [151, 142], the direction along which

diffusion is fastest, which is generally the direction of the white matter pathway [24]. The precise definition for the quantities shown in these images is given in Chapter 2. With these kinds of visualizations, the basic patterns in the tensor field can be seen, but the overall three-dimensional structure is not shown, nor are the six degrees of freedom in the individual tensor values. In both images, the tensor information was visualized by first reducing it to a simpler form, either a scalar or a vector.

The challenges of using scientific visualization to understand diffusion tensor fields are to decide *which* attributes of the tensor value should be displayed, *how* those tensor attributes will be visually abstracted into a comprehensible form, and *where* within the field the data should be displayed, versus being elided in the interests of clarity. These general problems are not unique to tensor visualization, but the intricate and complex structure of the white matter pathways within the central nervous system, and the multivariate nature of tensor values, demand that the visualization methods be especially judicious in their use of form, color, and shading to create a legible and informative image. The same methods can also characterize biological structure in other tissues, including cardiac muscle, as done in Section 3.1.5.

1.2 Overview

This dissertation presents a combination of methods that approach the task of diffusion tensor visualization and analysis in different and complementary ways.

Chapter 2 provides three kinds of background information: the mathematics of tensor values and their operations (Section 2.1), the basic physics of diffusion, MRI, and diffusion imaging (Section 2.2), and most importantly, a thorough description of tensor *shape* (Section 2.3). This section is important because it ties together the mathematical description of tensor shape with the biological and anatomical significance of shape in DT-MRI scans, and presents a two simple visual abstractions for conveying tensor shape.

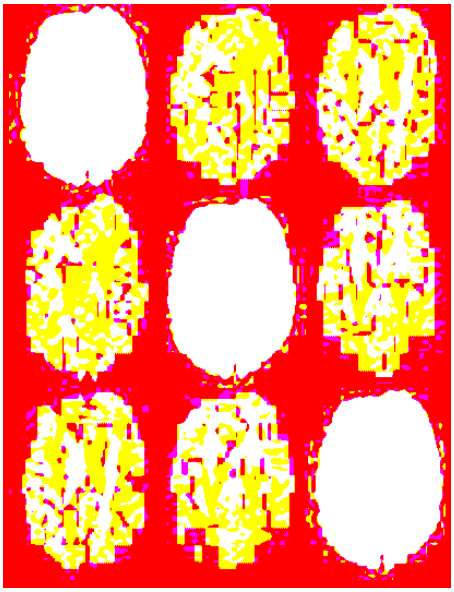
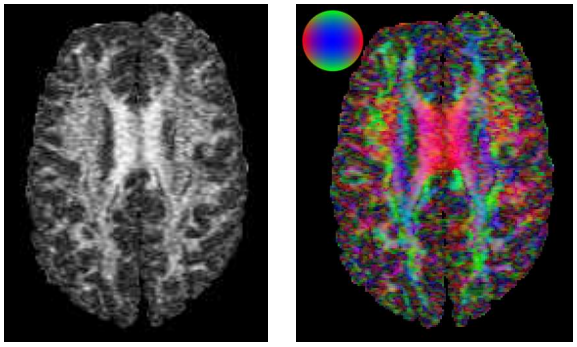


Figure 1.1. Diffusion tensor components



(a) Anisotropy (b) Directional colormap

Figure 1.2. Slice-based tensor visualizations

Chapter 3 gives two methods for conveying all tensor attributes in a restricted set of discrete locations, either by superquadric glyphs (Section 3.1), or by an adaptation of reaction-diffusion systems to create a texture of glyph-like spots (Section 3.2). The underlying form of the superquadric glyph smoothly changes between a sphere, a cylinder, and a cube, conveying the tensor shape and orientation more vividly than is possible with standard ellipsoidal glyphs. The irregular but dense placement of the reaction-diffusion tensor spots allows the underlying structure of the field to be seen more clearly than with regularly spaced glyphs.

Chapter 4 describes a *fiber tractography* method that models the pathways of axons through the white matter, by integrating the directional information of the tensor, similar to a streamline for vector visualization. Propagated from a small number of locations manually chosen by the user, tensorlines visualize the directional similarity along a continuous path of tensors that indicate end-to-end connectivity. Tensorlines are more robust than previous tractography methods because, by the technique of *deflection*, each point along the integration is based on more of the tensor information (both shape and orientation) than simply the single direction of largest diffusivity.

Chapter 5 goes deeper into tensor mathematics, describing a framework for characterizing not tensor shape or orientation, but *changes* in tensor shape and orientation. This chapter is not about visualization per se, but the framework provides a mathematical vocabulary both for enriching visualization of tensor fields, as well as enabling a more nuanced approach to tensor image processing. The approach of this chapter is based on evaluating gradients of tensor invariants, and the tangents to tensor rotations, both of which "live" in the space of tensor values, rather than the spatial domain of a tensor field.

Chapter 6 uses volume rendering to visualize tensor fields, in a manner conceptually opposite that of Chapter 3. Rather than using glyphs to show all the tensor information at a discrete and restricted set of locations, volume rendering seeks to show a restricted set of tensor attributes everywhere in the continuous tensor field. This includes, for example, the continuous structures and surfaces defined by an isosurface of the anisotropy level. Section 6.4 presents analytical shading of anisotropy isosurfaces, based on the gradient methods of Chapter 5, as a way to better communicate feature shape.

Chapter 7 describes preliminary work in applying the mathematics of Chapter 5 to the task of feature-preserving image processing of tensor fields. Feature boundaries are any

location at which particular tensor attributes change suddenly. The goal is to use local measurements of tensors and tensor derivatives to help the extraction of global features and structures, which can then be visualized by volume rendering, for example.

1.3 Contributions

The major contributions of this dissertation may be summarized as:

- A triangular barycentric space of tensor shape (introduced in Section 2.3.4, and used throughout the dissertation) [102, 103, 2]. This is the first abstract visual metaphor for describing and reasoning about the range of diffusion tensor shape, and the characteristics of different shape measures.

- A tensor glyph formed from superquadric surfaces (Section 3.1) [100, 61, 104]. This is the first tensor glyph that continuously changes its underlying geometry, and the first application of superquadrics to tensor visualization.

- Using reaction-diffusion textures as a visualization device (Section 3.2) [103]. Previously used only for creating organic-appearing texture maps on surface models, this work demonstrates that with some care, reaction-diffusion textures can emerge as a closely-packed set of tensor glyphs.

- The use of "deflection" (tensor multiplication) as an ingredient in fiber tractography (Chapter 4) [190]. Previous tractography methods used the direction of fastest diffusion as the only source of directional information.

- The generalization of direct volume rendering to diffusion tensor fields, with shading based on tensor shape and invariant gradients (Chapter 6) [103]. Direct volume rendering previously was applied only to scalar and vector fields.

- The biologically-informed decomposition of tensor changes into subspaces of shape changes and orientation changes (Chapter 5), to improve shading in tensor volume rendering (Section 6.4), and for enabling a novel indicator for proximity of orthogonally oriented fiber tracts (Chapter 7).

The possible applications of diffusion tensor imaging are growing, and visualization of tensor data is an active area of recent research (see surveys in [76]). This dissertation does not describe a general solution or an integrated system for understanding structure within diffusion tensor fields. The chapters contribute a number of distinct methods which broaden the possibilities of tensor visualization and analysis. Experience has shown that the methods do tend to complement each other, such as the local visualization created by the superquadric glyphs, versus the larger-scale forms shown in volume rendering. The methods' visual effectiveness is derived from how the elements of computer graphics are adapted and transformed, as with the deformation of the reaction-diffusion texture into a dense field of glyphs, or the use of smoothing shading in volume rendering.

Determining the ultimate success of the contributions may be done in different ways, as summarized at the end of major sections or chapters: Section 2.3, Section 3.1.6, Section 3.2.4, Section 4.4, Section 6.5, Section 5.9, and Section 7.3.

1.4 Data and Methods

The methods presented here were developed and debugged on different real and synthetic datasets. The real DT-MRI dataset used most often, and which (with the exception of Section 7.2) is the only scan use for the figures in this dissertation, is a particularly high resolution scan of the author, acquired in 1999 with about three hours of scanning. The scanning was done by Dr. Andrew Alexander at the University of Utah Hospital, on their 1.5 Telsa GE Signa Scanner. The original image resolution was 2.031 mm $\times$ 1.016 mm in-plane with a 260 mm $\times$ 260 mm field-of-view, up-sampled by "zero-fill" interpolation (essentially a sinc() filter) to 1.016 mm $\times$ 1.016 mm. Slice thickness was 2.0 mm, but two sets of slices were acquired, 1.0 mm apart, so that with interleaving the Z resolution is nominally 1.0 mm. The number of DWI averages was seven, using the six diffusion-sensitizing gradient directions originally published by Basser [15], with TE = 78.4 ms, TR = 5 s. The diffusion weighting factor b was about 995 s/mm^2.

CHAPTER 2

BACKGROUND

The material in this chapter serves as background for the research in the remainder in the dissertation. Section 2.1 gives the mathematical background for this research, including tensor analysis and the use of convolution to reconstruct and measure sampled data. Section 2.2 describes the biological and physical origins of diffusion tensor data. Section 2.3 summarizes existing knowledge about tensor invariants and shape metrics, with the aid of two graphical devices for representing tensor shape: the *eigenvalue wheel* which depicts the solutions to the cubic characteristic polynomial in terms of moments of the tensor eigenvalues, and a triangular barycentric space of shape. The eigenvalue wheel is based on a figure that appears in previous work by Nickalls [136]; the barycentric shape space is a novel presentation of the anisotropy metrics of Westin [196].

2.1 Mathematical Background

This section reviews the mathematics on which this dissertation depends, concentrating on tensor algebra. Certain mathematics, mostly linear algebra, are assumed:

- Equivalence relations and equivalence classes
- Vector spaces and inner product spaces, cross products
- Bases for vector spaces
- Representation of linear transforms by matrices
- Representation of rotations by orthogonal matrices

Linear algebra textbooks [177, 84] cover these concepts in depth.

The basics of tensor algebra, tensor operations, and matrix representations are given in Sections 2.1.1, 2.1.2, and 2.1.3. This dissertation is concerned only with symmetric tensors, and Section 2.1.4 describes the simplifying consequences of tensor symmetry in terms of eigenvalues and eigenvectors, which underly the ideas of tensor shape and orientation. More thorough explanations of these topics can be found in textbooks [36, 72, 49, 86].

Section 2.1.5 is important because it defines formally how tensors can be considered as elements of a vector space. Although it may seem confusing to say that a tensor *is* a vector, the compelling motivation for this is that established geometric intuitions about vectors, coordinates bases, and the projections of vectors onto each other, can then be applied to tensors. This intuition guides the mathematical design of the tensor image processing described in Chapter 5.

2.1.1 Tensor Algebra Basics

A second-order *tensor* can be thought of as a linear transformation between vector spaces. In this work, $\mathbb{R}^3$, the vector space modeling the three-dimensional physical world, will be both the domain and the range of the linear transformation. Unless otherwise noted, a tensor in this work will always be second-order, which means that the tensor maps vectors to vectors (as opposed to, say, a fourth-order tensor that maps second-order tensors to second-order tensors). A tensor $\mathbf{D}$ takes as input a vector $\mathbf{v}$ and produces, by tensor-vector multiplication, an output vector $\mathbf{w} = \mathbf{D}\mathbf{v}$. Linearity is defined in terms of the usual addition and scaling of vectors: $\mathbf{D}(\mathbf{u} + \mathbf{v}) = \mathbf{D}\mathbf{u} + \mathbf{D}\mathbf{v}$ and $\mathbf{D}(\alpha\mathbf{v}) = \alpha(\mathbf{D}\mathbf{v})$.

The basic elements and operations of tensor algebra are defined in terms of simpler operations like tensor-vector multiplication and vector dot products, without need for any coordinate system. Table 2.1 summarizes the basic ingredients of tensor algebra. In this summary, $\mathbf{C}$ and $\mathbf{D}$ are tensors, $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{w}$ are vectors, and α is a scalar.

A tensor $\mathbf{D}$ is *symmetric* if $\mathbf{D}^\mathbf{t} = \mathbf{D}$ and *antisymmetric*[1] if $\mathbf{D}^\mathbf{t} = -\mathbf{D}$. The set of all symmetric three-dimensional tensors will be notated Sym_3.

Not all tensors are invertible; those that are not invertible are called *singular*.

The following properties of the tensor product of two vectors are easily derived:

$$\begin{array}{rcl} (\mathbf{u}+\mathbf{v}) \otimes \mathbf{w} & = & \mathbf{u} \otimes \mathbf{w} + \mathbf{v} \otimes \mathbf{w} \\ \mathbf{w} \otimes (\mathbf{u}+\mathbf{v}) & = & \mathbf{w} \otimes \mathbf{u} + \mathbf{w} \otimes \mathbf{v} \\ \alpha(\mathbf{u} \otimes \mathbf{v}) & = & (\alpha\mathbf{u}) \otimes \mathbf{v} = \mathbf{u} \otimes (\alpha\mathbf{v}) \end{array} \tag{2.1}$$

The *Kronecker delta* δ will be useful in transforming tensor expressions, and is defined as:

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

Vectors and tensors are independent of any particular coordinate frame, but they have a unique *matrix representation* in a given ordered basis. All bases in this work are *orthonormal* bases: the basis vectors are all unit length and mutually orthogonal. As a consequence, the otherwise important distinction between *covariant* and *contravariant* indices of the tensor may be ignored. Tensors that are represented only in orthonormal

[1] The term *skew-symmetric* is also common; it is avoided here due to our use of "skewness" in a different context.

Table 2.1. Summary of elements of tensor algebra.

Tensor	Notation	Defining Property
identity tensor	$\mathbf{I}$	$\mathbf{I}\mathbf{v} = \mathbf{v}$ for all $\mathbf{v}$
zero tensor	$\mathbf{O}$	$\mathbf{O}\mathbf{v} = \mathbf{0}$ (zero vector) for all $\mathbf{v}$
sum, difference of $\mathbf{C}$, $\mathbf{D}$	$\mathbf{C} \pm \mathbf{D}$	$(\mathbf{C} \pm \mathbf{D})\mathbf{v} = \mathbf{C}\mathbf{v} \pm \mathbf{D}\mathbf{v}$ for all $\mathbf{v}$
(dot) product of $\mathbf{C}$, $\mathbf{D}$	$\mathbf{C}\mathbf{D}$	$(\mathbf{C}\mathbf{D})\mathbf{v} = \mathbf{C}(\mathbf{D}\mathbf{v})$ for all $\mathbf{v}$
transpose of $\mathbf{D}$	$\mathbf{D}^\mathbf{t}$	$\mathbf{v} \cdot \mathbf{D}^\mathbf{t}\mathbf{u} = \mathbf{u} \cdot \mathbf{D}\mathbf{v}$ for all $\mathbf{v}$, $\mathbf{u}$
inverse of $\mathbf{D}$	$\mathbf{D}^{-1}$	$\mathbf{D}\mathbf{D}^{-1} = \mathbf{D}^{-1}\mathbf{D} = \mathbf{I}$
tensor product (or *dyad*) of $\mathbf{u}$, $\mathbf{v}$	$\mathbf{u} \otimes \mathbf{v}$	$(\mathbf{u} \otimes \mathbf{v})\mathbf{w} = \mathbf{u}(\mathbf{v} \cdot \mathbf{w})$ for all $\mathbf{w}$

bases are called *Cartesian* tensors. This dissertation is only concerned with Cartesian tensors.

2.1.2 Matrix Representations

The orthonormality condition of a basis $\mathcal{B} = \{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}$ means that $\mathbf{b}_i \cdot \mathbf{b}_j = \delta_{ij}$, which simplifies expressions for components of vectors and tensors. The matrix representation of vector $\mathbf{v}$ in basis $\mathcal{B}$ is:

$$\begin{aligned} [\mathbf{v}]_{\mathcal{B}} &= \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \; ; \; v_i = \mathbf{b}_i \cdot \mathbf{v} \\ \Rightarrow \mathbf{v} &= \sum_{i=1}^{3} v_i \mathbf{b}_i = \sum_{i=1}^{3} (\mathbf{b}_i \cdot \mathbf{v}) \mathbf{b}_i \end{aligned} \tag{2.2}$$

This may be written more concisely with *index notation* (or *Einstein summation notation*), in which the repetition of an index within a term implies summation of the term over the index's range of values:

$$\mathbf{v} = v_i \mathbf{b}_i = (\mathbf{b}_i \cdot \mathbf{v}) \mathbf{b}_i \tag{2.3}$$

The effectiveness of index notation to manipulate and simplify otherwise complicated expressions more than compensates for the initial effort required to understand the notation. Appendix A provides a more detailed explanation of index notation and its usage. Index notation will be used in the remainder of the dissertation.

The matrix representation of tensor $\mathbf{D}$ in basis $\mathcal{B}$ is the matrix $[\mathbf{D}]_{\mathcal{B}}$ for which

$$[\mathbf{D}\mathbf{v}]_{\mathcal{B}} = [\mathbf{D}]_{\mathcal{B}} [\mathbf{v}]_{\mathcal{B}} \tag{2.4}$$

holds for all vectors $\mathbf{v}$. This definition can be expanded with Equation 2.2, and then column i of $[\mathbf{D}]_{\mathcal{B}}$ can be found by setting $\mathbf{v} = \mathbf{b}_i$:

$$\begin{bmatrix} \mathbf{b}_1 \cdot \mathbf{D}\mathbf{v} \\ \mathbf{b}_2 \cdot \mathbf{D}\mathbf{v} \\ \mathbf{b}_3 \cdot \mathbf{D}\mathbf{v} \end{bmatrix} = [\mathbf{D}]_{\mathcal{B}} \begin{bmatrix} \mathbf{b}_1 \cdot \mathbf{v} \\ \mathbf{b}_2 \cdot \mathbf{v} \\ \mathbf{b}_3 \cdot \mathbf{v} \end{bmatrix} \quad \Rightarrow \quad [\mathbf{D}]_{\mathcal{B}} = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{bmatrix}$$
$$D_{ij} = \mathbf{b}_i \cdot \mathbf{D}\mathbf{b}_j \tag{2.5}$$

The identity tensor $\mathbf{I}$ has the same matrix representation, notated I, in every basis:

$$\begin{aligned} ([\mathbf{I}]_{\mathcal{B}})_{ij} &= \mathbf{b}_i \cdot \mathbf{I}\mathbf{b}_j = \mathbf{b}_i \cdot \mathbf{b}_j = \delta_{ij} \\ \Rightarrow [\mathbf{I}] &= I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

A tensor which has the same representation in every basis is called *isotropic*, although the term also applies to a material property which is modeled with an isotropic tensor (such as diffusion within water). All isotropic tensors have the form $\alpha\mathbf{I}$ for some scalar α.

The matrix representation of $\mathbf{u} \otimes \mathbf{v}$ coincides with the *outer product* of the matrix representations of the vectors:

$$\begin{aligned}
([\mathbf{u} \otimes \mathbf{v}]_{\mathcal{B}})_{ij} &= \mathbf{b}_i \cdot (\mathbf{u} \otimes \mathbf{v})\mathbf{b}_j \\
&= \mathbf{b}_i \cdot (\mathbf{u}(\mathbf{v} \cdot \mathbf{b}_j)) \\
&= (\mathbf{u} \cdot \mathbf{b}_i)(\mathbf{v} \cdot \mathbf{b}_j) \\
&= (u_k \mathbf{b}_k \cdot \mathbf{b}_i)(v_l \mathbf{b}_l \cdot \mathbf{b}_j) \\
&= (u_k \delta_{ki})(v_l \delta_{lj}) \\
&= u_i v_j \\
\Rightarrow [\mathbf{u} \otimes \mathbf{v}]_{\mathcal{B}} &= \begin{bmatrix} u_1 v_1 & u_1 v_2 & u_1 v_3 \\ u_2 v_1 & u_2 v_2 & u_2 v_3 \\ u_3 v_1 & u_3 v_2 & u_3 v_3 \end{bmatrix}
\end{aligned}$$

In particular, the outer product of two basis vectors is represented (in the same basis) as a matrix which is all zeroes except for a single entry equal to 1:

$$\begin{aligned}
([\mathbf{b}_i \otimes \mathbf{b}_j]_{\mathcal{B}})_{kl} &= \mathbf{b}_k \cdot (\mathbf{b}_i \otimes \mathbf{b}_j)\mathbf{b}_l \\
&= \mathbf{b}_k \cdot (\mathbf{b}_i(\mathbf{b}_j \cdot \mathbf{b}_l)) \\
&= \mathbf{b}_k \cdot (\mathbf{b}_i \delta_{jl}) \\
&= \delta_{ik} \delta_{jl}
\end{aligned}$$

The ability to represent a tensor with a matrix permits some tensor operations, like multiplication, transposition, and inverse, to be understood, although not defined, in more familiar terms. The following relationships can be derived (for an arbitrary basis) from the definitions given so far:

$$\begin{aligned}
[\mathbf{CD}] &= [\mathbf{C}][\mathbf{D}] \\
[\mathbf{D}^{\mathbf{t}}] &= [\mathbf{D}]^{\mathbf{t}} \\
[\mathbf{D}^{-1}] &= [\mathbf{D}]^{-1}
\end{aligned}$$

This makes clear, for example, that a symmetric tensor is always represented by a symmetric matrix, that an antisymmetric tensor is always represented by an antisymmetric matrix, and that a tensor is invertible if and only if its matrix representation is invertible.

In the DT-MRI literature, it is fairly common to represent (or even identify) the diffusion tensor with the matrix representation in the coordinate frame determined by the MRI scanner. This is called the *laboratory frame*, which we notate $\mathcal{L}$, and in which the subscripts are x, y, and z:

$$[\mathbf{D}]_{\mathcal{L}} = \begin{bmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{xy} & D_{yy} & D_{yz} \\ D_{xz} & D_{yz} & D_{zz} \end{bmatrix}$$

2.1.3 Trace, Determinant, Contraction, Norm

Some functions of tensors, like trace tr() and determinant det(), are actually defined in terms of their standard matrix counterparts:

$$\operatorname{tr}(\mathbf{D}) = \operatorname{tr}([\mathbf{D}]) = D_{11} + D_{22} + D_{33} \tag{2.6}$$

$$\det(\mathbf{D}) = \det([\mathbf{D}]) = \begin{vmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{vmatrix} \tag{2.7}$$

As functions of *tensors*, tr() and det() are well-defined, because the trace and determinant, as functions of *matrices*, are invariant under similarity (change-of-basis) transforms, which is a standard result of linear algebra [84]. Also, a proof of the invariance of trace is given in Appendix A.

With the trace of a tensor, we can define the *double contraction*, notated with "**:**", which maps a pair of tensors to a scalar:

$$\mathbf{C} : \mathbf{D} = \mathrm{tr}(\mathbf{C}\mathbf{D}^{\mathsf{t}}) = C_{ij}D_{ij} \tag{2.8}$$

The index notation expression assumes an orthonormal basis, but any basis leads to the same value, by the invariance of trace. Note that a vector dot product can be expressed as $\mathbf{u} \cdot \mathbf{v} = u_i v_i$: one index is being "contracted" out by summation. The double contraction, in contrast, sums over two matrix indices.

With double contraction, we can define the tensor *norm*, which provides a (scalar) measure of the overall "size" of a tensor:

$$\|\mathbf{D}\| = \sqrt{\mathbf{D} : \mathbf{D}} = \sqrt{D_{ij}D_{ij}} \tag{2.9}$$

The tensor norm equals the Frobenius norm of the matrix representation:

$$\|\mathbf{D}\| = \|[\mathbf{D}]\|_F = \sqrt{D_{xx}^2 + 2D_{xy}^2 + 2D_{xz}^2 + D_{yy}^2 + 2D_{yz}^2 + D_{zz}^2} \tag{2.10}$$

2.1.4 Eigenvalues and Eigenvectors

An *eigenvalue* of a tensor $\mathbf{D}$ is a scalar λ for which there is a nonzero vector $\mathbf{v}$ satisfying

$$\mathbf{D}\mathbf{v} = \lambda\mathbf{v} \quad \Leftrightarrow \quad (\lambda\mathbf{I} - \mathbf{D})\mathbf{v} = \mathbf{0}$$

This means the tensor $\lambda\mathbf{I} - \mathbf{D}$ is singular, as is the matrix $[\lambda\mathbf{I} - \mathbf{D}]$:

$$\det([\lambda\mathbf{I} - \mathbf{D}]) = \det(\lambda I - [\mathbf{D}]) = 0 \tag{2.11}$$

Equation 2.11 defines a cubic polynomial equation, called the *characteristic equation*, of matrix $[\mathbf{D}]$. Section 2.3.1 will examine the characteristic polynomial in more detail. As can be easily shown from the change of basis formula for matrices, the characteristic equation is in fact intrinsic to tensor $\mathbf{D}$, rather than a function of any particular matrix representation [84].

Associated with eigenvalue λ is a set of *eigenvectors* $\mathbf{v}$ for which $\mathbf{D}\mathbf{v} = \lambda\mathbf{v}$. Actually, the set of eigenvectors is a vector space, sometimes called an *eigenspace*, because it is closed under addition and scalar multiplication. As a vector space, we can chose an orthonormal basis for it, although the space will often only be one-dimensional. Without loss of generality, then, we can assume that the eigenvectors used in this work are unit length.

The fact that the only tensors considered in this work are symmetric significantly simplifies their description in terms of eigenvectors and eigenvalues. From linear algebra we know a number of important properties of an arbitrary $n \times n$ symmetric matrix M [177]:

- M has n real eigenvalues and n orthogonal eigenvectors.
- M can be diagonalized as $M = R\Lambda R^{\mathsf{t}}$. R is an orthogonal matrix in which each column is an eigenvector. Λ is a diagonal matrix of eigenvalues.

- The dimension of the eigenspace associated with eigenvalue λ is equal to the multiplicity of λ as a root in the characteristic equation.

We know from these properties that for a given tensor $\mathbf{D}$ there is some orthonormal basis comprised solely of eigenvectors, $\mathcal{E} = \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$, called the *principal basis* (or *principal frame*), specific to $\mathbf{D}$, in which $\mathbf{D}$ is represented as a diagonal matrix of eigenvalues. From Equation 2.5,

$$\begin{aligned} ([\mathbf{D}]_{\mathcal{E}})_{ij} &= \mathbf{e}_i \cdot \mathbf{D}\mathbf{e}_j \\ &= \mathbf{e}_i \cdot \lambda_j \mathbf{e}_j \\ &= \lambda_j \delta_{ij} \\ \Rightarrow [\mathbf{D}]_{\mathcal{E}} &= \Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \end{aligned}$$

To constrain the choices of $\mathcal{E}$, we sometimes require that the eigenvalues be sorted $\lambda_1 \geq \lambda_2 \geq \lambda_3$. It is also helpful to set the directions of $\mathbf{e}_i$ so that $\mathcal{E}$ is right-handed frame: $\mathbf{e}_3 = \mathbf{e}_1 \times \mathbf{e}_2$. Even with these constraints, the fact that eigenvectors are known only up to line orientation ($\mathbf{e}_i$ and $-\mathbf{e}_i$ are both eigenvectors) means that there are eight possibilities for a right-handed basis $\mathcal{E}$ associated with sorted distinct eigenvalues. The set of eight possibilities is generated by 180 degree rotations around the eigenvectors.

We also know from the properties above that the matrix representation of a symmetric tensor $\mathbf{D}$ in the laboratory frame $[\mathbf{D}]_{\mathcal{L}}$ can be diagonalized, and the column vectors of the rotation matrix are the matrix representation of the eigenvectors in the laboratory frame:

$$\begin{aligned} [\mathbf{D}]_{\mathcal{L}} &= R\Lambda R^{\mathsf{t}} \\ R &= [[\mathbf{e}_1]_{\mathcal{L}} \ [\mathbf{e}_2]_{\mathcal{L}} \ [\mathbf{e}_3]_{\mathcal{L}}] \end{aligned} \tag{2.12}$$

Section 2.3.1 will describe in detail how to compute the eigenvalues of a tensor from its invariants, and Section 3.1.1 will describe how the corresponding eigenvectors are computed.

The *spectral decomposition* of a tensor $\mathbf{D}$ is a useful way to express the tensor directly in terms of eigenvalues λ_i and and eigenvectors $\mathbf{e}$:

$$\mathbf{D} = \mathbf{D}\mathbf{I} = \mathbf{D}(\textstyle\sum_i \mathbf{e}_i \otimes \mathbf{e}_i) = \sum_i (\mathbf{D}\mathbf{e}_i) \otimes \mathbf{e}_i = \sum_i \lambda_i (\mathbf{e}_i \otimes \mathbf{e}_i) \tag{2.13}$$

Appendix A includes a proof that $\mathbf{I} = \sum_i \mathbf{e}_i \otimes \mathbf{e}_i$.

2.1.5 Tensors as Vectors

Equation 2.5 showed how to recover the individual components D_{ij} of a matrix representation of $\mathbf{D}$ given some orthonormal basis $\mathcal{B} = \{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}$, by $D_{ij} = \mathbf{b}_i \cdot \mathbf{D}\mathbf{b}_j$. Related to this is an expression for $\mathbf{D}$ as a linear combination of tensor products of pairs

of basis vectors. For any vector $\mathbf{v}$, we express $\mathbf{Dv}$ by twice applying Equation 2.3 (first to $\mathbf{v}$ and then to $\mathbf{Db}_j$) and rearranging:

$$\begin{aligned}
\mathbf{Dv} &= \mathbf{Db}_j(\mathbf{b}_j \cdot \mathbf{v}) \\
&= \mathbf{b}_i(\mathbf{b}_i \cdot \mathbf{Db}_j)(\mathbf{b}_j \cdot \mathbf{v}) \\
&= (\mathbf{b}_i \cdot \mathbf{Db}_j)\mathbf{b}_i(\mathbf{b}_j \cdot \mathbf{v}) \\
&= (\mathbf{b}_i \cdot \mathbf{Db}_j)(\mathbf{b}_i \otimes \mathbf{b}_j)\mathbf{v} \\
&= D_{ij}(\mathbf{b}_i \otimes \mathbf{b}_j)\mathbf{v} \\
\Rightarrow \mathbf{D} &= D_{ij}(\mathbf{b}_i \otimes \mathbf{b}_j) \qquad (2.14)
\end{aligned}$$

Such a linear combination of tensor products is sometimes called a *dyadic*. It is important to recognize that Equations 2.14 and 2.13 define a *tensor* value, while Equation 2.5 ($D_{ij} = \mathbf{b}_i \cdot \mathbf{Db}_j$) describes a *matrix*. A portion of this work will rely on the idea of tensors as being elements of a nine-dimensional vector space. This is hinted at by the expression of $\mathbf{D}$ in Equations 2.14 as a linear combination of the nine different tensor products $\mathbf{b}_i \otimes \mathbf{b}_j$, which seem to serve as basis "vectors." We expand on this idea here.

Let $L(\mathbb{R}^3, \mathbb{R}^3)$ be the set of all linear transforms from $\mathbb{R}^3$ to $\mathbb{R}^3$. We know from linear algebra that this is a vector space, and that it is $3 * 3 = 9$ dimensional [84]. There is a zero transform $\mathbf{O}$ which is the additive identity:

$$\mathbf{Ov} = \mathbf{0}$$

and addition and scalar multiplication of transforms is defined in terms of the addition and scalar multiplication in $\mathbb{R}^3$:

$$\begin{aligned}
(\mathbf{C} + \mathbf{D})\mathbf{v} &= \mathbf{Cv} + \mathbf{Dv} \\
(\alpha\mathbf{D})\mathbf{v} &= \alpha(\mathbf{Dv})
\end{aligned}$$

For any two vectors $\mathbf{u}$ and $\mathbf{v}$ in $\mathbb{R}^3$, $\mathbf{u} \otimes \mathbf{v}$ is an element of $L(\mathbb{R}^3, \mathbb{R}^3)$, since by definition $\mathbf{u} \otimes \mathbf{v}$ maps from $\mathbb{R}^3$ to $\mathbb{R}^3$, and linearity can be verified with

$$\begin{aligned}
(\mathbf{u} \otimes \mathbf{v})(\alpha\mathbf{w} + \mathbf{x}) &= \mathbf{u}(\mathbf{v} \cdot (\alpha\mathbf{w} + \mathbf{x})) \\
&= \mathbf{u}(\alpha\mathbf{v} \cdot \mathbf{w} + \mathbf{v} \cdot \mathbf{x}) \\
&= \alpha\mathbf{u}(\mathbf{v} \cdot \mathbf{w}) + \mathbf{u}(\mathbf{v} \cdot \mathbf{x}) \\
&= \alpha(\mathbf{u} \otimes \mathbf{v})(\mathbf{w}) + (\mathbf{u} \otimes \mathbf{v})(\mathbf{x})
\end{aligned}$$

Given an orthonormal basis $\mathcal{B} = \{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}$ for $\mathbb{R}^3$, we define a set of nine vectors $\boldsymbol{\mathcal{B}}$ in $L(\mathbb{R}^3, \mathbb{R}^3)$ by:

$$\begin{aligned}
\boldsymbol{\mathcal{B}} &= \{\mathbf{B}_{11}, \mathbf{B}_{12}, \mathbf{B}_{13}, \mathbf{B}_{21}, \mathbf{B}_{22}, \mathbf{B}_{23}, \mathbf{B}_{31}, \mathbf{B}_{32}, \mathbf{B}_{33}\} \\
\mathbf{B}_{ij} &= \mathbf{b}_i \otimes \mathbf{b}_j
\end{aligned}$$

With Equation 2.14, any tensor $\mathbf{D}$ can be expressed as a linear combination of elements of $\boldsymbol{\mathcal{B}}$:

$$\mathbf{D} = (\mathbf{b}_i \cdot \mathbf{Db}_j)(\mathbf{b}_i \otimes \mathbf{b}_j) = D_{ij}\mathbf{B}_{ij}$$

This establishes that $\boldsymbol{\mathcal{B}}$ is a basis for $L(\mathbb{R}^3, \mathbb{R}^3)$. Furthermore, one can verify that the double contraction "$\mathbf{:}$" satisfies the properties of being an *inner product* in $L(\mathbb{R}^3, \mathbb{R}^3)$ (it is

scalar-valued, bilinear, symmetric, and positive definite), just like the usual dot product "$\cdot$" in $\mathbb{R}^3$. With respect to this inner product, $\boldsymbol{\mathcal{B}}$ is an *orthonormal* basis:

$$\begin{aligned}
\mathbf{B}_{ij} : \mathbf{B}_{kl} &= \mathrm{tr}(\mathbf{B}_{ij}\mathbf{B}_{kl}^{\mathsf{t}}) \\
&= \mathrm{tr}((\mathbf{b}_i \otimes \mathbf{b}_j)(\mathbf{b}_k \otimes \mathbf{b}_l)^{\mathsf{t}}) \\
&= \mathrm{tr}([\mathbf{b}_i \otimes \mathbf{b}_j]_{\mathcal{B}}[\mathbf{b}_k \otimes \mathbf{b}_l]_{\mathcal{B}}^{\mathsf{t}}) \\
&= ([\mathbf{b}_i \otimes \mathbf{b}_j]_{\mathcal{B}})_{mn}([\mathbf{b}_k \otimes \mathbf{b}_l]_{\mathcal{B}})_{mn} \\
&= \delta_{im}\delta_{jn}\delta_{km}\delta_{ln} \\
&= \delta_{ik}\delta_{jl} \qquad (2.15)
\end{aligned}$$

Having an orthonormal basis and an inner product in the space of tensors will greatly facilitate the analysis methods described in Chapter 5 and Chapter 7. Chapter 5 in particular develops the idea of defining, for each symmetric tensor, a basis for the three-dimensional vector space of shape changes around the tensor, and a separate basis for the three-dimensional vector space of orientation changes.

2.1.6 Third-order tensors

Chapters 5 and 6 will have occasion to use a *third-order* tensor. Just like a second-order tensor can be expressed as a linear combination of tensor products of pairs of basis vectors (Equation 2.14):

$$\mathbf{D} = D_{jk}(\mathbf{b}_j \otimes \mathbf{b}_k)\,,$$

a third order tensor $\boldsymbol{\mathcal{D}}$ is expressed as a linear combination of tensor products of triples of basis vectors [86]:

$$\boldsymbol{\mathcal{A}} = A_{ijk}(\mathbf{b}_i \otimes \mathbf{b}_j \otimes \mathbf{b}_k)$$

The third-order tensor $\mathbf{u} \otimes \mathbf{v} \otimes \mathbf{w}$ is defined by the properties:

$$\begin{aligned}
(\mathbf{u} \otimes \mathbf{v}) \otimes \mathbf{w} &= \mathbf{u} \otimes \mathbf{v} \otimes \mathbf{w} \\
\mathbf{u} \otimes (\mathbf{v} \otimes \mathbf{w}) &= \mathbf{u} \otimes \mathbf{v} \otimes \mathbf{w} \\
(\mathbf{u} \otimes \mathbf{v} \otimes \mathbf{w})\mathbf{x} &= (\mathbf{w} \cdot \mathbf{x})\mathbf{u} \otimes \mathbf{v}
\end{aligned}$$

Double contraction between a third-order tensor and a second-order tensor produces a first-order tensor, otherwise known as a vector. In the third-order tensor, the two indices closest to "**:**" are contracted out (summed over), while the other index remains to identify the basis vector:

$$\boldsymbol{\mathcal{A}} : \mathbf{D} = A_{ijk}D_{jk}\mathbf{b}_i$$

2.1.7 Group Action ψ, Tensor Shape, and Invariants

A *rotation* $\mathbf{R}$ is a tensor that preserves angles between vectors and lengths of vectors, and that has $\det(\mathbf{R}) = 1$. The matrix representation of a rotation in any orthonormal basis is an orthogonal matrix. The set of all possible rotations is notated SO_3, which is a *group* because it has an identity, because rotations are invertible, and because composition of rotations is associative [57]. SO_3 is *not* a vector space, however, because the zero tensor $\mathbf{O}$ is not a rotation, and because the sum of two rotations is not a rotation.

Fundamental to Chapter 5 is the function ψ that takes a rotation and a symmetric tensor to produce a rotated version of the tensor:

$$\begin{aligned} \psi : SO_3 \times Sym_3 &\mapsto Sym_3 \\ \psi(\mathbf{R}, \mathbf{D}) &= \mathbf{R}\mathbf{D}\mathbf{R}^{\mathsf{t}} \end{aligned} \tag{2.16}$$

Note that if $\mathbf{R} \neq \mathbf{I}$, then strictly speaking $\mathbf{D}$ and $\psi(\mathbf{R}, \mathbf{D})$ are distinct tensors. However, since $[\psi(\mathbf{R}, \mathbf{D})] = [\mathbf{R}][\mathbf{D}][\mathbf{R}]^{\mathsf{t}} = R[\mathbf{D}]R^{\mathsf{t}}$, the matrices $[\psi(\mathbf{R}, \mathbf{D})]$ and $[\mathbf{D}]$ (for any fixed coordinate frame) are the same as matrix representations of $\mathbf{D}$ in two different coordinate frames. One can therefore consider ψ as either rotating tensors, or rotating the laboratory frame in which tensors are acquired. ψ is a *group action*: each element of the group SO_3 defines a mapping from Sym_3 to Sym_3. ψ preserves the double contraction between any two tensors $\mathbf{C}$ and $\mathbf{D}$:

$$\begin{aligned} \psi(\mathbf{R}, \mathbf{C}) : \psi(\mathbf{R}, \mathbf{D}) &= \mathrm{tr}(\mathbf{R}\mathbf{C}\mathbf{R}^{\mathsf{t}}(\mathbf{R}\mathbf{D}\mathbf{R}^{\mathsf{t}})^{\mathsf{t}}) \\ &= \mathrm{tr}(\mathbf{R}\mathbf{C}\mathbf{D}^{\mathsf{t}}\mathbf{R}^{\mathsf{t}}) \\ &= \mathrm{tr}(\mathbf{C}\mathbf{D}^{\mathsf{t}}) \\ &= \mathbf{C} : \mathbf{D} \end{aligned}$$

Thus, ψ also preserves the tensor norm. $\psi(\mathbf{R}, \mathbf{D})$ may be written simply as $\psi(\mathbf{D})$ when $\mathbf{R}$ is fixed and assumed from context. The group action $\psi(\mathbf{D})$ is invertible and linear in $\mathbf{D}$.

The *orbit* $SO_3(\mathbf{D})$ of a tensor $\mathbf{D}$ is the set of all images of $\mathbf{D}$ under ψ:

$$SO_3(\mathbf{D}) = \{\mathbf{R}\mathbf{D}\mathbf{R}^{\mathsf{t}} | \mathbf{R} \in SO_3\}$$

The orbits of ψ partition Sym_3 into equivalence classes by similarity:

$$\mathbf{D}_1 \sim \mathbf{D}_0 \Leftrightarrow \exists\, \mathbf{R} : \mathbf{D}_1 = \psi(\mathbf{R}, \mathbf{D}_0)$$

"$\sim$" is an equivalence relation (it is reflexive, symmetric, and transitive) exactly because SO_3 is a group. We use this equivalence to formally define what we mean by a tensor's *shape*: $\mathbf{D}_1$ and $\mathbf{D}_0$ have the same shape if and only if $\mathbf{D}_1 \sim \mathbf{D}_0$. Because the characteristic polynomial of $\mathbf{D}$ and $\psi(\mathbf{R}, \mathbf{D})$ are the same, and because the characteristic polynomial is entirely determined by the eigenvalues, one may also identify the shape of a tensor with the set of its three eigenvalues.

A tensor invariant $J : Sym_3 \mapsto \mathbb{R}$ is a scalar function for which $\mathbf{D}_1 \sim \mathbf{D}_0 \Rightarrow J(\mathbf{D}_1) = J(\mathbf{D}_0)$. Invariants are constant on orbits of ψ: $J(\psi(\mathbf{D})) = J(\mathbf{D})$. Invariants are basic to this work because they can quantify various aspects of tensor shape, and do not change with changes in orientation. Two tensor invariants have already been mentioned so far– the trace and determinant.

2.1.8 Measurement by Convolution

Medical images produced by commercial magnetic resonance imaging (MRI) are generally a stack of two-dimensional images. The goal of the visualization and analysis, however, is to understand the underlying biological structures in the object that has been scanned and sampled. To recreate a continuous signal from the discretely sampled data, and to measure derivatives of the continuous signal, this work relies heavily on

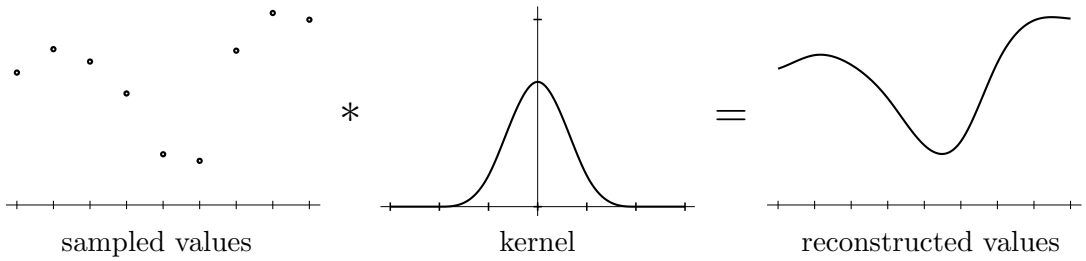


Figure 2.1. Convolving discretely sampled values with a continuous kernel generates a continuous signal. Note that the mathematical definition of convolution involves replacing the discrete sample values with scaled Dirac delta functions in the continuous domain.

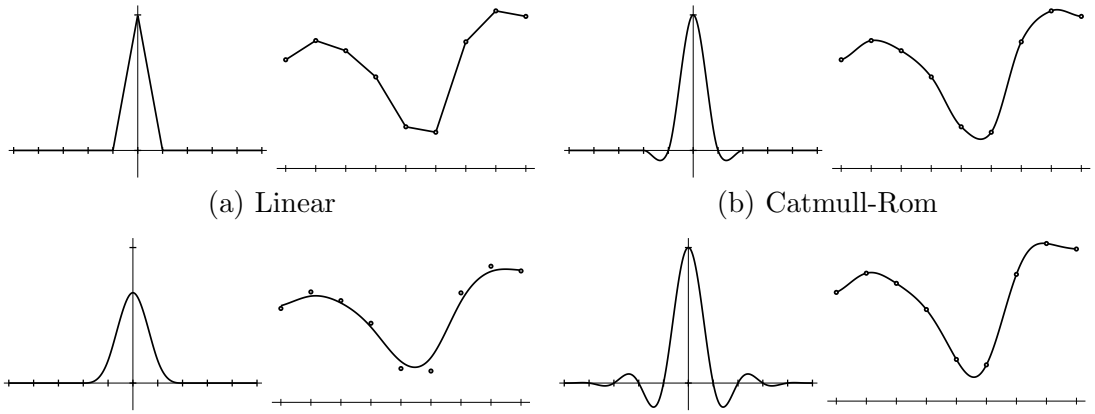
separable convolution with *continuous kernels*. Evaluation of kernel choices for a given measurement task is not the focus of this research, so the description below is at a general and descriptive level.

Figure 2.1 demonstrates the basic operation of convolving discretely sampled data with some continuous kernel to produce a continuous output signal. Mathematically, the sequence of sampled data values $v[i]$ (defined for some range of integral values i) is represented in the continuous domain as $v(x) = \sum_i v[i]\delta(x-i)$, where $\delta(x)$ is the Dirac delta function. The convolution of $v(x)$ with continuous kernel $h(x)$ is defined as:

$$\begin{aligned}(v * h)(x) &= \int_{-\infty}^{+\infty} v(\xi)h(x-\xi)d\xi \\ &= \int_{-\infty}^{+\infty} \sum_i v[i]\delta(\xi-i)h(x-\xi)d\xi \\ &= \sum_i v[i]h(x-i)\end{aligned}$$

In practice, the bounds of the summation will be limited by the *support* of $h(x)$: the range of positions for which $h(x)$ is nonzero. The result is the same as adding together copies of $h(-x)$ located at the integers i, and scaled by $v[i]$.

A large body of image processing research focuses on the task of optimizing the choice of kernel for a given filtering or measurement task [71]. Some of the factors that often have to be evaluated include the computational expense of kernel evaluation, frequency space characteristics, noise sensitivity, and ringing (or overshoot). Research in kernel design has also been a theme within the visualization literature [123, 27, 129, 130, 180], although trilinear interpolation is still the *de facto* norm. Especially important in processing volume data is the size of kernel support, since sampling values within a large (say, $10 \times 10 \times 10$) neighborhood can incur a serious performance penalty, due to the latency associated with accessing values at widely disparate memory addresses [146]. Figure 2.2 shows a variety of kernels reconstructing the same underlying sampled data. The simplicity of linear interpolation (Figure 2.2(a)) is offset by its poor aliasing, while windowed sinc kernels (such has by Hann windowing, Figure 2.2(d)) can be arbitrarily close to ideal in the frequency domain, at the expense of having a large support and expensive computation. Cubic kernels offer an effective and flexible compromise; the "BC" family of splines introduced by Mitchell and Netravali [128] are a two-parameter family of cubic kernels including Catmull-Rom (at (B,C) = (0,0.5); Figure 2.2(b)) and



(a) Linear (b) Catmull-Rom

(c) Uniform Cubic B-spline (d) Hann windowed sinc

Figure 2.2. Effect of kernel choice (left) on reconstruction result (right), for four different kernels. Hann windowed sinc (d) has a support ten samples wide.

the uniform cubic B-spline (at (B,C) = (1,0); Figure 2.2(c)). The B-spline is an example of a noninterpolating kernel which incorporates some smoothing in order to attenuate the response to high-frequency noise.

Ideally, the choice of kernel for reconstructing MRI would be informed by the underlying mathematics of image acquisition. The echo-planar imaging used in many MRI studies collects information for a slice of the dataset in frequency space, directly implying that the sinc() function is the appropriate choice for reconstruction kernel within-plane (X and Y coordinates), though the appropriate kernel choice for between-plane (Z coordinate) has a more complex relationship to the method of slice selection in the pulse sequence [118]. Windowing (as in Figure 2.2(d)) reduces the infinite spatial extent of sinc() to practical bound. However, simple trilinear interpolation is still the *de facto* norm for volume visualization of MRI data. Reconstruction is also an important issue for fiber tractography (Chapter 4), since the path integration relies on tensor values at locations between image grid points. Here too trilinear interpolation is the norm [41, 114, 203], and is the approach used in our work (Chapter 4).

Often the quantity to be measured within an image is not simply the reconstructed value, but a spatial derivative. Fortunately, convolution and differentiation are both linear operators which commute:

$$\begin{aligned} \left.\frac{d(v*h)(x)}{dx}\right|_{x=x_0} &= \left.\frac{d\sum_i v[i]h(x-i)}{dx}\right|_{x=x_0} \\ &= \sum_i v[i]h'(x_0-i) \\ &= (v*h')(x_0)\ . \end{aligned}$$

Because $(v*h')(x)$ is the analytical derivative of $(v*h)(x)$, convolving with the derivative of reconstruction filter provides a means of measuring the first derivative. This is illustrated in Figure 2.3. Second derivatives may be measured by convolving with second derivatives of reconstruction filters. This is illustrated in Figure 2.4.

Although it is true in general that the most accurate way to measure a derivative is by convolving with the derivative of an accurate reconstruction kernel, the findings of Möller

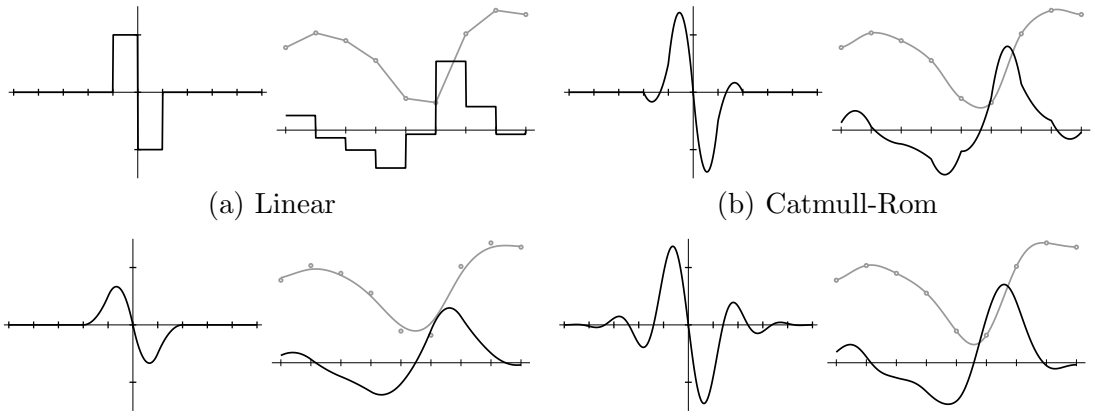
et al. [129, 130] demonstrate that is not a necessary rule. By evaluating polynomial kernels in terms of smoothness (the existence and continuity of derivatives) and accuracy (in terms of frequency spectrum properties), this work demonstrated that a first derivative kernel with a given level of smoothness and accuracy was generally not the derivative of a reconstruction kernel with the same smoothness and accuracy. Thus, determining reconstruction and derivative kernels are largely independent choices, though both are made in terms of practical constraints of computational expense and kernel support.

Noise sensitivity is another consideration when computing derivatives by convolution. Analytic differentiation has a linear frequency response, so noise (typically at higher frequencies) influences differentiation proportionally more than it does in value reconstruction [71]. For this reason, the derivative of a smoothing (noninterpolating) kernel such as the cubic B-spline can be preferable for reconstructing derivatives. Fortuitously, when evaluated at integral sample locations, the first and second derivatives of the B-spline are exactly equivalent to the standard central difference masks for discrete differentiation: $[-0.5\ 0\ 0.5]$ and $[1\ -2\ 1]$, respectively. Also, the cubic B-spline is the unique cubic filter which is second-order accurate (reconstructing quadratic polynomials exactly) [129].

The combination of value reconstruction and differentiation becomes more important on the context of two and three-dimensional images. We follow the approach adopted by others in visualization in the use of *separable convolution*. Separable convolution simply treats each dimension of the image in isolation. A one-dimensional reconstruction kernel $h(x)$ generates a three-dimensional reconstruction kernel $h(x, y, z) = h(x)h(y)h(z)$. The three-dimensional separable convolution sum runs over three image indices of the sampled data $v[i, j, k]$:

$$(v * h)(x, y, z) = \sum_{i,j,k} v[i, j, k]h(x - i)h(y - j)h(z - k) \ .$$

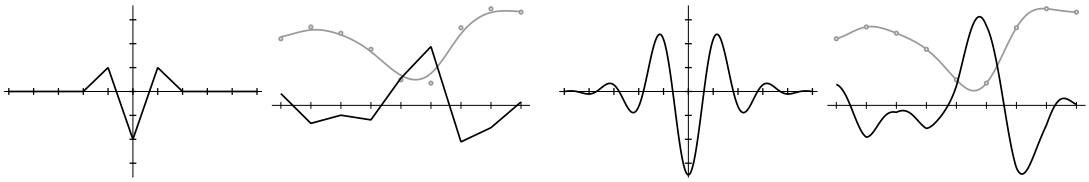
Using a separable kernel vastly simplifies the task of evaluating a partial derivative of the convolution:



(a) Linear (b) Catmull-Rom

(c) Uniform Cubic B-spline (d) Hann windowed sinc

Figure 2.3. Effect of derivative kernel choice (left) on derivative measurement result (right), for the same four kernels used in Figure 2.2. The original data values and value reconstruction results (convolving with the un-differentiated kernel) are shown in gray.



(a) Uniform Cubic B-spline (b) Hann windowed sinc

Figure 2.4. Effect of second derivative kernel choice (left) on second derivative measurement result (right), for two of the kernels used in Figure 2.2.

$$\begin{aligned}
\frac{\partial (v * h)(x,y,z)}{\partial x} &= \sum_{i,j,k} v[i,j,k] h'(x-i) h(y-j) h(z-k) \\
\frac{\partial (v * h)(x,y,z)}{\partial y} &= \sum_{i,j,k} v[i,j,k] h(x-i) h'(y-j) h(z-k) \\
\frac{\partial (v * h)(x,y,z)}{\partial z} &= \sum_{i,j,k} v[i,j,k] h(x-i) h(y-j) h'(z-k) \; .
\end{aligned}$$

In this way, any given first or second partial derivative may be measured by combining the appropriate reconstruction and derivative kernels along the different axes. This is the approach used throughout the dissertation to measure gradients. Chapter 6 on volume rendering in particular relies heavily on convolution with continuous kernels in order to make renderings of continuous field structures and their attributes.

2.2 Diffusion Tensors and their Acquisition

This section describes what diffusion tensors physically represent, and how magnetic resonance imaging can measure diffusion tensors in living tissue. Because this dissertation is focused on the visualization and analysis of diffusion tensor fields, not the physics and mechanics of tensor acquisition, the very simplified description below seeks only to give a flavor of the physics involved. See references [163, 38, 118, 150] for a more detailed explanation.

2.2.1 MRI basics

Diffusion tensor imaging is based on the random Brownian motion of water molecules within biological tissue. Most clinical magnetic resonance imaging (MRI) measures the hydrogen atoms in water molecules, specifically the single proton that is the nucleus of the hydrogen atom. Protons have an intrinsic angular momentum which generates a magnetic dipole moment. The dipole moments are normally randomly oriented. Placed within an external magnetic field, however, the moments become aligned, forming a net *longitudinal magnetization*. Rather than being in a completely fixed orientation relative to the external field, the individual proton dipole moment rapidly rotates, a motion called *precession*, analogous to the wobbling rotation of a spinning top in the presence of gravity. The frequency of the proton precession, called the Larmor frequency, is proportional both

to the external magnetic field strength, and a physical constant called the gyromagnetic ratio.

At the most general level, MRI is based on the principle that an oscillating magnetic field can interact with the precessing protons, causing their net magnetization to tilt away from the external field direction. The oscillating magnetic field comes in the form of a radio-frequency (RF) electromagnetic wave, or *RF pulse*, tuned to the Larmor frequency, so that energy transfers to the proton. The resonance between the RF signal and the proton precession has two effects. First, adding energy to a proton decreases the longitudinal magnetization. Second, from the rotating standpoint of a precessing proton, RF pulses at the Larmor frequency exert a constant transverse force, which gradually rotates the orientation of the proton. This increases phase coherence, creating a rotating *transverse magnetization* orthogonal to the longitudinal magnetization. It is the rotating transverse magnetization which generates a new electromagnetic signal, received by coils in the scanner, which allows MRI to measure material properties.

MRI measurements are primarily based on three intrinsic material characteristics. After the RF pulse is turned off, the exponential relaxation of longitudinal and transverse magnetization back to their original states is governed by two independent time constants, called T1 and T2, respectively. The T1 value measures coupling between the proton and its molecular environment, through which a proton can dissipate energy. The T2 value measures coupling between protons, which tends to disperse the relative phases of precession, thereby decreasing the phase coherence. Underlying both of these quantities is the physical density of protons undergoing precession (and creating a measurable signal), known as proton density or PD. T1 and T2 are in fact both aggregations of more fundamental material properties that may be measured in isolation with magnetic resonance spectroscopy. Differences in T1, T2, and PD values generate image contrast between different tissues, which enables their identification and visualization from MRI scans.

Creating a two-dimensional image from the magnetic resonance signal is possible only with precise control over the frequency and phase of the precession. Magnetic field *gradients* alter the external field strength in a controlled manner, so that the Larmor frequency varies linearly as a function of position. Combined with RF pulses of carefully controlled frequencies and durations, specific slices or regions of the object can be imaged. A specific combination of RF pulses and magnetic gradients is called a *pulse sequence*. A common pulse sequence called "spin echo" can be tuned to generate images which are predominantly weighted by one or another material attribute, such as T1, T2, or PD. Clinical MRI uses a combination of pulse sequences and chemical contrast agents to measure other tissue attributes indirectly through changes in T1 or T2 values.

2.2.2 Modeling Diffusion

Diffusion is the transport of one material through another by the action of random molecular motion due to thermal energy (Brownian motion). The movement of a material by diffusion (called flux) is proportional to the gradient in the concentration of the material. This is Fick's first law [44], which in one dimension is stated as

$$F = -D\frac{\partial C}{\partial x}\,, \tag{2.17}$$

where F is flux and C is concentration. The coefficient D is the *diffusion coefficient*, or *diffusivity*. The negative sign appears because movement is opposite the direction of

increasing concentration. Differentiating Equation 2.17 with respect to position:

$$\frac{\partial F}{\partial x} = -\frac{\partial}{\partial x}\left(D\frac{\partial C}{\partial x}\right) .$$

By conservation of mass, the spatial change in flux (flux in minus flux out) determines the temporal change in concentration. This is Fick's second law:

$$\frac{\partial C}{\partial t} = \frac{\partial}{\partial x}\left(D\frac{\partial C}{\partial x}\right) . \tag{2.18}$$

If diffusivity D is constant with respect to position, then the diffusion equation may be expressed and solved with:

$$\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial x^2} \;\Rightarrow\; C(x,t) = \frac{M}{2\sqrt{\pi D t}}\exp(-\frac{x^2}{4Dt}) . \tag{2.19}$$

Without loss of generality, the normalization $2\sqrt{\pi D}$ was chosen so that for any time t, the integral of $C(x,t)$ over all positions x is M, which can be taken as the total amount of material undergoing diffusion. $C(x,t)$ is a Gaussian with standard deviation $\sqrt{2Dt}$. With $M = 1$, $C(x,t)$ may be viewed as a probability density function. This is closely tied to a fundamental statement of the displacement of a particle undergoing Brownian motion in one dimension, derived from first principles by Einstein [60]:

$$\mathrm{RMS}(x) = \sqrt{2Dt} , \tag{2.20}$$

where $\mathrm{RMS}(x)$ is the root-mean-square of the displacement x of a particle from its initial position after time t. The distance $\sqrt{2Dt}$ is sometimes called the *diffusion length*, and it characterizes the scale over which diffusion occurs, given the diffusivity D and the diffusion time t.

In three dimensions, Fick's first law can take the form

$$\mathbf{j} = -D\nabla c .$$

In this case, vector $\mathbf{j}$ is the direction of the overall motion, or *net flux*, of the material with concentration gradient ∇c. When this equation locally models diffusion (that is, the direction of net flux is parallel to the concentration gradient) the material is said to be *isotropic* (the material has the property of isotropy). Some materials, however, have directional microstructure which permits diffusive motion faster in some directions that others. These materials are *anisotropic* (they have the property of anisotropy). To describe these materials, Fick's first law is generalized to a first-order model in which the direction of net flux is a *linear transform* of the concentration gradient:

$$\mathbf{j} = -\mathbf{D}\nabla c .$$

Rather than a scalar diffusion coefficient, $\mathbf{D}$ is a second-order *diffusion tensor*. Then, Fick's second law in three-dimensional anisotropy may be stated as:

$$\frac{\partial C}{\partial t} = \nabla\cdot(\mathbf{D}\nabla c) \tag{2.21}$$

Note that the diffusion tensor $\mathbf{D}$ is also differentiated, to capture the possibility of material inhomogeneities.

Though it not obvious from the above equations, diffusion tensors are symmetric: $\mathbf{D} = \mathbf{D}^{\mathbf{t}}$. The symmetry of diffusion tensors is a nontrivial result of statistical physics. Onsager's principle of "microscopic reversibility" is often invoked to explain the reciprocity relation associated with tensor symmetry: at the smallest scale, every type of molecular motion occurs just as often as its reverse [137, 138]. Casimir shows how Onsager's principle can be applied to derive the symmetry of conductivity tensors of heat and electricity [39].

2.2.3 Diffusion Tensor MRI

The principle of diffusion imaging is to modify the spin echo imaging sequence in a way that decreases the received signal amplitude according the amount of water diffusion. There are different ways of accomplishing this. The original method is the pulsed gradient spin echo (PGSE) sequence developed by Stejskal and Tanner [176], which uses a momentary magnetic field gradient (the *diffusion-sensitizing* gradient) to create a spatial variation in the phase of proton spin. The result is a linear relationship between phase and position along the gradient direction. To the extent that diffusion causes intermixing of water molecules tagged with different spin phases, the transverse magnetization is decreased, which in turn decreases the brightness of the T2-weighted image. The apparent diffusion coefficient (ADC) along the diffusion-sensitizing gradient is calculated from the change in the T2-weighted image value [28]. Basser demonstrated how ADC measurements over a carefully chosen distribution of gradient directions enable estimation of the effective diffusion tensor [15, 16].

Different diffusion-weighted image (DWI) sequences have different characteristics in terms of speed, sensitivity to motion, and distortion characteristics. For example, the twice-refocused spin echo of Reese et al. sequence reduces global distortions due to eddy currents with a more complex sequence of pulsed magnetic gradients [156]. For the purposes of this simple overview, it suffices to describe the equation governing the original pulsed gradient spin echo sequence of Stejskal and Tanner [176]. The DWI value A associated with diffusion-sensitizing gradient $\mathbf{g}$, in the presence of tensor $\mathbf{D}$, is given by:

$$\begin{aligned} A(\mathbf{g}) &= A_0 \exp(-b\, \mathbf{b} : \mathbf{D}) \\ b &= \gamma^2 \delta^2 (\Delta - \delta/3) \\ \mathbf{b} &= \mathbf{g} \otimes \mathbf{g}\,. \end{aligned} \tag{2.22}$$

The T2-weighted image value without *any* diffusion sensitization is A_0. The proton gyromagnetic ratio γ is known from physics. The PGSE sequence parameters include the delay Δ and duration δ of the diffusion-sensitizing gradient pulses. These variables determine b, the scalar diffusion weighting factor. In the simplest case, the $\mathbf{b}$ matrix is calculated solely from the gradient direction (as above). However, greater accuracy may be attained by taking into account the "cross-terms" induced by interactions between the diffusion sensitization and imaging gradients [16, 14]. In this case, the nominally non-diffusion-weighted image value $A(\mathbf{0})$ is distinct from A_0, so both A_0 and $\mathbf{D}$ must be estimated. In addition to measuring $A(\mathbf{0})$, at least six diffusion-weighted images must be acquired. To ensure that the tensor measurement is unbiased, the direction of gradient directions should be evenly distributed over the sphere [42, 79, 94]. The set of gradient directions $\{\mathbf{g}_i\}$ should satisfy:

$$\sum_i \mathbf{g}_i \otimes \mathbf{g}_i = \mathbf{I}$$

The diffusion tensor **D** may then be estimated by a linear least-squares fit of the logarithm of Equation 2.22, or by a nonlinear least-squares fit of Equation 2.22 itself.

2.2.4 Diffusion in Nervous Tissue

This dissertation focuses on diffusion tensor images acquired by MRI as part of studying the structure in the brain and the central nervous system. Broadly, the brain is composed of two types of nervous tissue: gray matter and white matter [73]. The cell bodies are in the gray matter, which is concentrated on the cortical surface and in nuclei deep with the brain. Within the white matter are the axons connecting the cortex, nuclei, and the rest of the central nervous system. Most axons are covered in a layer of myelin, a fatty substance which increases the efficiency of signal conduction. The internal cytoskeleton of the axon comprises microtubules and neurofilaments, involved in chemical transport and structural support, respectively [24]. Many regions of white matter are characterized by concentrated coherent pathways of axon bundles, called *fiber tracts*. The directional structure within and between axons gives rise to diffusion anisotropy.

It should be noted that the diffusion tensor is a model for diffusion only in those configurations well-characterized by Fick's first law. Specifically, the model assumes *free diffusion*, in which there are no discrete barriers to prevent molecular movement. Anisotropic diffusion can be free diffusion– the net flux is just faster in some directions than others. The heterogeneous microstructure of white matter, however, suggests that there are multiple diffusion barriers, which would lead to *restricted diffusion*. The failure of Fick's laws to completely model restricted diffusion has important consequences: the profiles of diffusive propagation are no longer Gaussian, and the diffusion coefficient will appear to vary as a function of diffusion length (Equation 2.20). A diffusion coefficient measured in the presence of restricted diffusion is called an *apparent diffusion coefficient* (ADC), and its physical significance depends on the time and distance parameters associated with the measurement [28]. In vivo studies of white matter of the brain have not shown variations in ADC consistent with restricted diffusion [29], but this may be explained by the inability to probe at an appropriately wide range of diffusion lengths, due to insufficient magnetic gradient strengths in clinical scanners. Studies of excised nervous tissue in laboratory scanners, on the other hand, have shown evidence for restricted diffusion [166]. More references, and a thorough review of the anatomy, issues, and evidence related to restricted diffusion in nervous tissue is given by Beaulieu [24].

In light of this, the diffusion tensor measured by MRI is not the intrinsic diffusion tensor within any subcellular component, but rather an *effective diffusion tensor* which summarizes and simplifies diffusion behavior over the heterogeneous structures within nervous tissue. The relative simplicity of the diffusion tensor has been recognized as a hindrance to fully characterizing regions where multiple white matter fibers cross or intersect [20, 198]. When Fick's first law is a poor model of diffusive behavior, the molecular displacement equi-probability isosurfaces may diverge from simple ellipsoids, acquiring a more complex multilobed shape [182]. Imposing the tensor model in this case obscures subtleties in the directional dependence of the ADC that may help recover connectivity information in the brain. To an extent as yet unknown, this may be resolved with increases in the spatial resolution of the imaging (see the discussion of partial volume effects in Section 2.3.3).

Another approach is to increase the angular resolution of the ADC measurements, and fit no model at all, or fit a model with more degrees of freedom than a single tensor (e.g., spherical harmonics). This is called *high angular-resolution diffusion* imaging [182, 64,

181, 65], in which every antipodal pair of vertices on a tessellated spherical mesh has an associated ADC value. This provides a much more complete account of the distribution of microstructure orientations, but the task of processing, analyzing, and visualizing this kind of data is formidable.

Although there is currently a great deal of enthusiasm about high angular-resolution diffusion (HARD) imaging, an implicit premise of this dissertation is that diffusion tensor imaging is not moot simply because HARD imaging provides a more complete account of the directional microstructure within each voxel. Even while fully recognizing that the diffusion tensor is an incomplete and somewhat simplistic indicator of microstructure, it still provides a wealth of information that enables a variety of biologically important visualization and analysis tasks. Computing and quantifying neuroanatomic models of gross white matter structure, for example, may not require knowing absolutely everything about how water diffuses through a given region. Furthermore, image processing tasks such as edge detection and feature-preserving smoothing may be significantly enhanced by the information from diffusion tensors, even though yet more information is potentially available from HARD imaging. For context, it is interesting to consider the immense variety of essential scientific and diagnostic tools that are based on traditional magnetic resonance imaging, which primarily measures only three fundamental material characteristics: proton density, and the T1 and T2 relaxation constants.

2.3 Quantifying Tensor Shape

This section reviews previous work and existing knowledge on tensor invariants and their relationship to tensor eigenvalues. We describe tensor invariants in the context of the role they play in solving the cubic characteristic equation of a tensor, to calculate the tensor eigenvalues. Three particular invariants that are statistical properties of the eigenvalues (mean, variance, and skewness) are used in the analytical formulae for the tensor eigenvalues. This relationship between the eigenvalues and their statistics is summarized with a graphical device we term the *eigenvalue wheel*, which provides an intuitive way to reason about the degrees of freedom in tensor shape. We finish the chapter with an alternative way of describing tensor shape, a novel triangular barycentric space which bridges the extremes of spherical, linear, and planar shapes.

The motivation to describe tensor shape in terms of invariants, rather than individual eigenvalues, is based on two related considerations. Primarily, as explained in Section 2.3.3 below, we feel that in the context of tensor image processing, certain invariants, especially the moments of the eigenvalues, have a more biologically significant relationship to tensor shape than do the eigenvalues themselves. The secondary consideration is that we want to avoid unnecessary computational expense. Section 2.3.1 (below) shows how in the standard solution of the cubic characteristic equation, the eigenvalue moments are actually precursors to the computation of the individual eigenvalues, even though one might think that the reverse is true. In some circumstances, the descriptive power of the eigenvalue moments makes the additional computation of the individual eigenvalues less compelling.

2.3.1 Invariants Towards Eigenvalues

The eigenvalues of a tensor $\mathbf{D}$ are computed by solving the cubic polynomial in Equation 2.11:

$$\det(\lambda\mathbf{I} - \mathbf{D}) = 0 \; .$$

The determinant may be evaluated in any coordinate frame, such as the laboratory frame in which the tensor was acquired:

$$\begin{aligned}
\det(\lambda\mathbf{I}-\mathbf{D}) &= \det(\lambda I-[\mathbf{D}]_{\mathcal{L}}) \\
&= \begin{vmatrix} \lambda-D_{xx} & -D_{xy} & -D_{xz} \\ & \lambda-D_{yy} & -D_{yz} \\ (Sym) & & \lambda-D_{zz} \end{vmatrix} \\
&= \lambda^3+(-D_{xx}-D_{yy}-D_{zz})\lambda^2 \\
&\quad +(D_{xx}D_{yy}+D_{xx}D_{zz}+D_{yy}D_{zz}-D_{xy}^2-D_{xz}^2-D_{yz}^2)\lambda \\
&\quad -2D_{xy}D_{xz}D_{yz}-D_{xx}D_{yy}D_{zz}+D_{xz}^2D_{yy}+D_{xx}D_{yz}^2+D_{xy}^2D_{zz}) \\
&= \lambda^3-J_1\lambda^2+J_2\lambda-J_3\ ; \\
J_1 &= D_{xx}+D_{yy}+D_{zz} \\
J_2 &= D_{xx}D_{yy}+D_{xx}D_{zz}+D_{yy}D_{zz}-D_{xy}^2-D_{xz}^2-D_{yz}^2 \\
J_3 &= 2D_{xy}D_{xz}D_{yz}+D_{xx}D_{yy}D_{zz}-D_{xz}^2D_{yy}-D_{xx}D_{yz}^2-D_{xy}^2D_{zz}\ .
\end{aligned} \tag{2.23}$$

From Section 2.1.4 we know that the characteristic polynomial is an invariant, and hence its coefficients J_1, J_2, J_3 are invariant. These are the *principal invariants* (or *fundamental invariants*) of the tensor, which can be expressed in terms of trace and determinant as follows [36, 86]:

$$\begin{aligned}
J_1 &= \mathrm{tr}(\mathbf{D}) \\
J_2 &= \frac{\mathrm{tr}(\mathbf{D})^2-\mathrm{tr}(\mathbf{D}^2)}{2} \\
J_3 &= \det(\mathbf{D})\ .
\end{aligned} \tag{2.24}$$

Equation 2.23 represents how the principal invariants are typically computed in practice, based on the matrix components of the tensor represented in the laboratory frame, as measured by diffusion-weighted MRI (Section 2.2.3). The relationship between the invariants J_i and the eigenvalues λ_i can be seen by computing the characteristic polynomial in the principal frame $\mathcal{E}$ (Section 2.1.4):

$$\begin{aligned}
\det(\lambda I-[\mathbf{D}]_{\mathcal{E}}) &= \begin{vmatrix} \lambda-\lambda_1 & 0 & 0 \\ 0 & \lambda-\lambda_2 & 0 \\ 0 & 0 & \lambda-\lambda_3 \end{vmatrix} \\
&= (\lambda-\lambda_1)(\lambda-\lambda_2)(\lambda-\lambda_3) \\
&= \lambda^3-(\lambda_1+\lambda_2+\lambda_3)\lambda^2+(\lambda_1\lambda_2+\lambda_1\lambda_3+\lambda_2\lambda_3)\lambda-(\lambda_1\lambda_2\lambda_3) \\
&= \lambda^3-J_1\lambda^2+J_2\lambda-J_3 \\
\Rightarrow J_1 &= \lambda_1+\lambda_2+\lambda_3 \\
J_2 &= \lambda_1\lambda_2+\lambda_1\lambda_3+\lambda_2\lambda_3 \\
J_3 &= \lambda_1\lambda_2\lambda_3\ .
\end{aligned} \tag{2.25}$$

Another invariant useful for the determination of eigenvalues is the squared norm:

$$\begin{aligned}
J_4=\|\mathbf{D}\|^2=\mathbf{D}:\mathbf{D} &= J_1^2-2J_2 \\
&= D_{xx}^2+2D_{xy}^2+2D_{xz}^2+D_{yy}^2+2D_{yz}^2+D_{zz}^2 \\
&= \lambda_1^2+\lambda_2^2+\lambda_3^2\ .
\end{aligned}$$

Much of the DTI literature has noted the utility of the J_i invariants as measures of the diffusion tensor shape that do not require diagonalization [16, 12, 22, 187, 78, 208].

Any arithmetic combination of invariants is another invariant. The eigenvalues of a tensor are invariant, and they are computed from some intermediate invariants defined in terms of J_i. Below, we follow the standard formulas for solving a cubic polynomial [33, 192, 155], and adapt the notation of [155]. Equivalent invariant expressions are included for reference.

$$Q = \frac{J_1^2 - 3J_2}{9} = \frac{J_4 - J_2}{9} = \frac{3J_4 - J_1^2}{18} \tag{2.26}$$

$$R = \frac{-9J_1J_2 + 27J_3 + 2J_1^3}{54} = \frac{-5J_1J_2 + 27J_3 + 2J_1J_4}{54} \tag{2.27}$$

$$\Theta = \frac{1}{3}\cos^{-1}\left(\frac{R}{\sqrt{Q^3}}\right) . \tag{2.28}$$

Finally, the analytic formulae for the three eigenvalues are:

$$\begin{aligned} \lambda_1 &= J_1/3 + 2\sqrt{Q}\cos(\Theta) \\ \lambda_2 &= J_1/3 + 2\sqrt{Q}\cos(\Theta - 2\pi/3) \\ \lambda_3 &= J_1/3 + 2\sqrt{Q}\cos(\Theta + 2\pi/3) . \end{aligned} \tag{2.29}$$

2.3.2 Eigenvalue Wheel

The form of Equation 2.29 suggests a geometric analogy, shown in Figure 2.5, which we term the *eigenvalue wheel*. A wheel with three equally placed spokes is centered on the real number line at $J_1/3$. The radius of the wheel is $2\sqrt{Q}$, and Θ measures the orientation of the spokes. The eigenvalues are the projection of the spoke ends onto the horizontal axis. This wheel appears in Figure 2 of previous work by Nickalls [136]. He offers no

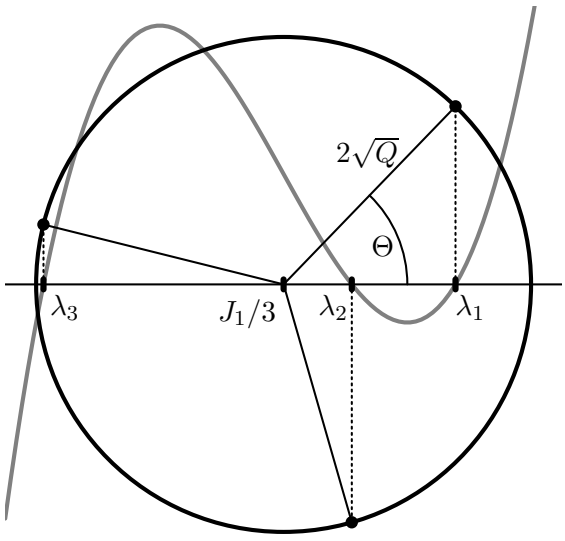


Figure 2.5. Diagram of cubic characteristic polynomial (in gray), its roots, and the invariants required for determining the roots with Equation 2.29.

name for the wheel, and presents it in the context of solving cubics in general (not as eigenvalues of a tensor). It should be noted that this circle does *not* lie on the complex plane: the symmetry of the $\mathbf{D}$ ensures that we have three real eigenvalues, rather than one real and two complex-conjugate eigenvalues.

The central moments μ_1, μ_2, μ_3 of the eigenvalues determine the geometric parameters of the eigenvalue wheel. The central moments are defined as:

$$\begin{aligned}
\mu_1 &= \langle\lambda\rangle = \frac{1}{3}\sum \lambda_i = \frac{\lambda_1+\lambda_2+\lambda_3}{3} && (2.30)\\
&= J_1/3 \\
\mu_2 &= \frac{1}{3}\sum(\lambda_i-\mu_1)^2 = \frac{2(\lambda_1^2+\lambda_2^2+\lambda_3^2-\lambda_1\lambda_2-\lambda_1\lambda_3-\lambda_2\lambda_3)}{9} && (2.31)\\
&= 2Q \\
\mu_3 &= \frac{1}{3}\sum(\lambda_i-\mu_1)^3 && (2.32)\\
&= \frac{2(\lambda_1^3+\lambda_2^3+\lambda_3^3)-3(\lambda_1^2\lambda_2+\lambda_1\lambda_2^2+\lambda_1^2\lambda_3+\lambda_1\lambda_3^2+\lambda_2^2\lambda_3+\lambda_2\lambda_3^2)+12\lambda_1\lambda_2\lambda_3}{27} \\
&= \frac{2J_1^3-9(\lambda_1^2\lambda_2+\lambda_1\lambda_2^2+\lambda_1^2\lambda_3+\lambda_1\lambda_3^2+\lambda_2^2\lambda_3+\lambda_2\lambda_3^2)}{27} \\
&= \frac{2J_1^3-9J_1J_2+27\lambda_1\lambda_2\lambda_3}{27} = \frac{2J_1^3-9J_1J_2+27J_3}{27} \\
&= 2R\,.
\end{aligned}$$

The variance of the eigenvalues is μ_2, and the standard deviation is $\sigma = \sqrt{\mu_2} = \sqrt{2Q}$. The skewness of the eigenvalues, α_3, is a dimensionless quantity defined as [5, 192]:

$$\begin{aligned}
\alpha_3 &= \frac{\mu_3}{\sigma^3} = \frac{R}{\sqrt{2Q^3}} = \frac{\cos(3\Theta)}{\sqrt{2}} && (2.33)\\
\Rightarrow \Theta &= \frac{1}{3}\cos^{-1}(\sqrt{2}\;\alpha_3)\,.
\end{aligned}$$

It should be noted that, somewhat confusingly, "skewness" is also sometimes used to refer to the third central moment, μ_3 [42, 12, 78].

The statistical measures of the eigenvalues are directly bound to the geometric parameters of the eigenvalue wheel: the mean is the wheel center, the variance determines the radius, and the skewness determines the orientation. Each of these degrees of freedom corresponds to an intuitive element of tensor shape: size, anisotropy, and type of anisotropy. Because there can be only three degrees of freedom in any set of 3×3 tensor invariants, the eigenvalue wheel is a complete and succinct means of conveying tensor shape. The geometric intuition that one can modify the wheel's location, radius, and orientation in isolation from one another is grounded in the statistical property that mean, variance, and skewness are not just independent invariants (such as J_1, J_2, J_3), but are *orthogonal* measures of the eigenvalue distribution. That is, viewing μ_1, μ_2, and α_3 as scalar functions over the space of $(\lambda_1,\lambda_2,\lambda_3)$ triples, and letting ∇_λ denote the gradient in this space:

$$\nabla_\lambda\mu_1\cdot\nabla_\lambda\mu_2 = 0\;;\;\nabla_\lambda\mu_1\cdot\nabla_\lambda\alpha_3 = 0\;;\;\nabla_\lambda\mu_2\cdot\nabla_\lambda\alpha_3 = 0\,. \qquad (2.34)$$

A self-contained proof of this fact is given in Appendix B. This Appendix also shows that $\nabla_\lambda\mu_3$ and $\nabla_\lambda\mu_2$ are not orthogonal, motivating the distinction between "skewness"

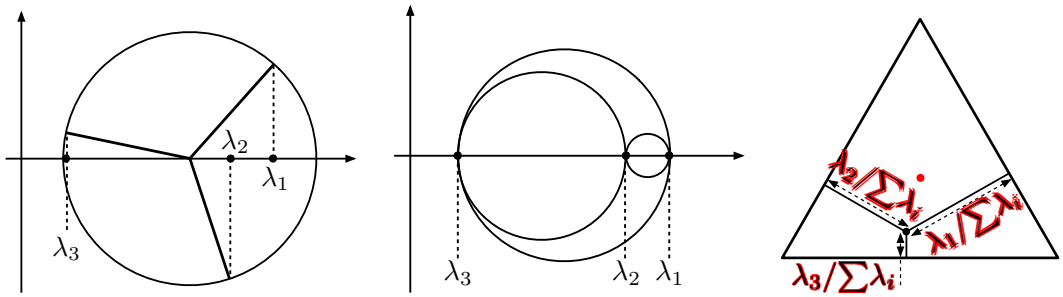
referring to α_3 (our use) versus μ_3. The orthogonality of eigenvalue mean, variance, and skewness was noted by Bahn [5] in the context of developing a cylindrical coordinate system for $(\lambda_1, \lambda_2, \lambda_3)$ space. Previous work in continuum mechanics defined orthogonal measures of hyperelastic strain in terms of the mean, variance, and skewness of the *logarithms* of the eigenvalues of the strain tensor, though without using the statistical terminology adopted here [47].

Equations 2.30 through 2.33 demonstrate an interesting relationship (previously noted in [78]) between the eigenvalues and their moments in the context of solving the characteristic polynomial with Equations 2.26 through 2.29): the statistical moments of the eigenvalues (mean, variance, and skewness) are actually *prerequisites* to the finding the eigenvalues themselves. Also, the eigenvalue wheel illustrates an important property of eigenvalue sorting. From Equations 2.28 and 2.33 we can see that range of Θ and α_3 are bounded: as α_3 varies from $-1/\sqrt{2}$ to $1/\sqrt{2}$, Θ varies from $\pi/3$ to 0, assuming the standard branch choice of defining $\cos^{-1}(x) \in [0, \pi]$. By looking at Figure 2.5, one can see that the limited range in Θ means that the λ_1, λ_2, and λ_3 defined by Equation 2.29 are *already* sorted in descending order [78][2].

Figure 2.6 compares three different tensor shape diagrams. The eigenvalue wheel diagram (Figure 2.6(a)) is superficially similar to the Mohr circle diagram of continuum mechanics, which has been previously applied to DTI [30, 31]. The Mohr circle diagram (Figure 2.6(b)) uses three nested circles to convey the eigenvalues and their pair-wise differences. The eigenvalue wheel, in contrast, uses a single circle to display the eigenvalue mean and variance, with the spoke angle determining the individual eigenvalues. Another tensor shape diagram previously described in the DTI literature uses a trilinear coordinate system (Figure 2.6(c)) in which the set of eigenvalues is indicated by a single point in the triangular space defined by $\lambda_1 + \lambda_2 + \lambda_3 = 1$ and $\lambda_i > 0$ [78].

Figure 2.7 illustrates the three orthogonal degrees of freedom in tensor shape (μ_1, μ_2, and α_3), both with the eigenvalue wheel, and with glyphs. The style of glyphs used here is described in Section 3.1; it suffices to say here that the dimensions of the glyph along its principal axes are the three eigenvalues. Figure 2.7(c) in particular makes clear that the extremal values of skewness are associated with equality between two eigenvalues. This can be confirmed from Equation 2.29: when $\Theta = \pi/3$ (minimal α_3), $\lambda_1 = \lambda_2$,

[2]More precisely, the standard branch choice in $\cos^{-1}$ induces a particular ordering in the λ_i; a different branch choice would lead to a different (fixed) ordering.



(a) Eigenvalue wheel (b) Mohr circles (c) Trilinear coordinates

Figure 2.6. Comparison of tensor shape diagrams.

and when $\Theta = 0$ (maximal α_3), $\lambda_2 = \lambda_3$. The glyph representations also suggest that equality between two eigenvalues generates *axial symmetry* in the tensor. This can be mathematically confirmed by starting with the spectral decomposition $\mathbf{D} = \sum_i \lambda_i(\mathbf{e}_i \otimes \mathbf{e}_i)$, assuming $\lambda_1 = \lambda_2 > \lambda_3$ (left-most tensor in Figure 2.7(c)), and letting $\mathbf{R}$ be rotation by angle θ around $\mathbf{e}_3$. The matrix representation of $\psi(\mathbf{R}, \mathbf{D})$ in the principal basis $\mathcal{E}$ of $\mathbf{D}$ is then:

$$
\begin{aligned}
[\psi(\mathbf{R},\mathbf{D})]_{\mathcal{E}} &= [\mathbf{R}]_{\mathcal{E}}[\mathbf{D}]_{\mathcal{E}}[\mathbf{R}]_{\mathcal{E}}^{\mathsf{t}} \\
&= \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_1 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1\cos(\theta) & -\lambda_1\sin(\theta) & 0 \\ \lambda_1\sin(\theta) & \lambda_1\cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_1 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \\
&= [\mathbf{D}]_{\mathcal{E}} \\
\Rightarrow \psi(\mathbf{R},\mathbf{D}) &= \mathbf{D}\ .
\end{aligned}
$$

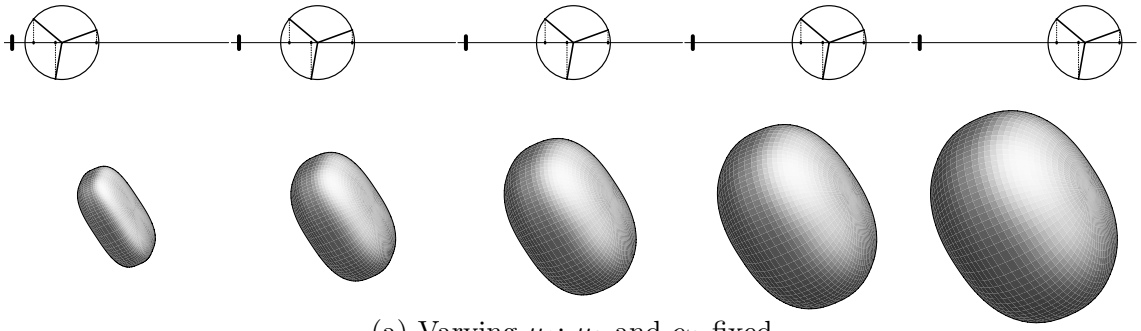
That is, $\mathbf{D}$ is unchanged by rotations around $\mathbf{e}_3$, which is the definition of axial symmetry.

2.3.3 Anatomical Significance of Tensor Shape

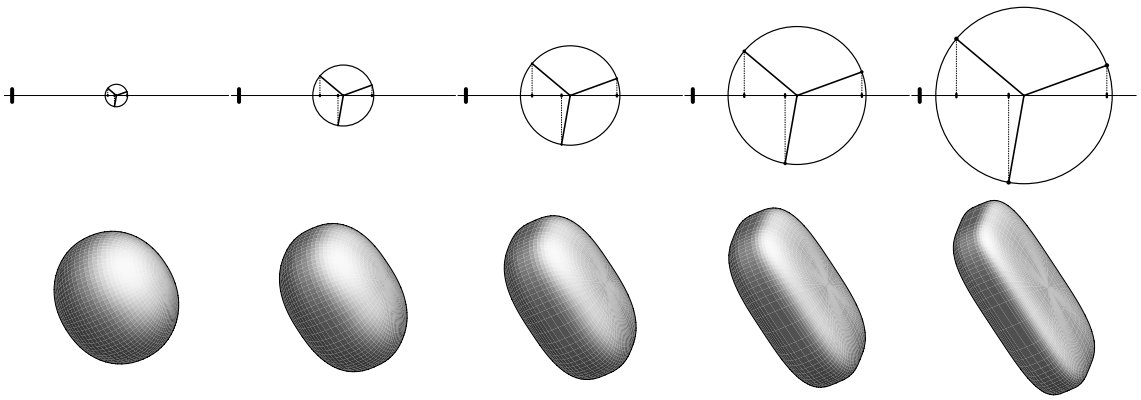
As statistical measures, mean, variance, and skewness are typically employed in contexts where the distribution in question contains more than three values, as is the case here. Also, the use of mean, variance, and skewness is often based on some application-specific assumptions about the form of the distribution being measured. We do not make any such assumptions. Our use of eigenvalue mean, variance, and skewness is grounded first in the geometric intuition illustrated in Figure 2.7, and secondly, in the anatomical and biological significance of these measures.

The eigenvalues of the diffusion tensor are the *apparent diffusion coefficients* (ADCs) along the tensor eigenvectors, as measured in the diffusion-weighted image experiment [15]. The eigenvalue mean μ_1 is the *bulk mean diffusivity* [14], the average of ADC over all possible directions. This quantity readily distinguishes the cerebral spinal fluid (CSF) of the ventricles (high μ_1) from the white and gray matter (low μ_1). In addition, we take as a significant cue the empirical fact that μ_1 is essentially constant across the white and gray matter of healthy brains [153, 186, 187, 6, 14]. In fact, reductions in bulk mean diffusivity have been established as an important indicator of acute ischemic stroke [134, 173].

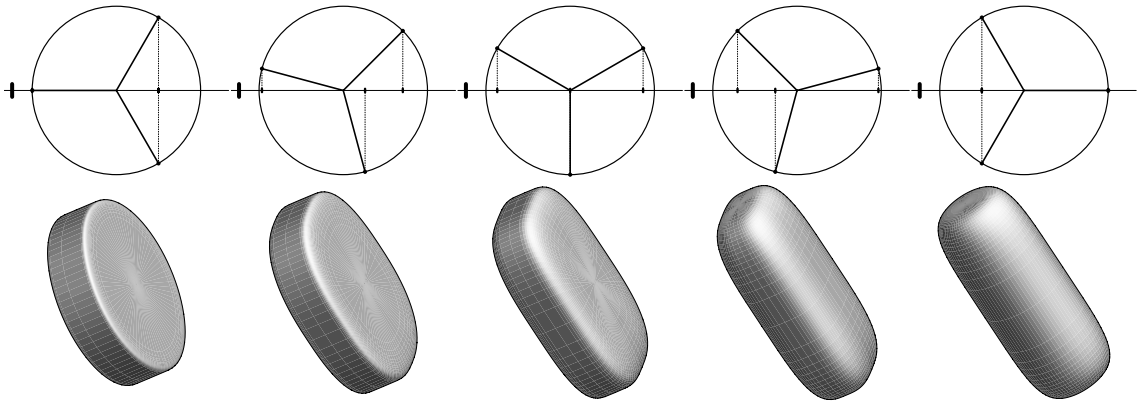
The variance of the eigenvalues μ_2 measures the extent of the directional dependence of the ADC. From a biological standpoint, this indicates directional microstructure, permitting diffusion of water in some directions faster than others. A diffusion tensor is termed *isotropic* when the eigenvalues are all equal, and *anisotropic* when the tensor eigenvalues differ [15]. Two common anisotropy measures in particular are defined in terms of the eigenvalue variance (as well as other invariants). Basser and Pierpaoli defined fractional anisotropy as [21, 152]:



(a) Varying μ_1; μ_2 and α_3 fixed.



(b) Varying μ_2; μ_1 and α_3 fixed.



$\alpha_3 = -1/\sqrt{2}$	$\alpha_3 = -1/2$	$\alpha_3 = 0$	$\alpha_3 = 1/2$	$\alpha_3 = 1/\sqrt{2}$
$\Theta = \pi/3$	$\Theta = \pi/4$	$\Theta = \pi/6$	$\Theta = \pi/12$	$\Theta = 0$

(c) Varying α_3 through its full range by equal steps in Θ; μ_1 and μ_2 fixed.

Figure 2.7. Visualizations of degrees of freedom of tensor shape, with eigenvalue wheel (top) and superquadric glyphs (bottom). The eigenvalue wheel location, radius, and orientation correspond to tensor size, anisotropy, and anisotropy type, respectively.

$$\mathrm{FA} = \sqrt{\frac{3}{2}}\frac{\|\mathbf{D}-\mu_1\mathbf{I}\|}{\|\mathbf{D}\|} = \frac{3}{\sqrt{2}}\sqrt{\frac{\mu_2}{J_4}} = 3\sqrt{\frac{Q}{J_4}} = \sqrt{\frac{J_4 - J_2}{J_4}}\,, \tag{2.35}$$

and relative anisotropy (here scaled by $1/\sqrt{2}$ to lie in [0,1]) as:

$$\mathrm{RA} = \frac{\|\mathbf{D}-\mu_1\mathbf{I}\|}{\sqrt{2}\|\mu_1\mathbf{I}\|} = \frac{\sqrt{\mu_2}}{\sqrt{2}\mu_1} = \frac{3\sqrt{Q}}{J_1}\,. \tag{2.36}$$

As can be determined by Equations 2.35 and 2.36, and as illustrated in Figure 2.7(a), varying μ_1 while holding μ_2 and α_3 fixed will change the tensor anisotropy, at least as measured by FA or RA. Given the ubiquity of FA as an anisotropy measure in the DT-MRI literature, one could argue that this is a problem with the μ_1, μ_2, α_3 orthogonal measures for characterizing shape. An alternative set of three orthogonal shape metrics is J_4 (sum of squares of eigenvalues), FA, and α_3. For the purposes of doing image processing on diffusion tensor fields, we adopt the μ_1, μ_2, α_3 measures, based on the idea that aligning one axis of the shape coordinate system with a variable that is empirically constant (μ_1) allows anisotropic anatomical features to be more easily characterized by changes in the other two variables (μ_2 and α_3).

The relationship between eigenvalue skewness and tensor shape was shown in Figure 2.7(c). This degree of freedom in tensor shape isolates the variation between anisotropic tensors that are "planar" (oblate, large in two axes and small in the other) versus "linear" (prolate, large along one axis, small in the others). The shapes vary from planar to linear as skewness ranges from negative to positive. This kind of shape variation is not measured by FA or RA, and from Equation 2.34 one can show that skewness is *orthogonal* to FA and RA. The problematic aspect of skewness is evident from Equation 2.33: it is most meaningful when variance μ_2 is high, and it is in fact numerically undefined when variance is zero. Thus, it makes sense to describe an anisotropic tensor as varying in skewness (being more linear or more planar), while an isotropic tensor varies only in overall size, not skewness. By analogy, it makes sense to describe the *hue* of a color that is clearly distinct from gray, while grays may vary only in intensity.

Another problematic aspect of skewness is that it is apt to be more sensitive to measurement noise than μ_1 or μ_2, given its involvement of higher powers of differences among eigenvalues, and given that the individual eigenvalues are significantly biased by noise, as noted in the DTI literature [152, 23, 172, 14]. On the other hand, because skewness can be formulated entirely in terms of the principal invariants (Equation 2.33), and is actually a prerequisite to finding the individual eigenvalues (Equations 2.28 and 2.29), its noise sensitivity may have been previously overstated. A careful analysis of the effect of measurement noise on α_3 is needed, but beyond the scope of this work. In any case, isolating this degree of freedom in tensor shape will allow subsequent analysis to selectively respect or disregard its changes according to any prior knowledge about its significance in the presence of measurement noise.

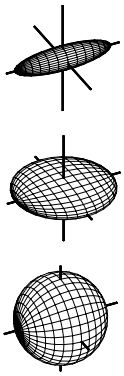
The distinction between linear and planar anisotropy matters in two related contexts. The phenomenon of *partial voluming* is a basic characteristic of discretely sampled medical images, in which the sample value records a measurement over some spatial extent related to the spacing between samples. When two (or more) adjacent distinct materials occupy the region represented by a single sample, the sample value will blend characteristics of the constituent materials. Previous analysis of partial voluming in DTI demonstrated a bias towards planar anisotropy caused by measurement mixing of adjacent regions of linear anisotropy along orthogonal orientations [4, 3]. Locations in the brain characterized by

this configuration of white matter fibers include the right-left trans-pontine tracts ventral to the inferior-superior corticospinal tracts in the brainstem, and the right-left tracts of the corpus callosum inferior to the cingulum bundles (anterior-posterior).

Planar anisotropy can also arise in more complex configurations, rather than mere adjacency of orthogonal fiber orientations. Previous work in visualizing regions of significant planar anisotropy characterized locations where populations of differently-oriented fibers apparently mix at a fine scale, far below that of the image resolution [198]. Locations with this configuration include the medial-lateral fanning within the otherwise anterior-posterior tract direction of the superior longitudinal fascicle and the intersection of the medial-lateral tracts of the corpus callosum with the inferior-superior tracts of the corona radiata (as confirmed by high-angular resolution DWI [182]). More generally, Tuch has demonstrated that in the brain, the residual error of fitting a tensor model to high-angular resolution DWI increases with planarity of the tensor [181]. Because eigenvalue skewness encapsulates the distinction between linear and planar anisotropy, it is appropriate to isolate this variable for the purposes of analyzing and processing diffusion tensor images.

2.3.4 Barycentric Shape Metrics

The anisotropy metrics described above are all directly expressible in terms of the principal invariants and the eigenvalue moments, functions that enjoy the property of being symmetric with respect to permutations of the eigenvalues. A different set of anisotropy metrics was defined by Westin et al. [196] by explicit reference to the *sorted* eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$. The certainty ($c$) with which a tensor can be said to be of a linear, planar, or spherical configuration is parameterized with c_l, c_p, and c_s:



$$\begin{aligned} c_l &= \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3} \\ c_p &= \frac{2(\lambda_2 - \lambda_3)}{\lambda_1 + \lambda_2 + \lambda_3} \\ c_s &= \frac{3\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \end{aligned} \tag{2.37}$$

The metrics above are normalized by the trace, but alternative normalizations have also be described, such as dividing by the maximum eigenvalue [194], or the L2 norm of the eigenvalues [195]. In all cases, by design the metrics sum to unity, and thus parameterize a *barycentric* space of tensor shape. The three barycentric coordinates of a point within an equilateral triangle give the distances between the point and sides of a triangle [43]. The barycentric shape space defined by Equation 2.38 is visualized in Figure 2.8. At each corner of the triangular space, one of the metrics is equal to unity, and the other two are zero. The intermediate values of (c_l, c_p, c_s) within the interior of the triangle represent the smooth continuum of shapes between the three extremes. The linear and planar metrics c_l and c_p both generally increase with higher anisotropy, and $c_a = c_l + c_p$ has been proposed as an alternative to FA or RA as an overall anisotropy metric. The intuitive

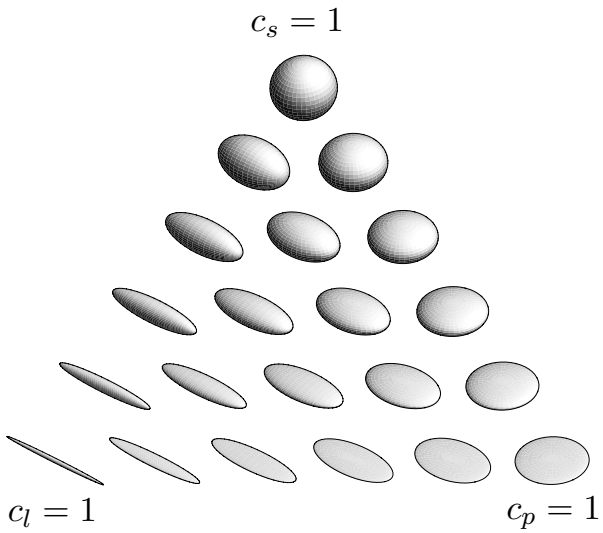
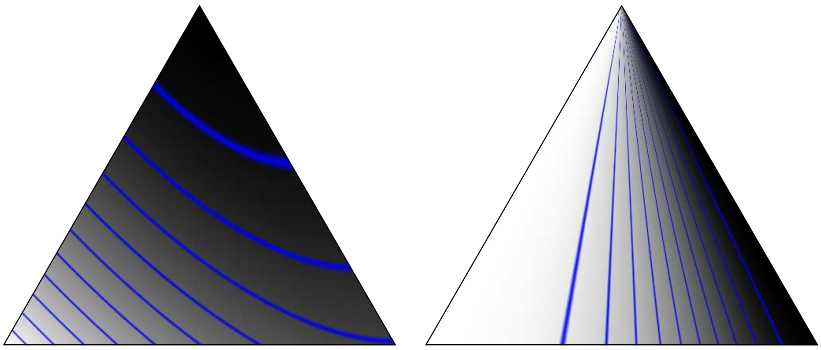


Figure 2.8. Barycentric space of tensor shapes.

geometric interpretation of the barycentric shape space motivates their repeated use in this dissertation, such as in the formulation of superquadric tensor glyphs (Section 3.1), tensorlines (Chapter 4), and volume rendering (Chapter 6). One particularly important role of c_l is to provide some measure of how meaningful the orientation of the principal eigenvector $\mathbf{e}_1$ is.

It is useful to connect the barycentric shape space back to the other main descriptors of tensor shape, the eigenvalue moments. As defined above, the eigenvalue mean μ_1 is constant within the barycentric triangle. Plots of eigenvalue variance μ_2 and skewness α_3 are shown Figure 2.9. A more detailed comparison of different anisotropy measures, in terms of the barycentric shape space, is given by Alexander et al. [2].



(a) μ_2 (b) α_3

Figure 2.9. Plots of eigenvalue variance μ_2 and skewness α_3 in barycentric shape space.

2.3.5 Discussion

This chapter presented no research per se; however the essential concept of tensor shape was explained in detail: from the formulaic standpoint of calculating eigenvalues, from the statistical standpoint of the moments of the eigenvalues, and from two different geometric standpoints (the eigenvalue wheel, and the barycentric shape space). The geometric degrees of freedom in the eigenvalue wheel will be differentiated in the space of tensor values in Chapter 5 to characterize changes in tensor shape.

Although the eigenvalue wheel is based on a figure previously published by Nickalls [136], this is apparently its first use in the context of describing tensor shape. The barycentric shape space is a contribution of this chapter, first appearing in [102]. The success of the barycentric shape space is confirm by its utility in engineering superquadric glyphs (Section 3.1), assessing the distribution of anisotropy shapes throughout a dataset (Figure 4.1), and as the domain of transfer functions for volume rendering (Section 6.2). Barycentric histograms have also been adopted by Zhang et al. as an aid to voxel classification [201].

CHAPTER 3

GLYPHS AND TEXTURES

Tensor field visualization is a challenging task due in part to the multivariate nature of individual tensor samples. This chapter presents methods for visualizating of individual tensor values at discrete locations, using superquadric glyphs and reaction-diffusion textures. Experience has shown that the ability to scrutinize the individual tensor samples is an important first step in exploring and understanding a dataset and the structures within it. Also, low-level visualization is an important tool in debugging errors of data acquisition or representation, such as a disagreement between the handed-ness of the image raster frame and the tensor coordinate frame. Accidentally reversing the order of the slices in the dataset volume is one common example of this.

The principal intent in these methods is to permit local inspection of field properties, rather than extraction or representation of large-scale anatomical structures. However, judicious combination of glyphs and textures can indicate the presence of larger patterns and structures, which may be subsequently explored with fiber tractography (Chapter 4) or volume rendering (Chapter 6).

3.1 Superquadric Tensor Glyphs

Tensor glyphs visualize individual tensor values by mapping the tensor eigenvectors and eigenvalues to the orientation and shape of a base geometric primitive. The standard glyph geometries for tensor visualization, cuboids and ellipsoids, are deficient in different ways. Cuboids can display misleading orientation for tensors with underlying rotational symmetry. Ellipsoids differing in shape can be confused, from certain viewpoints, because of similarities in profile and shading. This section addresses the problems of asymmetry and ambiguity with a new tunable continuum of glyphs based on superquadric surfaces. Superquadric tensor glyphs enjoy the necessary symmetry properties of ellipsoids, while also imitating cuboids and cylinders to better convey shape and orientation, where appropriate.

Note: With the exception of Appendix C, this material was previously published in [100]. The application to cardiac DT-MRI (Section 3.1.5 was previously published as [61]. The same glyph method has also been applied to anatomical covariance tensors derived from deformable registration of cortical surfaces [104].

3.1.1 Eigenvector Calculation

Because the tensor *eigenvectors* will be needed for glyph visualization and for later chapters, we include here a brief overview of the process used to compute the eigenvectors. This computation necessarily involves a particular coordinate frame in which to represent the tensor as a matrix; we use the laboratory frame. From Equation 2.29 we know the

multiplicity of the eigenvalue as a root of the characteristic polynomial. If the multiplicity of eigenvalue λ is one, then the matrix

$$[\mathbf{D}]_{\mathcal{L}} - \lambda I = \begin{bmatrix} D_{xx} - \lambda & D_{xy} & D_{xz} \\ & D_{yy} - \lambda & D_{yz} \\ (Sym) & & D_{zz} - \lambda \end{bmatrix} = [\mathbf{l}_1 \ \mathbf{l}_2 \ \mathbf{l}_3]$$

has rank two, and its column vectors $\mathbf{l}_i$ span the two-dimensional column-space of the matrix. We then calculate the pair-wise cross-products of $\mathbf{l}_i$, which we term *cross vectors*:

$$\begin{aligned} \mathbf{c}_1 &= \mathbf{l}_1 \times \mathbf{l}_2 \\ \mathbf{c}_2 &= \mathbf{l}_1 \times \mathbf{l}_3 \\ \mathbf{c}_3 &= \mathbf{l}_2 \times \mathbf{l}_3 \end{aligned}$$

Theoretically, all the cross vectors $\mathbf{c}_i$ are parallel, and are all eigenvectors associated with eigenvalue λ. Given the limited numerical precision of floating point representation, however, a better method of computing the eigenvector in question is to average the directions of the cross vectors. This is complicated by the fact that two cross vectors may point in opposite directions. A reliable way of correcting this is to find the longest cross vector $\mathbf{c}_m$, and change the change the sign of the other two $\mathbf{c}_i$ so that $\mathbf{c}_i \cdot \mathbf{c}_m \geq 0$. The average of the sign-adjusted $\mathbf{c}_i$ has proven to be an accurate means of computing the eigenvector.

If the multiplicity of eigenvalue λ is two (say, $\lambda_1 = \lambda_2$), then the rank of $[\mathbf{D}]_{\mathcal{L}} - \lambda I$ is one, and the column vectors $\mathbf{l}_i$ are all theoretically orthogonal to the eigenspace for eigenvalue λ. This in turn means they are all parallel to $\mathbf{e_3}$, the eigenvector associated with λ_3. We use the sign-adjusted average of $\mathbf{l}_i$ to get a reliable estimate of $\mathbf{e}_3$. Then, by picking any two perpendicular vectors both orthogonal to $\mathbf{e}_3$, we find eigenvectors $\mathbf{e}_1$ and $\mathbf{e}_2$ to span the eigenspace. If all eigenvalues are equal, then any three mutually orthogonal vectors may be chosen as eigenvectors.

3.1.2 Introduction and Related Work

Glyphs, or icons, depict multiple data values by mapping them onto the shape, size, orientation, and surface appearance of a base geometric primitive [154]. In glyph-based tensor visualization, the shape and orientation information are derived from the eigenvalues and eigenvectors of the tensor [164]. Section 2.3.1 described how the eigenvalues are computed from tensor invariants.

The tensor field is defined as a function which maps from the spatial domain to the space of symmetric tensor values:

$$\mathbf{D} : \mathbb{R}^3 \ \mapsto \ Sym_3 \ .$$

Glyph-based visualizations typically sample $\mathbf{D}$ on a regular grid, and then transform some base glyph surface geometry G_0, such as a sphere, according to the tensor sample attributes and location. The matrix representation and glyph transformation are computed in laboratory frame. Recall from Section 2.1.4 (Equation 2.12), that the matrix $[\mathbf{D}(\mathbf{x})]_{\mathcal{L}}$ can be diagonalized:

$$[\mathbf{D}(\mathbf{x})]_{\mathcal{L}} = R \Lambda R^{\mathsf{t}} \ ,$$

where Λ is the diagonal matrix of eigenvalues, computed as described in Section 2.3.1, and the columns of matrix R are the eigenvectors, computed as described above. The tensor glyph $G(\mathbf{x})$ at field location $\mathbf{x}$ is then defined by [164]:

$$\begin{aligned} \mathbf{D}(\mathbf{x}) &= R\Lambda R^{-1} \\ \Rightarrow G(\mathbf{x}) &= \{R\Lambda\mathbf{p} + \mathbf{x} \;\; \forall\, \mathbf{p} \in G_0\} \; . \end{aligned} \tag{3.1}$$

That is, given the diagonalization $R\Lambda R^{-1}$ of the tensor sample at location $\mathbf{x}$, the glyph geometry is mapped through $R\Lambda$, and then translated by $\mathbf{x}$. By *not* applying rotation R^{-1}, the axis-aligned features of G_0 (such as the edges of a unit cube) become representations in $G(\mathbf{x})$ of the tensor eigenvalues and eigenvectors. Glyph geometries used in the DT-MRI literature include cubes [207], cylinders [198], octahedra [189], and, most commonly, spheres [153, 152, 21, 110]. Because eigenvectors are known only up to line orientation (Section 2.1.4), practical tensor glyph geometries are constrained to shapes with 180 degree rotational symmetry.

Strictly speaking, the superquadric shape we propose for the glyph geometry is a *superellipsoid*, which was popularized by the architect Piet Hein in the 1960s [67]. The larger class of shapes properly referred to as *superquadrics*, which includes superellipsoids, superhyperboloids of one and two pieces, and supertoroids, were presented by Barr [10] in the context of solid modeling for computer graphics. Subsequent work by Pentland [148, 149] and Bajcsy and Solina [7] applied superellipsoids in computer vision, using the tunable geometry (in addition to a smooth deformation) to simultaneously segment objects from a scene and fit their shape parameters. Jaklic et al. review this area of computer vision [91].

Superquadric tensor glyphs build on previous research by Shaw and Ebert which applies superquadrics to glyph-based visualization [167, 168, 58, 59]. They describe how parameterizing shape variations to encode data variables should enable effective and intuitive "perceptualizations," given that distinguishing shape from contours and shading is apparently a *preattentive* process [143]. Offering a continuous two-parameter space of shapes, superquadrics are a natural choice for a tunable geometric primitive. The ability to discern differences between rendered superquadrics was experimentally quantified [168], and the superquadric glyphs were successfully used for document corpus visualization [167] and scientific visualization of magnetohydrodynamic flow [58, 59]. The novel contribution of this section is the application of superquadrics as a *tensor* glyph rather than simply a multivariate glyph. This requires selecting an intuitive subset of the superquadric parameter space to encode tensor shape, and ensuring that the display of tensor orientation faithfully conveys the symmetries that can arise in the tensor eigensystem (described in Section 2.3.2).

Some DT-MRI voxels within the largest white matter structures of the brain (such as the corpus callosum) exhibit purely linear anisotropy at the spatial scale of the individual voxels, because the whole voxel region is homogeneously uni-directional. However, the complex branching and crossing of the white matter tracts, combined with the limited resolution of the DT-MRI modality, produces many measurements with significant planar anisotropy. Visualizing the locations and orientation of planar anisotropy is a step towards understanding the complex nature of white matter connectivity. The first step in this direction was taken by Wiegell et al., using cylindrical glyphs (aligned with the minor eigenvector $\mathbf{e}_3$) to visualize the orientation and context of regions of high planar anisotropy [198]. The "stream-surface" approach of Zhang et al. provides an alternative

way of visualizing planar anisotropy regions, by integrating a surface patch along the two orthogonal directions with highest diffusivity [204, 203].

3.1.3 Motivation

Evaluating existing tensor glyph geometries and their properties is facilitated with an intuitive domain that spans all possible tensor shapes. Such a domain is afforded by the barycentric anisotropy metrics, described already in Section 2.3.4. The barycentric shape space is drawn in Figure 3.1 using cuboid glyphs to emphasize the changes in aspect ratio over the triangular domain. Complete isotropy is at the top corner ($c_s = 1$), and anisotropy increases toward the lower edge.

The advantage of the cuboid glyphs is how their sharp edges give an unambiguous depiction of the glyph orientation, but with this comes their drawback: misleading depiction of under-constrained orientation. Because $c_p = 0 \Rightarrow \lambda_2 = \lambda_3$ for the linear shapes at the left edge of the triangle, computation of the corresponding eigenvectors $\mathbf{e}_2$ and $\mathbf{e}_3$ may return *any* two perpendicular vectors within the plane normal to the principal eigenvector $\mathbf{e}_1$. An analogous problem occurs with the planar shapes along the right edge of the triangle. The cuboid edges depict orientation with a visual clarity that is disproportionate to the low numerical accuracy with which the eigenvectors can be calculated [70]. For intermediate shapes, however, the sharp edges of the cuboids are good at depicting legitimate tensor orientation.

Cylinder glyphs resolve this problem by aligning their axis of rotation along the eigenvector for which the numerical accuracy is greatest, as done in Figure 3.2. There is unfortunately a *discontinuity* problem, with a seam down the middle of the shape space. Arbitrarily small changes in the tensor shape can result in discontinuous changes in the glyph direction, even though the precise location of the seam is somewhat arbitrary. Specifically, an alternate definition of c_l, c_p, c_s (normalized by λ_1 instead of $\lambda_1 + \lambda_2 + \lambda_3$ [194]) produces a slightly different distribution of intermediate shapes within an otherwise similar barycentric shape domain. In addition, because cylinders have only one axis of symmetry, cylindrical glyphs depict meaningless orientation for spherical tensors, which have no intrinsic orientation.

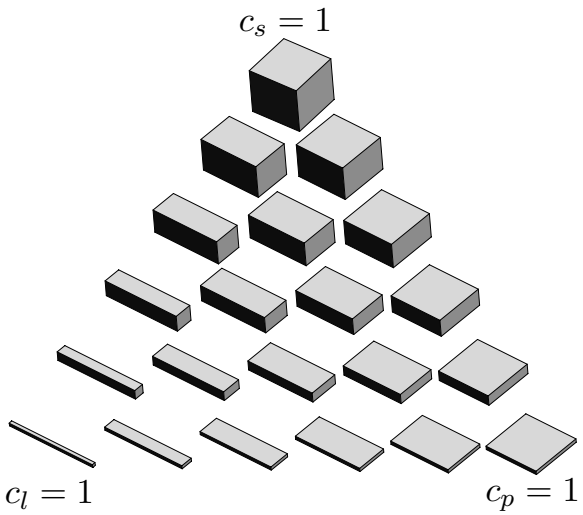


Figure 3.1. Tensor shapes, with cuboids.

Ellipsoidal glyphs, shown in Figure 3.3, avoid all such symmetry problems. There is, however, a problem of visual ambiguity. Glyphs with differing *tensor* shapes exhibit similar *image-space* shapes, with only shading cues for disambiguation. Figure 3.4 demonstrates a pathological example. A wide range of tensors rendered with ellipsoid glyphs can appear similar from one viewpoint (Figure 3.4(a)), though they are clearly different when seen from another viewpoint (Figure 3.4(b)). This example is important because it demonstrates that even standard, intuitive glyph geometries can sometimes fail to properly convey data attributes.

3.1.4 Method

The problems of asymmetry and ambiguity can be addressed with a glyph geometry that changes according to the underlying tensor shape. Ideally, the best of Figures 3.1, 3.2, and 3.3 could be combined: cylinders for the linear and planar cases, spheres for the spherical case, and cuboids for intermediate cases, with smooth blending in between. The general strategy is that *edges on the glyph surface signify anisotropy*: anisotropy implies a difference in eigenvalues, which implies confidence in computing eigenvectors [70], which implies lack of axial symmetry, which can be visually highlighted by a strong edge on the glyph surface. When two eigenvalues are equal, the indeterminacy of the eigenvectors ought to be conveyed by a circular glyph cross-section.

Superquadrics accomplish this goal. They can be parameterized explicitly (useful for generating a polygonal representation):

$$\mathbf{q}_z(\theta,\phi) = \begin{pmatrix} \cos^\alpha\theta\sin^\beta\phi \\ \sin^\alpha\theta\sin^\beta\phi \\ \cos^\beta\phi \end{pmatrix}, \quad \begin{matrix} 0 \le \phi \le \pi \\ 0 \le \theta \le 2\pi \end{matrix}, \tag{3.2}$$

where $x^\alpha = \mathrm{sgn}(x)|x|^\alpha$. The superquadric surface may also be represented implicitly (useful for ray-tracing):

$$q_z(x,y,z) = \left(x^{2/\alpha} + y^{2/\alpha}\right)^{\alpha/\beta} + z^{2/\beta} - 1 = 0. \tag{3.3}$$

Figure 3.5 shows how α and β control superquadric shape. From the entire space of

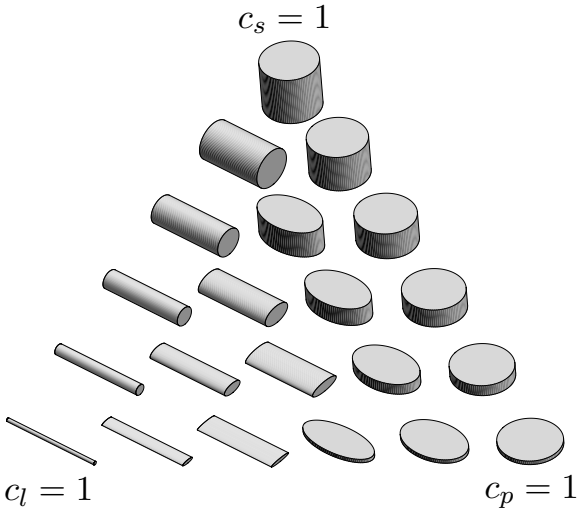


Figure 3.2. Tensor shapes, with cylinders.

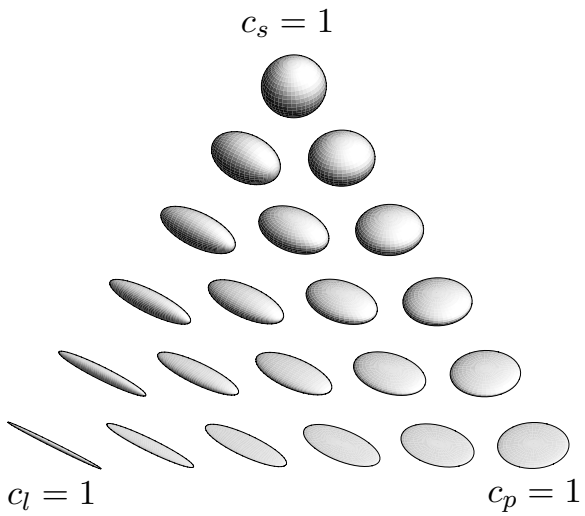
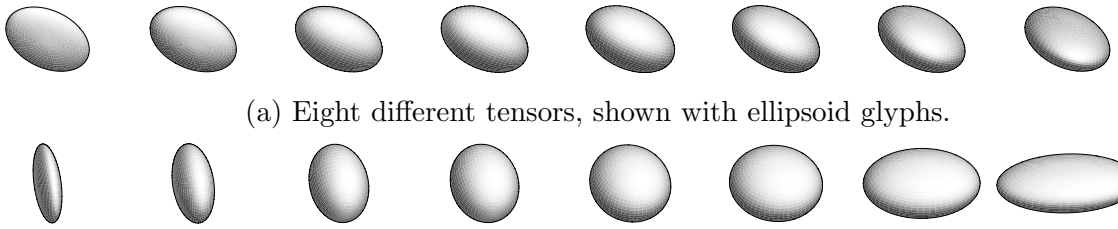


Figure 3.3. Tensor shapes, with ellipsoids.



(a) Eight different tensors, shown with ellipsoid glyphs.

(b) Same eight glyphs, but with a different viewpoint.

Figure 3.4. From some viewpoints, ellipsoids poorly convey tensor shape.

superquadrics defined by $\alpha > 0$ and $\beta > 0$, the superquadric tensor glyphs draw from only a small subset, indicated by a gray triangle in the background of Figure 3.5. The domain of superquadric tensor glyphs is defined by: $\beta \leq \alpha \leq 1$. Note that the formulations of $\mathbf{q}_z$ and q_z are *not* symmetric with respect to axis permutation. Aside from the spherical case, the superquadrics may have axial symmetry around only the z axis (when $\alpha = 1$). Thus, as a counterpart, we also use superquadrics which attain axial symmetry around the x axis:

$$\mathbf{q}_x(\theta, \phi) = \begin{pmatrix} \cos^\beta \phi \\ -\sin^\alpha \theta \sin^\beta \phi \\ \cos^\alpha \theta \sin^\beta \phi \end{pmatrix} , \quad \begin{matrix} 0 \leq \phi \leq \pi \\ 0 \leq \theta \leq 2\pi \end{matrix} , \tag{3.4}$$

$$q_x(x, y, z) = \left(y^{2/\alpha} + z^{2/\alpha}\right)^{\alpha/\beta} + x^{2/\beta} - 1 = 0. \tag{3.5}$$

With these ingredients, superquadric tensor glyphs are now defined in terms of the geometric anisotropy metrics c_l, c_p, and a user-controlled edge sharpness parameter γ:

$$
\begin{aligned}
c_l \geq c_p \implies & \begin{cases} \alpha = (1 - c_p)^\gamma \\ \beta = (1 - c_l)^\gamma \\ \mathbf{q}(\theta, \phi) = \mathbf{q}_x(\theta, \phi) \\ q(x, y, z) = q_x(x, y, z) \end{cases} \\
c_l < c_p \implies & \begin{cases} \alpha = (1 - c_l)^\gamma \\ \beta = (1 - c_p)^\gamma \\ \mathbf{q}(\theta, \phi) = \mathbf{q}_z(\theta, \phi) \\ q(x, y, z) = q_x(x, y, z) \ . \end{cases}
\end{aligned}
\tag{3.6}
$$

Superquadric tensor glyphs use Equation 3.6 to define a base glyph geometry G_0 that is made into a tensor visualization via Equation 3.1. The rationale for how α and β are defined in Equation 3.6 can be understood with reference to Figure 3.5. For tensors that are more linear than planar ($c_l \geq c_p$), the glyph shape becomes more distinctly cylindrical as c_l increases and β decreases. True axial symmetry is only present when $c_p = 0 \Rightarrow \alpha = 1$. As the planar component increases with c_p, the shape gradually tends away from axial symmetry due to lower α, increasing the prominence of edges around the glyph circumference. Analogous reasoning holds for $c_l < c_p$. When $c_l = c_p$, $\alpha = \beta$, and $q_x(x, y, z) = q_z(x, y, z)$, in which case the x axis (Equations 3.4, 3.5) and the z axis (Equations 3.2, 3.3) superquadrics are identical. Thus, like cylinders (Figure 3.2), superquadric tensor glyphs *do* have a seam between the linear and planar sides of the shape space, but the seam is *mathematically* continuous. Figure 3.6 illustrates how the parameterization change may have an effect on a tessellation-based surface representation.

Figure 3.7 illustrates superquadric glyphs with the same tensors, lighting, and viewpoint as used in Figures 3.1, 3.2, and 3.3. The glyphs have the necessary symmetry

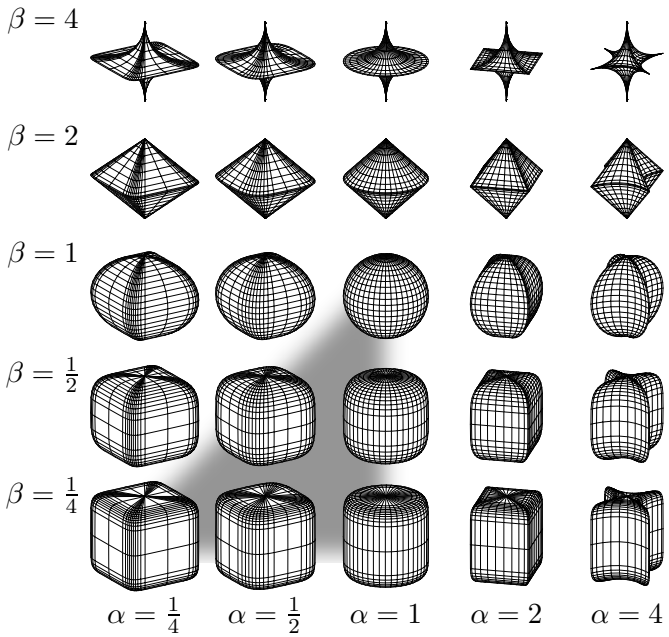
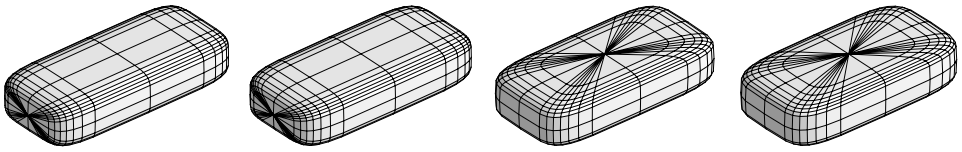


Figure 3.5. Superquadrics defined by Equation 3.2. The gray triangle in the background indicates the subset of the shape space employed by superquadric tensor glyphs. Edges indicate the tessellation resulting from uniform steps in ϕ and θ.



$c_l = 0.33$	$c_l = 0.31$	$c_l = 0.29$	$c_l = 0.27$
$c_p = 0.27$	$c_p = 0.29$	$c_p = 0.31$	$c_p = 0.33$

Figure 3.6. Parameterization change across the linear/planar seam, from $c_l > c_p$ to $c_l < c_p$ ($\gamma = 3$).

properties of ellipsoids, but they tend to convey orientation and shape more clearly by imitating cylinders and cuboids where appropriate. The edge sharpness parameter γ controls how rapidly edges form as c_l and c_p increase, allowing the user to control the visual prominence of orientation information at low anisotropy levels. Ideally, application characteristics would enable an informed choice of γ: visualizations of noisy measurements might use a lower (more conservative) γ than would visualizations of high-precision simulation data. Note that pure ellipsoids can be recovered as a special case, with $\gamma = 0$.

Figure 3.8 shows how superquadric glyphs are better at conveying shape than the ellipsoid glyphs in Figure 3.4, using the same tensors, viewpoint, and lighting. For example, the third and sixth glyphs from the left have precisely linear ($c_p = 0$) and planar ($c_l = 0$) shapes, respectively. The existence and the orientation of the resulting axial symmetry is easier to see with superquadrics than with ellipsoids.

3.1.5 Results

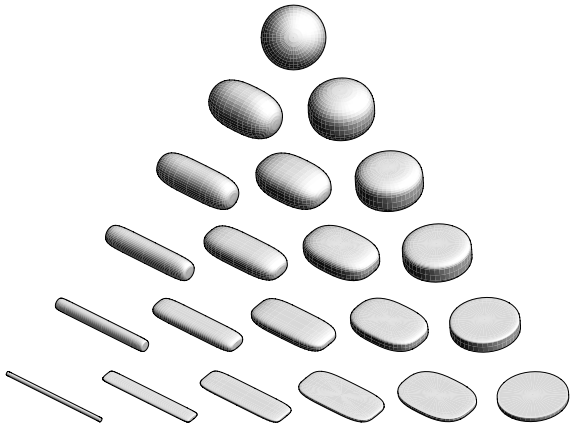
Before showing results from superquadric tensor glyphs, we define here the spherical colormap which will be used in numerous figures throughout the dissertation.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = c_l \begin{bmatrix} |\mathbf{e}_1 \cdot \mathbf{x}| \\ |\mathbf{e}_1 \cdot \mathbf{y}| \\ |\mathbf{e}_1 \cdot \mathbf{z}| \end{bmatrix} + (1 - c_l) \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} . \tag{3.7}$$

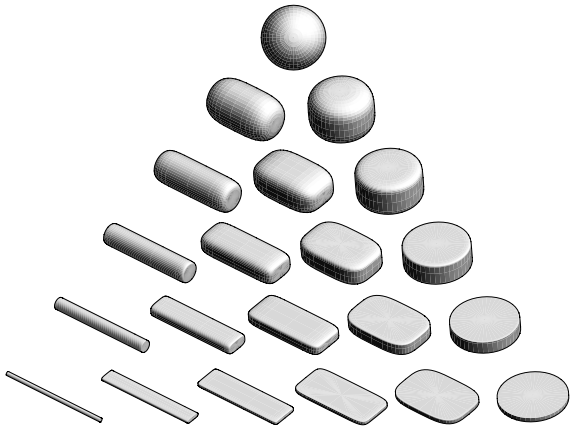
Though far from an unambiguous mapping of the principal eigenvector $\mathbf{e}_1$ to color, this has become the *de facto* coloring scheme within the DT-MRI literature, and certain major anatomical features are recognized by it [151, 142]. Equation 3.7 actually deviates slightly from convention by using c_l rather than FA as the anisotropy measure to desaturate color. Using c_l is more mathematically justified, however, since FA is nonzero for planarly anisotropic tensors (for which $\mathbf{e}_1$ is not uniquely defined), while c_l is by definition zero.

The figures in this Section are created by ray-tracing. Ray-tracing is a method of image synthesis based on computing light along rays from the virtual camera into the synthetic scene, through the pixel locations on the image plane. By nature, ray-tracing has the virtue of sublinear computational time with respect to scene complexity [95], which has motivated its recent use as a means of interactive scientific visualization [145, 144]. Appendix C describes an important implementation detail that facilitates the efficient ray-tracing of superquadric tensor glyphs.

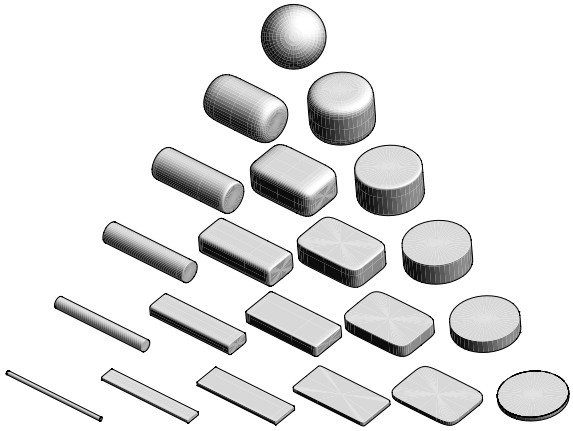
Section 3.1.2 described how one task of diffusion tensor visualization is the identification of regions of high planar anisotropy. For this task, Figure 3.9 compares the effectiveness of superquadric tensor glyphs and ellipsoids for visualizing a portion of an



(a) $\gamma = 1.5$

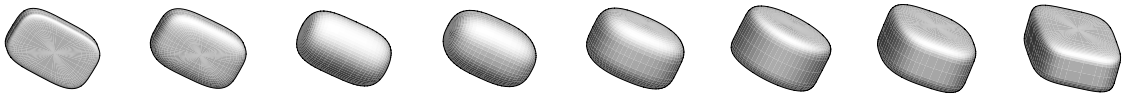


(b) $\gamma = 3.0$

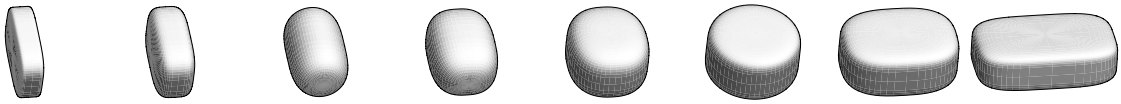


(c) $\gamma = 6.0$

Figure 3.7. Tensor shapes, with superquadric glyphs, and three different values of edge sharpness parameter γ.



(a) Same tensors, viewpoint, and lighting as Figure 3.4(a), but with superquadric glyphs.



(b) Same as Figure 3.4(b), but with superquadric glyphs.

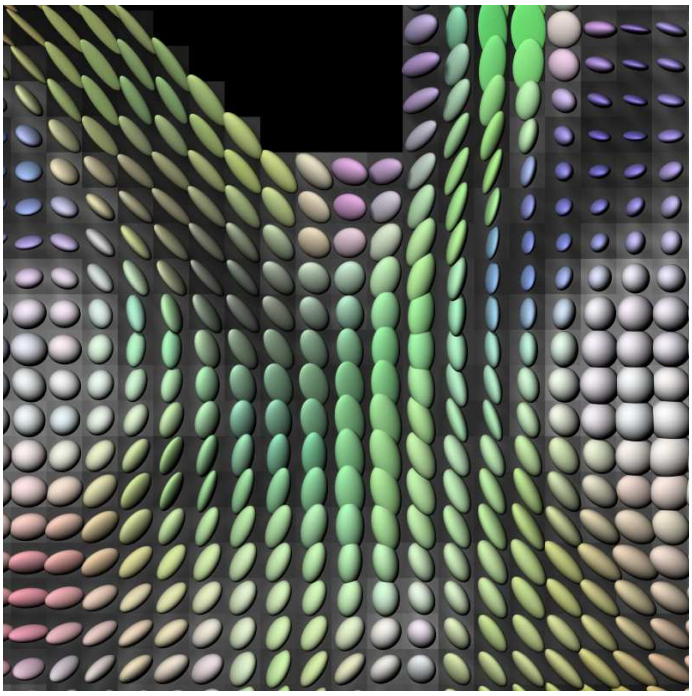
Figure 3.8. Superquadrics convey shape differences more reliably than ellipsoids ($\gamma = 3$).

axial slice through a diffusion tensor dataset, centered on the right half of the splenium of the corpus callosum (the black region is the lateral ventricle). The background squares represent isotropy levels for each sample ("interesting" anisotropic tensors have a darker background and hence greater contrast with the glyph). The RGB coloring of the glyphs is from Equation 3.7. Planarly anisotropic samples are located near the center of the image. With ellipsoids, it is difficult to discern which of the glyphs represent planar anisotropy, and it harder to appreciate the differences in shape that may occur between neighboring samples. Also, the straight edges of the linearly anisotropic superquadric glyphs provide a stronger orientation indication than possible with the rounded contours of ellipsoids.

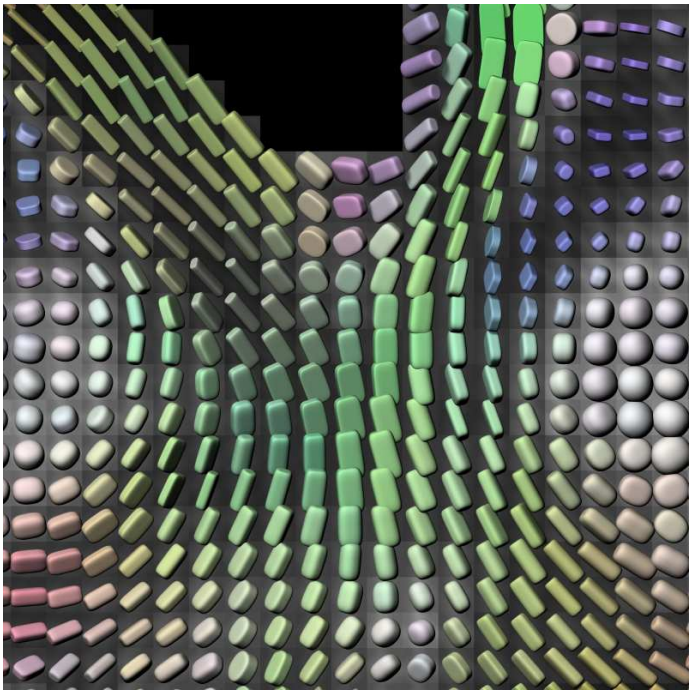
In three-dimensional glyph-based visualizations of tensor volumes, it is important to restrict the number of glyphs, to avoid creating an illegible mass. In diffusion tensors, glyphs may be culled according to an anisotropy threshold (such as FA > 0.5) so that isotropic tensors (belonging to gray matter or cerebral spinal fluid) are hidden, resulting in a coarse depiction of the major white matter pathways. Figure 3.10 uses this method to compare ellipsoid and superquadric glyphs for visualizing half of a diffusion tensor volume, centered again at the right half of the splenium of the corpus callosum. The superquadrics depict the amount and orientation of the planar component in the white matter more clearly than the ellipsoids. Comparing the planar orientation with the direction of adjacent linear anisotropy is an example of a visualization query which is better answered by the new glyph method.

Thought not a focus of this dissertation, DT-MRI has proven useful for measuring the directional structure of the muscular wall of the heart (the myocardium), and this provides the context for a second demonstration of superquadric tensor glyphs. This work was done in collaboration with Dr. Daniel Ennis [61] as part of his doctoral studies at Johns Hopkins University. The orientation of the contractile muscle cells (myofibers) within the myocardium smoothly rotates as a function of position between the epicardial and endocardial surfaces [178]. Diffusion tensor imaging can measure this feature, in that the principal eigenvector $\mathbf{e}_1$ indicates the myofiber direction [88, 165] However, the myocardium is also characterized by a laminar or sheet-like organization [115], although there is relatively little work in assessing this structure through DT-MRI [55].

Figure 3.11 demonstrates the difference between ellipsoid and superquadric glyphs in visualizing myocardial structure. Both glyphs show the twisting of the main fiber direction, but the edges in the superquadric glyphs provide a cleaner depiction of the direction, especially towards the endocardium, where the tensor anisotropy is lower. In

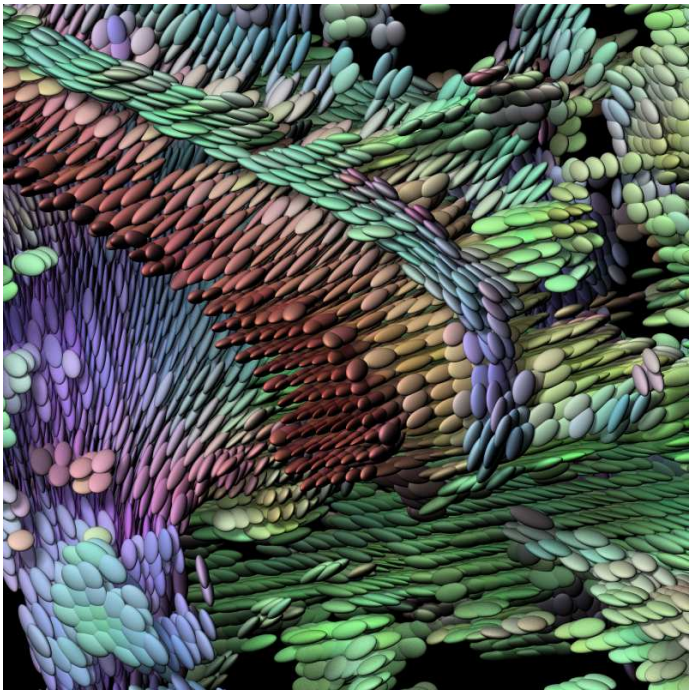


(a) Ellipsoids

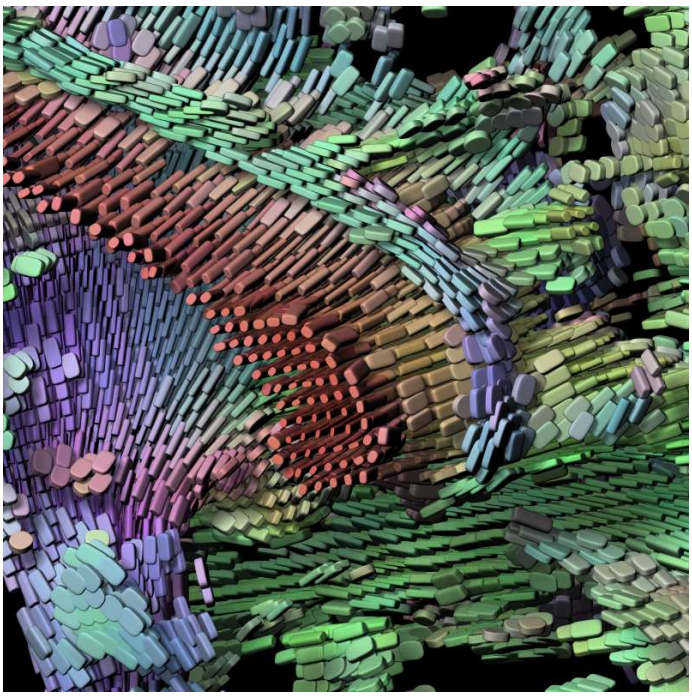


(b) Superquadrics ($\gamma = 3$)

Figure 3.9. Slice of DT-MRI dataset of brain visualized with ellipsoids (top) and superquadrics (bottom).

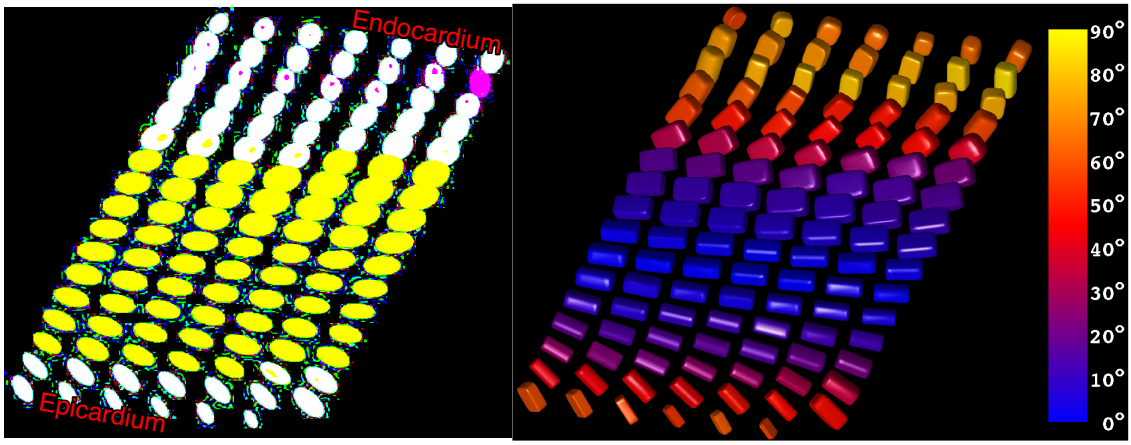


(a) Ellipsoids



(b) Superquadrics ($\gamma = 3$)

Figure 3.10. Three-dimensional region of DT-MRI dataset of brain visualized with ellipsoids (top) and superquadrics (bottom).



(a) Ellipsoids (b) Superquadrics ($\gamma = 4$)

Figure 3.11. Slice of DT-MRI dataset of canine myocardium (left ventricle), using ellipsoids (left) and superquadrics (right). Colormap encodes fiber orientation relative to sampling plane. Figure courtesy of Dr. Daniel Ennis.

addition, at the midwall, the superquadric glyphs clearly show an increase in *orthotropy*: wherein the eigenvalues are all distinct (half-way between linear and planar). This feature had not been previously reported in the cardiac DT-MRI literature, and is significant because the orientation of the planar component is broadly consistent with the purported sheet architecture. Based on these initial findings from the glyph-based visualizations, future work will seek to quantify the orientation and organization of the orthotropic regions.

3.1.6 Discussion

In comparing Figures 3.4(a) and 3.8(a), one could argue that various rendering effects would help clarify the shape differences among the ellipsoids: different lighting, specular highlights, or surface textures, for example. Interactive manipulation and stereo rendering would also help. On the other hand, these modifications would also benefit the depiction of superquadrics. The same diffuse lighting and fixed viewpoint, which failed to distinguish the ellipsoid shapes, were sufficient to differentiate the superquadrics. Using data-driven variable geometry (Equation 3.6), *in addition* to the eigenvalue-based scaling, helps superquadric glyphs convey shape more explicitly than previous tensor glyphs.

It should be pointed out that even though Figures 3.9 and 3.10 were ray-traced with a resolution and sampling (800×800, 49 rays per pixel) which makes interactive rendering unlikely, the superquadric glyph method itself does not rule out interactively, which is an important part of the overall scientific visualization process. On the contrary, polygonal models created by uniform steps in the (θ, ϕ) parameter space of Equations 3.2 and 3.4 can easily create the necessary shapes, due to the fortuitous property that the parameterization slows down near the edges (as indicated in Figure 3.5), thus creating more faces in higher curvature areas. The BioTensor visualization program, for example, uses polygonal superquadric glyphs as one of the interactive tools available for exploring tensor fields [32] (these are seen in Figures 4.8 and 4.9).

The success of the superquadric glyphs rests on the mathematical property that their

underlying geometry changes as function of tensor shape, while the base (spherical) geometry of ellipsoidal glyphs does not. The consequence of this, demonstrated in Figure 3.4, is that the set of ellipsoidal glyphs is closed under the scalings and shears which hold the silhouette fixed. This is known in the computer vision literature as the "Bas-Relief Ambiguity" [25]. Superquadric glyphs do not suffer this ambiguity because scalings and shears will change the tensor shape, which in turn changes the glyph geometry, as shown in Figure 3.8. The glyphs have been enthusiastically adopted by two external collaborators, Dr. Daniel Ennis for the visualization of myocardial diffusion tensors (Figure 3.11) [61], and Dr. Paul Thompson for the visualization of anatomic covariance tensors [104]. Both collaborators have noted that the glyphs need no additional explanation when shown to colleagues, and that people familiar with ellipsoidal tensor glyphs appreciate the shape contrasts created by the superquadrics.

Starting with a more expressive glyph geometry allows further effects (color, textures, etc.) to be saved for encoding additional degrees of freedom that may be required in a more complex visualization application. The best way to enrich *three-dimensional* glyph-based visualizations with extra information is an important direction of future work, since there are usually a number of related field values which should be visualized along with tensors. Inspiration may be drawn from artistic methods of painting and illustration [110, 111, 106, 158]. Specifically, the composition of multiple glyphs into a depiction of larger-scale structure may benefit from context-sensitive and multiscale variation of rendering style [80].

The incentive to create sharp edges in the superquadric glyphs was based on the observation that edges generate a strong visual cue for orientation. However, it is the mathematical property of axial *symmetry* that constrains the glyph to be cylindrical and spherical according to the tensor eigensystem, and the idea of *continuity* that informed the design of an invisible seam through the middle of barycentric shape space. The combination of aesthetic judgment and mathematical constraint may be useful in the design of other visual abstractions for multivariate and tensor visualization.

3.2 Reaction-Diffusion Textures for Visualization

This section describes an application of reaction-diffusion textures, developed in mathematical biology and previously applied to computer graphics, to the task of tensor visualization. The result is a texture of spots (in two dimensions) or "blobs" (in three) which are visualizations of the local tensor attributes. The individual spots are effectively tensor glyphs, and their stochastic placement (a natural consequence of the reaction-diffusion process) allows the overall trends in shape and orientation to be seen clearly.

Note: Most of this material was previously published in [103]. However, the original presentation had an error in the mathematical exposition and implementation (fixed in Equation 3.13), and gave inadequate advice on parameter setting. The original work demonstrated the generalization to three dimensions; below we focus on two-dimensional textures for the sake of brevity. The idea of using reaction-diffusion textures for visualization has since been adopted and extended to visualize vector fields and their uncertainty, in two-dimensions, by Sanderson et at. [162].

3.2.1 Introduction and Related Work

Using reaction-diffusion texture as a tensor visualization tool is based on two separate areas of previous work: flow visualization with textures, and reaction-diffusion texture

synthesis for computer graphics. The spot noise method deforms a texture of randomly positioned spots according to local flow properties [188, 50]. Line Integral Convolution (LIC) blurs a white noise texture along the streamline direction [37, 174]. The problem of extending these methods to volumetric flow has been addressed in various ways. Three-dimensional spot noise textures can be rendered by an extension of the "splatting" method of scalar field volume rendering [197, 45, 46], or by rendering the texture only near contour surface [126]. Three-dimensional LIC can be made more comprehensible by virtual dye advection [169], color-codings and halos [90], and interactive graphics hardware [157]. A common attribute of these methods is that the texture underlying the visualization is essentially random. An interesting recent exception to this by Li et al. uses a volume of rasterized and parameterized streamlines to allow arbitrary textures to be interactively placed and animated within a volumetric flow [117].

Textures have also played a role in previous approaches to tensor visualization. Laidlaw et al. used a stripe texture on elliptical brush strokes to convey the out-of-plane component of the principal eigenvector in an approach that can be termed "painterly": the visualization is a composition of different layers, each of which conveys one attribute of the data, by some combination of color or opacity, or by the accumulation of small glyphs mimicking brush strokes [110]. LIC has been advanced as a diffusion tensor visualization method, in which the noise texture is convolved along the eigenvectors. Ou and Hsu were the first to do this, by using traditional LIC along the principal eigenvector as the input to a second stage which blurs along the secondary eigenvector, to visualize myocardial structure [139]. Sigfridsson et al. perform local frequency-space filtering of a noise texture according to tensor eigenvalues and eigenvectors to form volume which is rendered in combination with interspersed glyphs [171]. The HyperLIC approach of Zheng and Pang adapts LIC by performing filtering of a noise texture within a polygonal area defined and deformed by the local tensor attributes [205].

We prefer to base our tensor visualizations on texture which is more organized or structured than random. One source of organized textures is the simulation of a combination of chemical undergoing simultaneous reaction and diffusion. Turing proposed reaction-diffusion equations as a mathematical model for biological *morphogenesis*: the emergence of organized shape and structure essential to the growth of an organism [183, 135]. Turing's paper describes systems of nonlinear partial differential equations modeling the reactions between a number of chemicals (called "morphogens"), which undergo continuous reaction (creation and destruction) and diffusion. Simulating the reaction-diffusion system results in an assignment of morphogen levels at every node of the simulation grid, and the results are visually assessed by mapping a morphogen concentration level to a grayscale image intensity. Discrete implementations of reaction-diffusion systems became a popular method in computer graphics for generating textures with the development of methods for normalizing the density of texture on parameterized surface and polygonal models, and for generating a rich variety of texture patterns [184, 199]. The Gray-Scott reaction-diffusion equations provide another source of organized textures [147].

To a large extent, the value of reaction-diffusion textures as the basis of tensor visualization is in how the texture spots tend to organize and space themselves according to the underlying field attributes. In this sense, the method addresses the problem of glyph and streamline placement in vector and tensor visualization. Turk and Banks gathered adjacent and similar vector streamlines until a minimal representative set remains [185]. For visualizing the vector and tensor attributes of two-dimensional incompressible flow, Kirby et al. draw arrow and ellipsoid glyphs in two visually distinct layers, with a

placement strategy that minimizes overlap between glyphs within each layer [106].

3.2.2 Turing's Reaction-Diffusion Formulation

Turing's presentation of the mathematical properties of morphogenesis is largely theoretical, although his Section 10 gives two concrete examples of a system of chemicals undergoing continuous reaction and transformation, and includes a numerical solution of the first system on a closed ring of 20 nodes. Bard and Lauder examined Turing's second chemical system, verifying with computer simulations that in one dimension it generates the periodic pattern Turing predicted [9]. Significantly, they also showed that in two dimensions, the pattern of spots which develops is organized in character, but not strictly regular, and not entirely predictable. Thus, although unlikely as a mechanism for limb and finger growth, Turing's theory may apply to pigmentation patterns in skin and fur, consistent with how the approach has been adopted in computer graphics. Turk, for example, starts with a discretization of Turing's second system, expressing the relationship between morphogens a and b as [184]:

$$\begin{aligned} \Delta a_i &= s(16 - a_i b_i) + d_a(a_{i+1} - 2a_i + a_{a-i}) \\ \Delta b_i &= \underbrace{s(a_i b_i - b_i - \beta_i)}_{\text{reaction}} + \underbrace{d_b(b_{i+1} - 2b_i + b_{i-1})}_{\text{diffusion}} . \end{aligned} \tag{3.8}$$

The variables a_i and b_i represent the morphogen quantities at node i in the (linear) grid. The morphogen diffusivities d_a and d_b are global constants, and s controls the contribution of the reaction terms. The value β_i is equal to 12, based on Turing's exposition. However, Turk includes in β a small amount of additive noise, the purpose of which is to move the system away from the unstable equilibrium of the initial conditions defined by $a_i = b_i = 4$ for all i. Turk demonstrates numerical solutions of this system in one and two dimensions with $s = 0.03125$, $d_a = 0.25$, and $d_b = 0.0625$ [184].

The diffusion terms in Equation 3.8 represent the standard three-point mask for measuring second derivatives by discrete convolution. The system may be expressed in a continuous and dimensionally general manner as:

$$\begin{aligned} \frac{\partial a}{\partial t} &= s(\alpha - ab) + d_a \nabla^2 a \\ \frac{\partial b}{\partial t} &= s(ab - b - \beta) + d_b \nabla^2 b . \end{aligned} \tag{3.9}$$

We have made two modifications based on the implementation and experience of Lawlor [112]. First, we have replaced the constant 16 in the first equation with a variable growth constant α. The (α, β) parameter space offers additional control of the form the textures which emerge from the system. Second, we keep α and β fixed as global parameters, moving the low amplitude noise out of β and into the initial concentration fields a and b.

We discretize the equations in space and time on a regular grid:

$$\begin{aligned} \Delta a &= \Delta t(s(\alpha - ab) + d_a L * a) \\ \Delta b &= \Delta t(s(ab - b - \beta) + d_b L * b) , \end{aligned} \tag{3.10}$$

where L is the mask that implements the Laplacian by convolution. The two-dimensional mask is:

$$L = \frac{1}{\Delta x^2} \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} . \tag{3.11}$$

In the interests of simplicity in implementation, we use simple Euler integration to solve the systems; Δt must therefore be small enough to avoid instabilities. An implicit solution such as Crank-Nicholson [44] has greater stability, and a framework such as multigrid (as proposed by Lawlor [112]) can decrease the computational time.

Figure 3.12 shows the results from Equation 3.10 for a small range of (α, β) settings. The extreme sensitivity of the texture patterns to such parameter settings is the most problematic aspect of using a reaction-diffusion system for visualization purposes. Note that while Turing suggested $(\alpha, \beta) = (16, 12)$ for one-dimensional patterns, spots seem to be more reliably formed with $(\alpha, \beta) = (16.1, 11.9)$. This parameter variation is apparently controlling something akin to the surface tension of the spot boundaries, and our parameter choice is based on the goal of creating well-formed spots which act as individual glyphs.

Figure 3.13 shows how the solution behaves as a function of the Euler time step Δt. Notice that as long as Δt is small enough to avoid divergence, using the same initial noise conditions will result in essentially an identical texture, which is indirectly determined by the initial noise pattern.

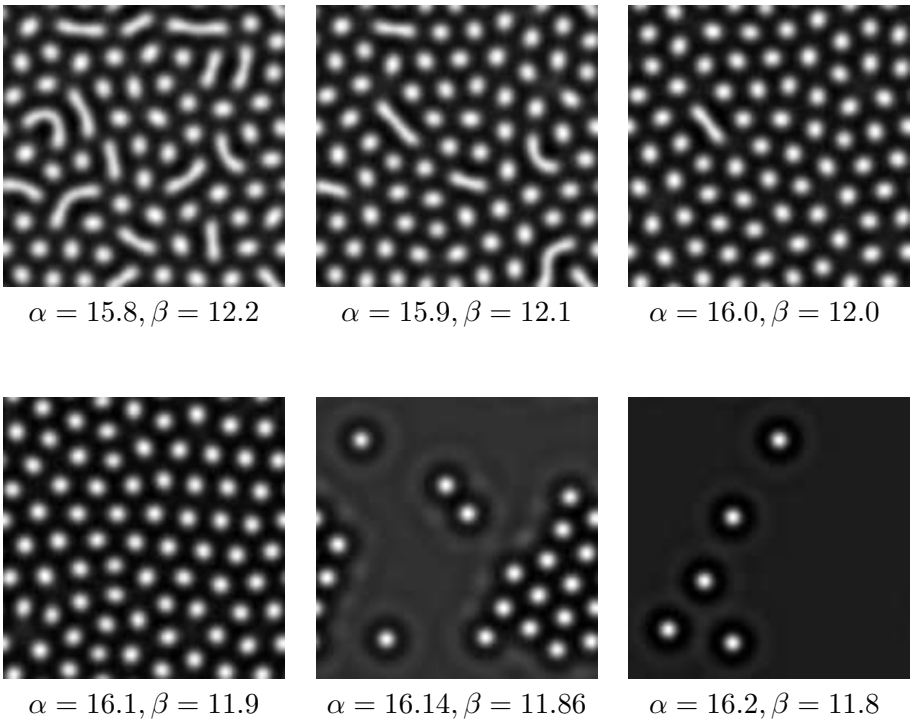


Figure 3.12. Results from Equation 3.10 on a 128×128 grid, using $\Delta t = 0.5$, $\Delta x = 1.0$, $d_a = 0.25$, $d_b = 0.0625$, $s = 0.0125$, and the specified values of α and β. The amount of morphogen a is shown in grayscale. In all cases, number of iterations is 20000.

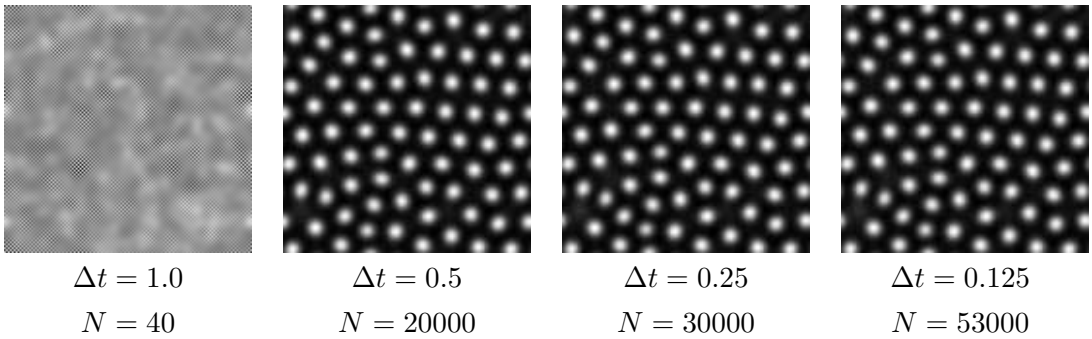


Figure 3.13. Demonstration of lowering the Euler time step Δt. If Δt is too large, the system diverges (left-most image). If it converges, it tends to converge to the same texture, assuming that the number of iterations N is large enough.

3.2.3 Tuning Textures with Tensors

The reaction-diffusion texture is turned into a tensor visualization by locally adjusting the diffusion properties according to the diffusion tensor field. Specifically, Equation 3.9 is replaced with:

$$\begin{aligned} \frac{\partial a}{\partial t} &= s(\alpha - ab) + \nabla \cdot (\mathbf{D}_a \nabla a) \\ \frac{\partial b}{\partial t} &= s(ab - b - \beta) + \nabla \cdot (\mathbf{D}_b \nabla) b \end{aligned} \tag{3.12}$$

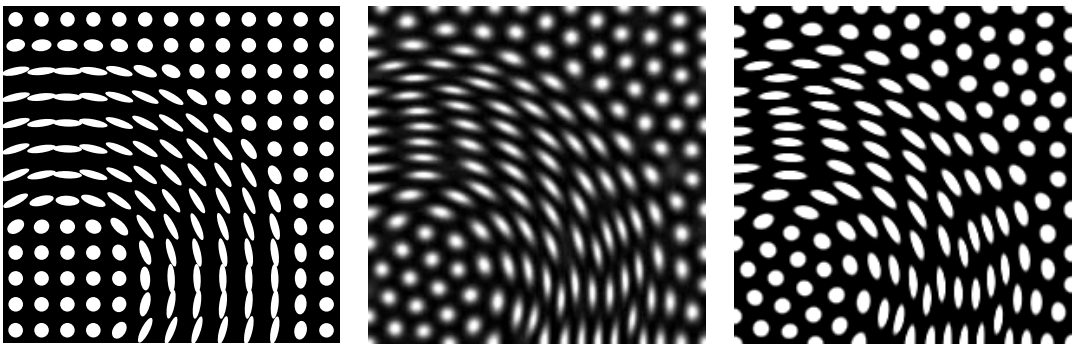
The diffusion terms should be recognized from Equation 2.21 as Fick's second law in inhomogeneous anisotropy. For a general concentration c and diffusion tensor $\mathbf{D}$, this can be expanded (with index notation) as:

$$\nabla \cdot (\mathbf{D} \nabla c) = \frac{\partial}{\partial x_i}(D_{ij} \frac{\partial c}{\partial x_j}) = D_{ij} \frac{\partial^2 c}{\partial x_i \partial x_j} + \frac{\partial D_{ij}}{\partial x_i} \frac{\partial c}{\partial x_j} \tag{3.13}$$

The first term is the double contraction of $\mathbf{D}$ with the Hessian of $\mathbf{c}$. The second term vanishes where the anisotropy is homogeneous. We use standard first and second central differences to implement all these derivatives (Section 2.1.8).

Two implementation details should be noted. First, to ensure that the texture spots cover approximately the same area as a tensor sample, the tensor data should be up-sampled by some factor when mapped onto the reaction-diffusion simulation grid. A larger up-sampling factor produces a higher-resolution visualization, but at the expense of more computation. Earlier work [103] used an upsampling factor of eight, below we use between 12 and 15. The upsampling is performed as a preprocess using a cubic filter (Section 2.1.8). Second, because the diffusion length is proportional to the square root of the diffusivity, rather than the diffusivity itself (Equations 2.19 and 2.20), the spots which emerge from the texture will be scaled by the square roots of the tensor eigenvalues. To correct this, we replace each tensor $\mathbf{D}$ on the upsampled grid with $\mathbf{D}^2$.

Figure 3.14 show results from a synthetic tensor dataset. The texture spots have the

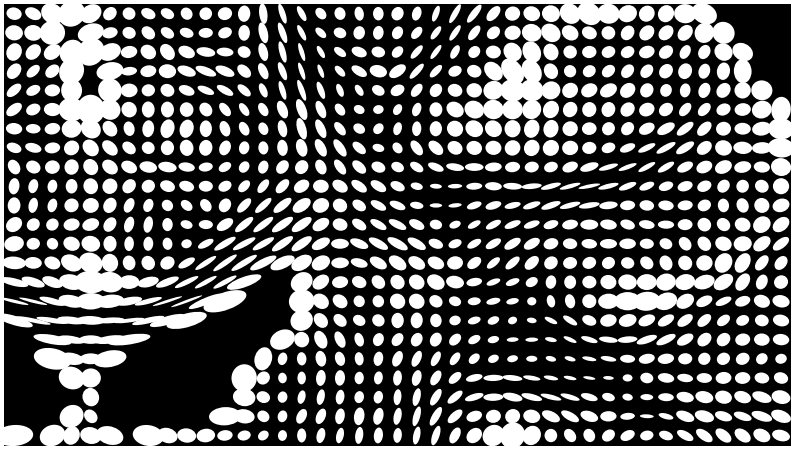


(a) Ellipse glyphs (b) Reaction-diffusion texture (c) Texture thresholded to resemble (a)

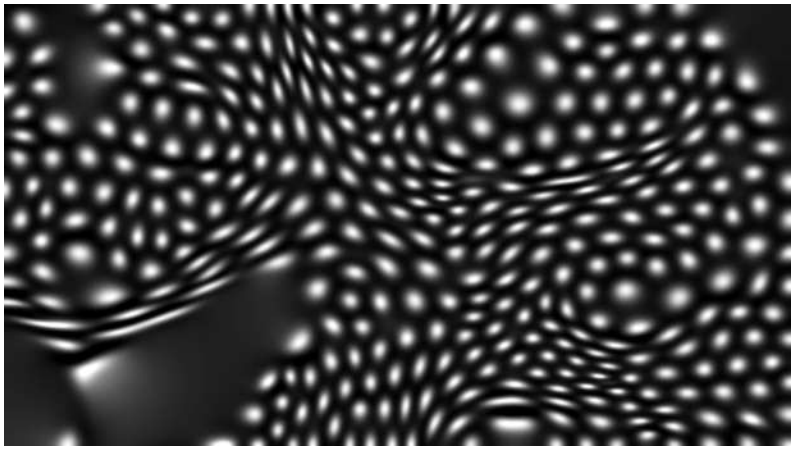
Figure 3.14. Synthetic tensor dataset of a slice of torus, visualized by ellipse glyphs (a), and a reaction-diffusion texture (b). Thresholding the texture (c) confirms that the spots have the correct aspect ratio.

same orientation and shape as the ellipse glyphs, but are positioned to avoid any overlap. The packing naturally arises from the reaction-diffusion equations. In isotropic regions, the spots form a hexagonal grid, and within the anisotropic region the packing ensures that there are no large gaps.

Figure 3.15 demonstrates the method on a portion of an actual DT-MRI scan. Because



(a) Ellipse glyphs



(b) Reaction-diffusion texture

Figure 3.15. Portion of coronal slice of DT-MRI dataset, including the ventricles in the lower-left, visualized by ellipse glyphs (a), and a reaction-diffusion texture (b).

this is only a two-dimensional coronal slice (along the Y axis), only the D_{xx}, D_{xz}, and D_{zz} components of the diffusion tensor contribute to the texture formation. This figure demonstrates how the placement of spots in the texture better follows the underlying features of the data, compared to the strict grid of ellipses. Diffusivity is highest inside the ventricles, between the hemispheres, and within the sulci, as these are locations with cerebral spinal fluid. The ellipse glyphs are larger in these regions, and hence overlap. More troubling from a visualization standpoint is that, the ellipse glyphs overlap when

the direction anisotropy is aligned with the underlying grid. On the other hand, the distribution of the texture spots is in accordance to their size and shape, allowing the texture to more closely follow the curving path of white matter as it projects towards the cortical surface. An unfortunate aspect is that spots on a boundary of cerebral spinal fluid tend to spill out.

The reaction-diffusion textures provide the basis for a "painterly" approach to building up a visualization, similar in spirit to earlier work by Laidlaw [110], shown in Figure 3.16. The small images at the top of the figure show the ingredients to this image: fractional

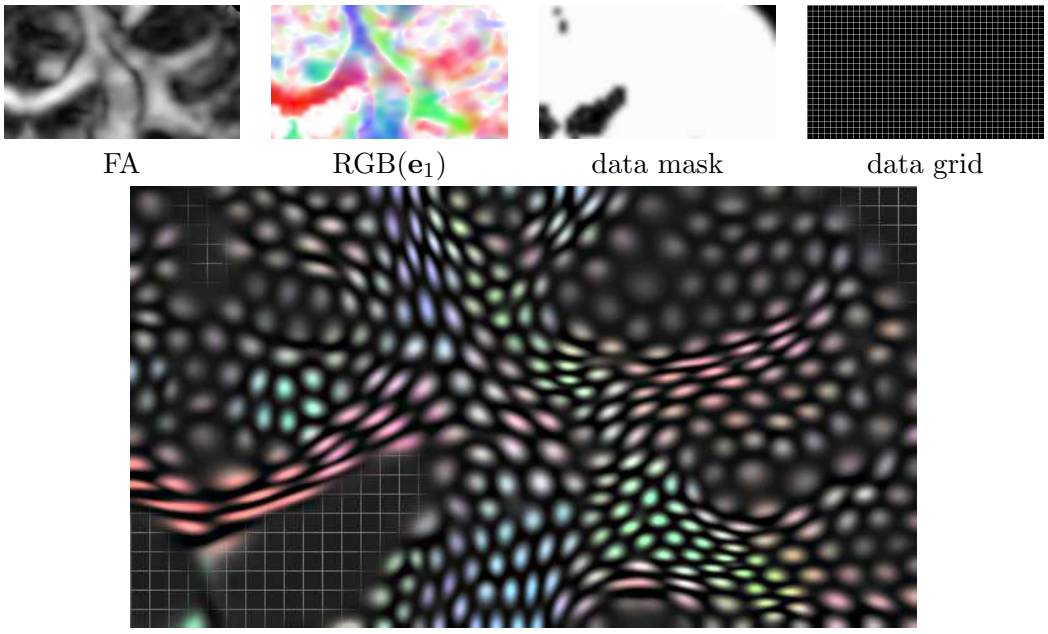


Figure 3.16. Reaction-diffusion texture-based visualization.

anisotropy FA, the RGB colormap of principal eigenvector $\mathbf{e}_1$ (Equation 3.7), a mask covering the interesting data sample locations, and a grid indicating the underlying data resolution, prior to upsampling. The product of the reaction-diffusion texture, the FA image, and the RGB colors, is masked and composited over the grid. The RGB coloring of the final image shows three-dimensional directional information, such as the superior longitudinal fascicle, the green area in the lower right. As mentioned in Section 2.3.3, this is a region which is primarily direction anterior-posterior (into and out of a coronal slice), but with a significant planar anisotropy component, which is clearly shown in the texture visualization.

3.2.4 Discussion

We have describe how to coax Turing's model of morphogenesis into visualizations of diffusion tensor data. As this work represents the first time such textures have been used for scientific visualization, the "success" of this proof-of-concept work rests in a visual comparison of the images in Figures 3.14 and 3.15. For further validation, one

could do a shape quantification of the texture spots with a covariance matrix, and verify that it matches the diffusion tensor at the corresponding location, however Figure 3.14 suggest that the match will be very good. Such a technique could be used, however, to fix or remove any errant spots. Further extending the sort of effects demonstrated in Figure 3.16 could allow additional data attributes to be indicated, such as noise or uncertainty [93, 162].

As a visualization tool, reaction-diffusion textures have numerous short-comings: a number of parameters had to be adjusted in order for the textures to form as desired (so that the individual spots serve as glyphs), the method of parameter adjustment was essentially trial and error, and the computational expense of generating the texture was significant. Furthermore, there is little in the way of a user interface to this method. The user has no way of adjusting spot placement, other than restarting the simulation and hoping that the texture will form differently.

Still, the effectiveness of Figure 3.16 suggests that such textures can improve visualizations. Future work, however, should probably focus on the overall effect that the textures achieve, rather than adjusting specific properties of the reaction-diffusion system itself. Specifically, the major result of this method is that optimizing glyph placement is an important step in creating a high-quality visualization. To our knowledge this specific problem has not been examined in the context of diffusion tensor visualization. Reaction-diffusion patterns provide an indirect way of solving the optimization, in the sense that texture seems to optimize the density of packing of spots. Even though reaction-diffusion implementations based on graphics hardware will be significantly faster, we feel a direct computational solution to optimizing glyph packing and placement is called for.

CHAPTER 4

FIBER TRACKING WITH TENSORLINES

Long-range communication between separate parts of the brain is conducted over pathways, called fiber tracts, in which collections of axons and neuronal filaments run in parallel [73]. The principal eigenvector of the diffusion tensor, the direction along which the apparent diffusion coefficient is highest, is aligned with the direction of the major white matter pathways [153, 122, 24, 14]. Basser proposed the technique of *fiber tractography*, in which the direction information from the diffusion tensor field is spatially integrated to create a geometric model for the white matter pathway [13, 20]. Provided that one understands that the physical scales of individual axons (microns) and individual voxels (millimeters) are roughly three orders of magnitudes apart [24], and that therefore tractography can at best indicate aggregate directional organization, fiber tractography can extract interesting structural information from DT-MRI datasets.

This chapter presents a method for tractography that incorporates the whole diffusion tensor, rather than simply its principal eigenvector, into the integration of fiber pathways. The two main motivations for this approach are the phenomenon of partial voluming, and the fact that some regions of the brain contain multiple crossing fibers directions. As described in Section 2.3.3, both of these situations can produce anisotropy which is more planar in shape than linear. Extraction of putative white matter fiber pathways can be based on the directional information provided by planar anisotropy, as well as linear anisotropy. We accomplish this by incorporating tensor multiplication into the path integration: the input direction towards a tensor sample is multiplied by the tensor value to help determine the output direction. Note that this is essentially the "power method" of matrix eigenvector computation [70], applied to the task of fiber tracking. We use the term "deflection" to refer to the redirection of a vector by tensor multiplication, and the fiber tracks based on deflection are termed *tensorlines*.

Note: This material was originally presented in [190], joint work with David Weinstein (first author) and Eric Lundberg. The tensorline method in particular, and deflection as the basis of tractography in general, has since been studied in detail by Lazar et al., who have quantified the robustness to noise [113], and demonstrated its ability to extract the major white matter pathways [114]. A fairly recent review of tractography approaches, limitations, and solutions is given by Mori [133].

4.1 Introduction

The goal of fiber tractography is hindered by the low spatial resolution of DT-MRI voxels compared to fine structure of axon pathways, and the physical proximity of pathways oriented in different directions. If we had very high resolution data (perhaps on the order of tens of microns), and the direction of fiber pathways within a voxel was purely homogeneous, then tractography would be straightforward. Unfortunately,

clinical DT-MRI datasets are still relatively low resolution, and like any measurement, they are subject to noise. Still, the directional information contained in tensors, linearly anisotropic or not, can play a role in calculating a fiber tract. Specifically, the direction of the secondary eigenvector in planar anisotropy can help provide a more complete picture of the underlying neural connectivity. To appreciate the range of anisotropy shapes present within actual data, Figure 4.1 shows a histogram of a DT-MRI scan, in the barycentric space defined by the $\{c_l, c_p, c_s\}$ shape metrics (Section 2.3.4). The important property of this histogram is that there is no obvious tendency towards linear anisotropy, suggesting that there is substantial partial voluming taking place throughout the measured volume.

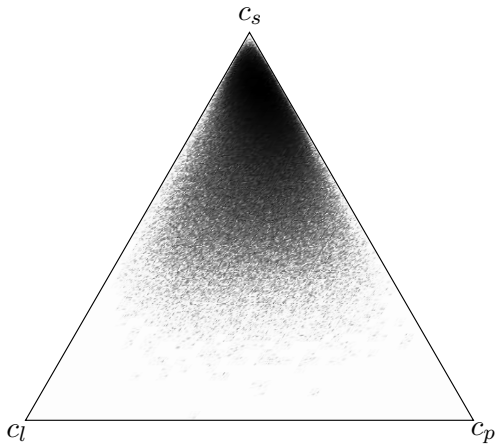


Figure 4.1. Barycentric histogram of a diffusion tensor MRI dataset. The coordinates correspond to the amount of linear, planar and spherical anisotropy in the tensor. The lack of clustering suggests considerable partial voluming is taking place throughout the volume.

Fiber tractography is based on the hyperstreamline method of tensor visualization [51, 52], which applies the streamline method of flow visualization to the field of tensor eigenvectors. Advected along a single eigenvector, hyperstreamlines depict the additional tensor information with an ellipsoidal cross-section which is oriented with the other two eigenvectors, and scaled by the respective eigenvalues.

While hyperstreamlines are useful method for tensor visualization (since they strive to communicate all the tensor attributes along their path), the path of the hyperstreamline is not always ideal for tractography purposes. Whereas streamlines produced the path indicated in light gray in Figure 4.2, which can "get lost" in isotropic regions, our method produced the path of the dark gray *tensorline* from that same image - a path that continued along its present course when it encountered a region of isotropic diffusion.

In Figure 4.3, we see one schematic example of how propagating a streamline according to the $\mathbf{e}_1$ direction of the tensors can be misleading. For explanatory simplicity, we have displayed a two-dimensional slice of data with two-dimensional diffusion tensors indicated by the ellipses. In this figure we see a field which varies from being somewhat linearly anisotropic on the left and right, to nearly isotropic in the middle. We have indicated the $\mathbf{e}_1$-directed streamline with a dotted line and the borders of a hyperstreamline with gray. What is not clear from either of these visualizations is that diffusion near the middle of

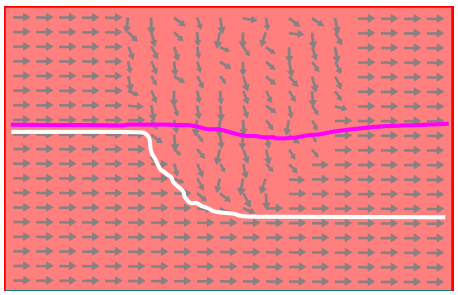


Figure 4.2. Visualization of hyperstreamline (red) and tensorline (yellow) propagation though anisotropic (left, right and bottom) and isotropic regions (top middle) in synthetic data. The directions of the major eigenvectors are indicated with arrows. Note that the tensorline continues along its present course, despite encountering a region of isotropic diffusion.

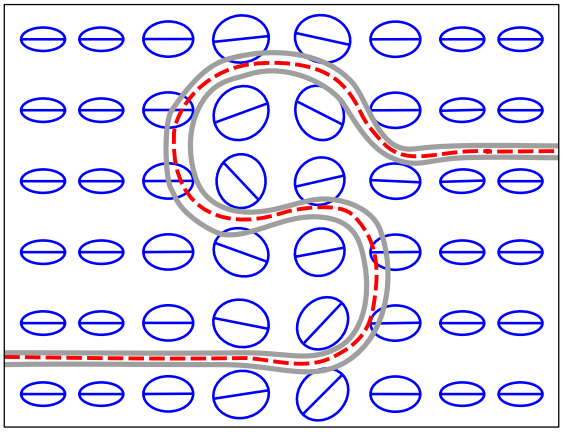


Figure 4.3. Hyperstreamline advection through a nearly isotropic region. The S-shaped path is an artifact of the noise in the region and conveys misleading information about the direction of flow through the field.

the frame is nearly isotropic. The S-shaped path through the middle of the field is not indicative of complex structure in the data, but might be an artifact of measurement noise or partial voluming. However, it is in no way obvious to the viewer that this is the case.

Extending our visualization to three dimensions, we can have confusion in isotropic regions as well as in planar anisotropic regions. In the case of planar anisotropy, the confusion is a result of the field having nearly equivalent first and second eigenvalues and, therefore, major and medium eigenvectors that are only meaningful insofar as they span a particular plane, but not in their particular orientations within that plane. Similarly, for isotropic regions, none of the eigenvector directions are individually meaningful.

4.2 Method

Section 2.2.2 described how diffusion is a probabilistic phenomenon, and how the diffusion tensor models the probability density function (PDF) of where a particle's Brownian motion will move it over time. To visualize this PDF, we cover the unit sphere with dots, as is shown for the unit circle in Figure 4.4. All of the points on the unit sphere (circle) are then transformed by the diffusion tensor, resulting in an ellipsoid (ellipse). Furthermore, the resulting distribution of dots on that ellipsoid (ellipse) corresponds to the probable distribution of particles as they diffuse from the origin. As we can see in Figure 4.4, the dots have a higher density in the $\mathbf{e}_1$ direction, and a lower density in $\mathbf{e}_2$. This corresponds to the higher likelihood that a particle will be diffused in the $\mathbf{e}_1$ direction, and a lower likelihood that it will be diffused in some other direction.

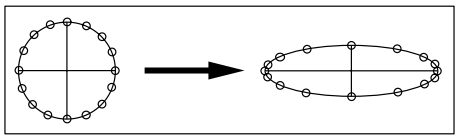


Figure 4.4. Redistribution of uniform sample resulting from anisotropic diffusion tensor. Particles have a higher probability of being diffused in the $\mathbf{e}_1$ (major eigenvector) direction of the ellipsoid.

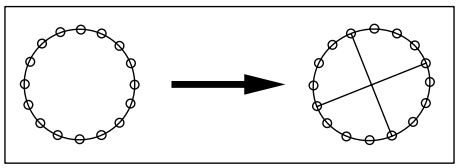


Figure 4.5. Redistribution of uniform sample resulting from nearly isotropic diffusion tensor. Probability of particle diffusion is approximately the same in every direction.

However, if we look at the case in Figure 4.5, the dots on the diffusion ellipsoid (ellipse) are nearly uniformly distributed. This case is representative of any nearly isotropic diffusion tensor. In contrast to the previous example, here there is approximately the same probability for particle diffusion in *any* direction. In such a case, choosing the major eigenvector as the diffusion direction is very much an arbitrary decision. This instability is also depicted in Figure 4.6, where we are examining the effects of applying various tensors to a unit vector. In each row of the image, we apply a different tensor. The matrix is given on the left, then the corresponding ellipse is shown (with the major eigenvector axis indicated in gray), followed by the vector upon which the matrix will operate, and finally the transformed vector is shown on the right. We also indicate the effects of the transformation on the x and y unit vectors, depicted in dashed lines. We note that for the last row, the diffusion tensor is nearly isotropic (as evidenced by the nearly circular ellipse). The vector we are transforming by this matrix is almost orthogonal to

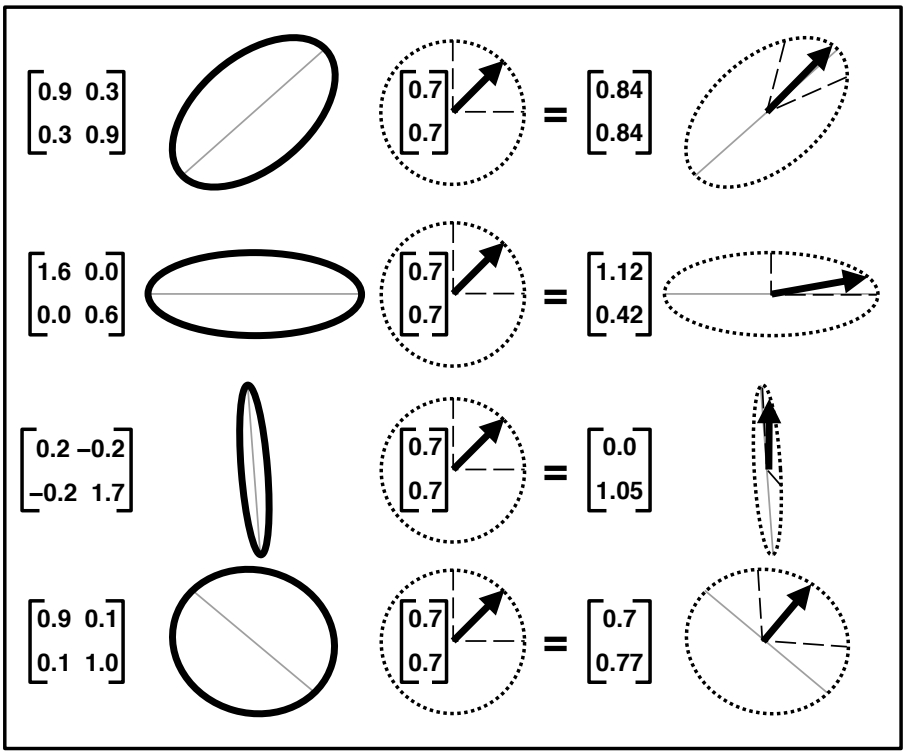


Figure 4.6. Remapping of unit vector through four different tensors. Each row gives the geometric interpretation of applying a different tensor (left matrix and left ellipsoid) to the same example vector (middle vector and middle circle) and the resultant transformed vector (right vector). The gray axis of the ellipses is the direction the major eigenvector and is the direction in which a hyperstreamline would be propagated; in contrast, the dark arrow on the right is the diffusion modulated direction. Our method uses a combination of these two terms to produce more stable propagation paths through isotropic regions.

the first eigenvector. However, since the tensor is nearly isotropic, the output vector on the right is only slightly rotated from its initial position shown in the middle.

If we follow an individual particle's path as it moves through the volume (being probabilistically diffused as it travels), we get a streamline traced through the field. Delmarcelle's method propagates these hyperstreamlines by always choosing to diffuse the particle in the direction of the major eigenvector of the tensor through which it is traveling. Delmarcelle has thus reduced the problem of advecting through a tensor field to the problem of advecting through the $\mathbf{e}_1$ vector field. Although the streamline paths generated with this method are, in fact, the most likely pure diffusion paths, they can at times be misleading, as we saw in Figure 4.3.

Delmarcelle's hyperstreamlines assume a pure diffusion model; however, in regions of the data with planar and spherical anisotropy, the first principal component is a rather arbitrary direction. These ambiguities result in unstable propagation. Our method stabilizes the propagating by incorporating two additional terms to propagate through ambiguous regions. Because these terms supplement the diffusion motion with a velocity term, we refer to them as *advection* vectors, in the spirit of advection-diffusion particle

physics.

The "advection" vector used to stabilize propagation is combination of two vectors, with relative weightings chosen by the user. These vectors correspond to the *incoming* direction $\mathbf{v}_{in}$ (the direction of the previous propagation step), and the *deflection* direction $\mathbf{v}_{\text{def}}$ (the incoming vector, transformed by the tensor). Specifically, we compute:

$$\mathbf{v}_{\text{def}} = \mathbf{D}\mathbf{v}_{\text{in}}, \tag{4.1}$$

where $\mathbf{D}$ is the diffusion tensor. We note that as a preprocess, we scale our diffusion tensor $\mathbf{D}$ by $2/\lambda_{\max}$, where $\lambda_{\max}$ is the largest eigenvalue found within the brain white and gray matter (as described in Section 2.3.3, and threshold of eigenvalue mean μ_1 provides a simple effective test for whether a voxel belongs to brain tissue or highly diffusive cerebrospinal fluid). This scaling has the effect of normalizing the diffusion term to be to more appropriately scaled with respect to the advection terms.

The propagation direction used for path integration is a linear combination of $\mathbf{e}_1$, $\mathbf{v}_{\text{in}}$ and $\mathbf{v}_{\text{def}}$. Because $\mathbf{e}_1$ and $-\mathbf{e}_1$ are both eigenvectors, we can avoid "doubling back" on ourselves by (at each step of the integration) negating $\mathbf{e}_1$ if $\mathbf{v}_{\text{in}} \cdot \mathbf{e}_1 < 0$. The way in which these vectors are combined to produce the next propagation step vector, $\mathbf{v}_{\text{out}}$, depends on the shape of the local tensor. We use the c_l anisotropy metric (Section 2.3.4) to assess the numerical confidence in the computed value of $\mathbf{e}_1$, to determine the relative contribution of $\mathbf{e}_1$:

$$\mathbf{v}_{\text{out}} = c_l\mathbf{e}_1 + (1 - c_l)((1 - w_{\text{punct}})\mathbf{v}_{\text{in}} + w_{\text{punct}}\mathbf{v}_{\text{def}}), \tag{4.2}$$

A user-controlled parameter w_{punct} determines the relative contribution of $\mathbf{v}_{\text{in}}$ and $\mathbf{v}_{\text{def}}$. Equation 4.2 is designed to satisfy the constraints listed in in Table 4.2. The first and last rows of this table are straightforward. The second and third rows describe what should happen in regions of planar anisotropy. That is, these two cases cannot be disambiguated based on anisotropy type alone, so we resort to a user-controlled parameter w_{punct}. This coefficient can take on values from 0 to 1, and affects how much the propagation should be encouraged to "puncture" through planar tensors oriented normal to its path, versus turning into the plane spanned by the directions of fastest diffusivity. This property depends largely on the type of data being investigated, which is why it has been left as a user-definable coefficient. For example, when identifying white matter association tracts, a puncture coefficient of 0.20 worked well in practice, as shown in Figure 4.2. For more rigid datasets, the appropriate coefficient choice would likely by somewhat higher.

To constrain the computed fiber paths to regions which are plausibly white matter, we use a conservative anisotropy threshold such as FA > 0.1 to terminate fibers when they

Table 4.1. Constraints for tensorline propogation.

Anisotropy Shape	**Direction In**	**Desired Out**
Linear	Any	$\mathbf{e}_1$
Planar	Tangential to disk	$\mathbf{v}_{\text{in}}$ or $\mathbf{v}_{\text{def}}$
Planar	Normal to disk plane	$\mathbf{v}_{\text{def}}$
Spherical	Any	$\mathbf{v}_{\text{in}}$ or $\mathbf{v}_{\text{def}}$

are straying into cerebrospinal fluid or isotropic regions. Discretely propagating along the $\mathbf{v}_{\text{out}}$ tensorlines, we generate different paths than we did by simply advecting through the $\mathbf{e}_1$ principal eigenvector field. This is illustrated in the schematic Figure 4.7, where we revisit the case shown in Figure 4.3. Now we have added solid tensorlines, and we note that where the hyperstreamlines wandered through the isotropic region in the middle, the tensorlines continued straight through, with only minor fluctuations.

4.3 Results

In the previous sections, we have shown simple images to illustrate indicate the theoretic efficacy of tensorline propagation. In this section we show a tensorline visualization of actual diffusion tensor MRI data. The dataset is a $128 \times 128 \times 60$ volumetric dataset, with a diffusion tensor matrix at each voxel. The 60 slices extend from the tops of the eyes at the bottom, to the top of the cortical surface at the top.

Figure 4.8 shows an example of how deflection in tensorlines serves to stabilize the direction of the fiber tract. The standard tractography in Figure 4.8(a) gets diverted from an anatomically plausible path by a transient change in the direction of $\mathbf{e}_1$ in voxels with high planar anisotropy (high c_p, low c_l). The tensorlines in Figure 4.8(a) do not have this problem because (from Equation 4.2), a low c_l causes the tensorline to travel in the weighted average of its current direction and the deflection vector $\mathbf{v}_{\text{def}}$, which points along a direction of high diffusivity. The RGB colormap and superquadric glyphs on the nearby cutting plane surface provide the anatomical context to demonstrate that the tensorline path is anatomically plausible, extending upwards to the cortical surface. Although the cutting plane (and the glyphs placed within it) and the tensorlines are at slightly different locations within the dataset, the shape and extent of the corpus callosum and corona radiata in the anterior/poster direction (the green direction of the axis set in the upper right corner) mean that cutting plane indicates the directional properties of the

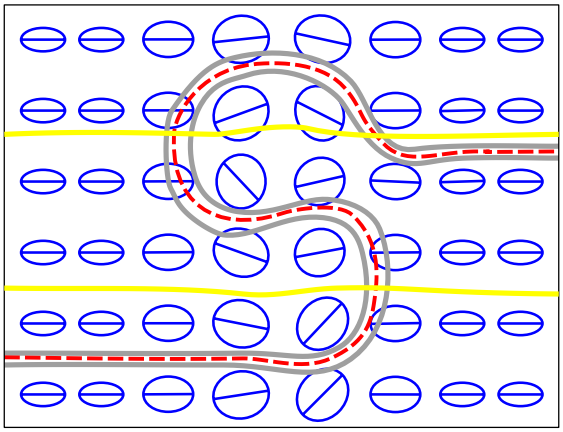


Figure 4.7. Comparison of tensorlines (solid yellow) and hyperstreamline (core is dashed red, border is gray). Note the tensorlines continue with only minor fluctuations through the isotropic region in the center of the figure, whereas the hyperstreamline is diverted into an S-shaped path.

region through which the tensorlines propogate. Figure 4.9 shows additional examples of tensorlines extracting major white matter features.

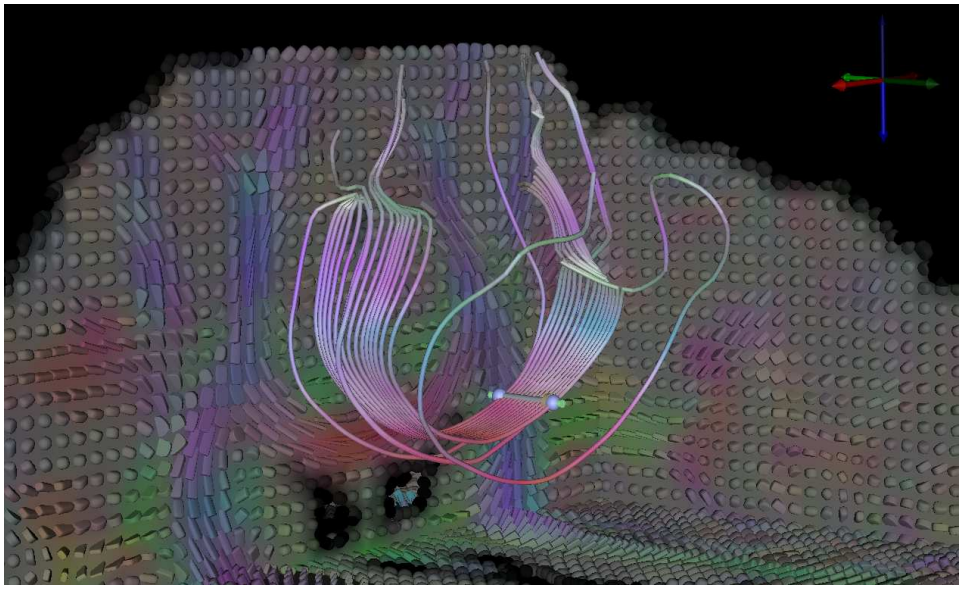
A standard termination criterion for tractography is a *fractional* anisotropy threshold, such as FA = 0.1 or FA = 0.2 [133]. Because a tensor may have purely planar anisotropy ($c_l = 0$) and nonzero FA (see Figure 2.9(a)), this means that $\mathbf{e}_1$-based tractography can become unstable. To compensate, an additional termination criterion is that the curvature of the fiber path should be low [133]. This is a reasonable compensation for an inherently unstable algorithm. A strength of the tensorlines method is that the stability of its paths does not depend on this additional criterion.

4.4 Discussion

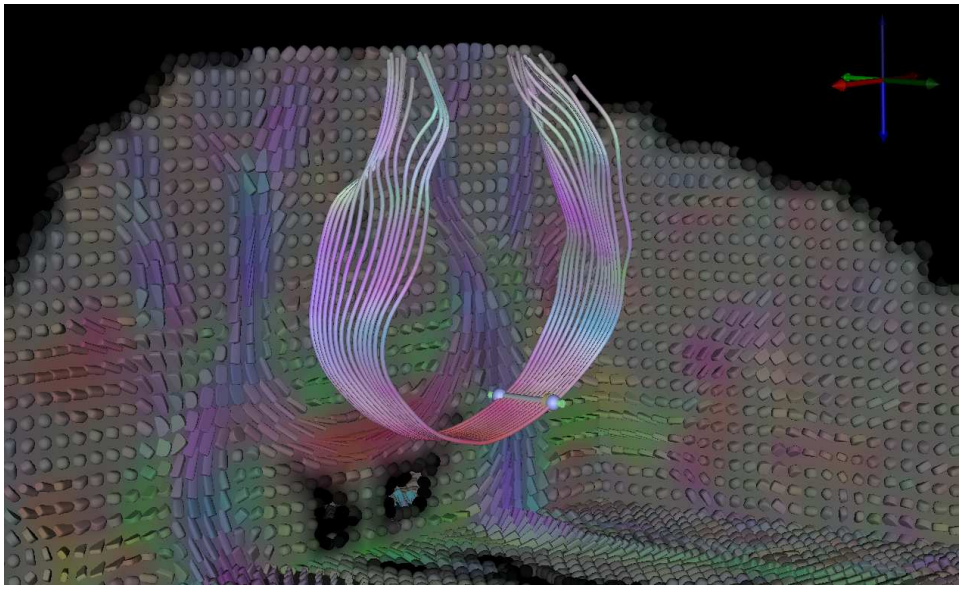
We have introduced a novel fiber tractography method, called tensorlines, for visualizing diffusion tensor fields. Tensorlines extend the traditional tractography methods by stabilizing propagation through regions with planar anisotropy and isotropy. Tensorlines travel along $\mathbf{e}_1$ when it is numerically well-defined (high c_l values), and tend towards directions of high diffusion probability by using *deflection*: multiplication by the diffusion tensor. The free parameters in computing tensorlines include the usual components of path integration (step size, integration method, interpolation method), a termination criterion (such as an anisotropy threshold), and the w_{punct} parameter. This means that besides the user-defined seed point, there are a manageable number of parameters to tune to create a tensorlines visualization.

The success of the tensorlines method is evidenced by Figures 4.8 and 4.9, and should be evaluated primarily relative to other methods published contemporaneously (circa 1999). Aspects of the tensorline method which likely impart greater accuracy include the fact that we interpolate the whole diffusion tensor rather than just the principle eigenvector, and we perform trilinear interpolation to form a continuous tensor field, rather than step from voxel face to voxel face [132, 200]. Subsequent work by Lazar et al. demonstrated the accuracy of tensorlines (in synthetic data) in the presence of measurement noise [113], and has demonstrated that a deflection-based method accurately extracts the major white matter fiber tracts [114]. More recent state-of-the-art tractography work includes the approach of Zhukov et al. [206], which regularizes the path by a local spline fit to the tensor values, and the approach of Zhang et al. [204, 203] which includes "stream surfaces" advected in regions of planar anisotropy to explicitly visualize the neighborhoods where tractography will be problematic. Ultimately, though, the correctness of tractography as an indicator of axonal connectivity will have to be verified by means of a careful histology [133].

In the future, we would like to encode additional information into the tensorline surface, in a way similar to Delmarcelle's method for surfaces about hyperstreamlines [51]. We could encode the deviation from $\mathbf{e}_1$ as a texture on the surface, providing the user with a visual clue as to the difference between the various vectors being weighted in Equation 4.2. Another way to enrich the visualization created by tensorlines is to place superquadric glyphs along the path. This represents a different strategy for choosing the set of discrete locations at which to display all the tensor information. Another simple improvement would be to have the cross-section of the tensorline vary as a superellipse: like with hyperstreamlines, the cross-section could encode the additional eigenvectors and eigenvalues, but a superellipsoidal cross-section may confer the same benefits of visual clarity that the superquadrics offered over ellipsoids. The placement of the seed

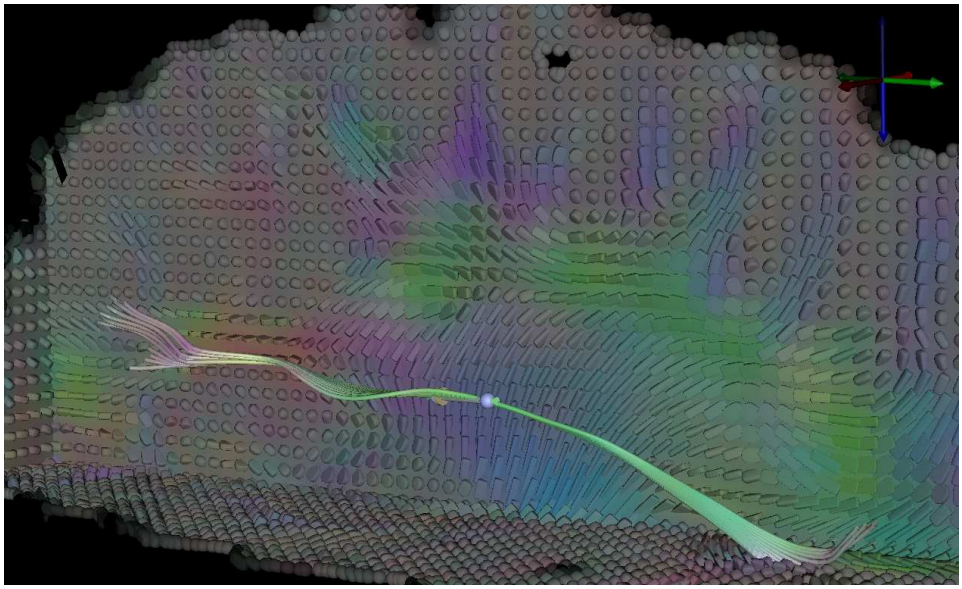


(a) $\mathbf{e}_1$-based tractography

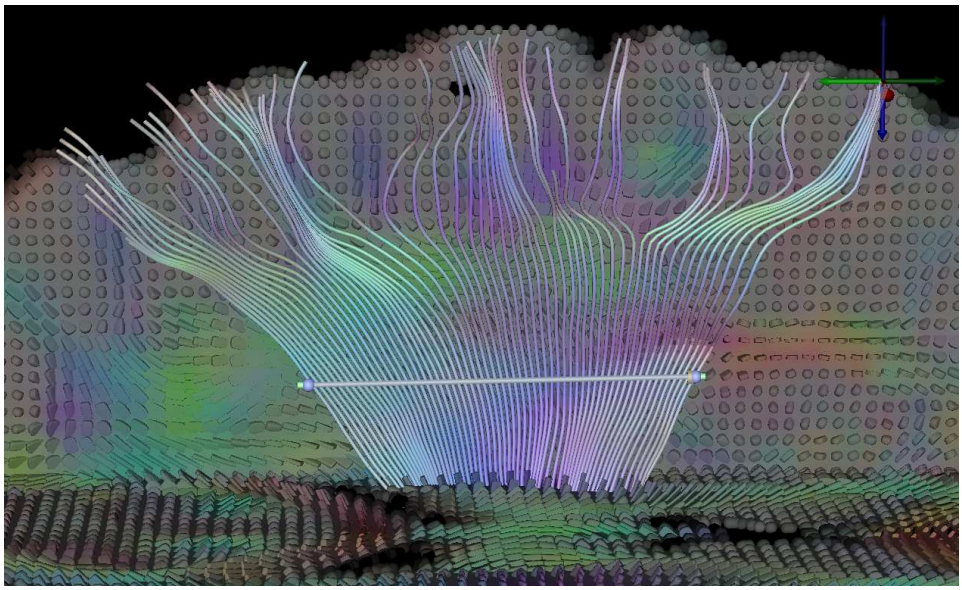


(b) Tensorlines

Figure 4.8. Comparison of behavior of standard $\mathbf{e}_1$-based tractography (a) and tensorlines (b) in the corpus callosum and corona radiata. At voxels with a significant planar component, the direction of $\mathbf{e}_1$ can change suddenly, while tensorlines continue on an anatomically plausible path.



(a) Superior longitudinal (arcuate) fasciculus



(b) Internal capsule and corona radiata

Figure 4.9. Examples of major white matter tracts visualized by tensorlines. Note superquadric glyphs populating the cutting plane surfaces.

points could be automated to some extent, perhaps by looking for centroids of spots in a reaction-diffusion texture.

We are also interested in stabilizing advection by propagating groups of streamlines together as a cohesive bundle [191]. Our earlier work in this area focused on bundles which advected through flow fields. In the future, we would like to extend these ideas to also apply to tensor fields.

CHAPTER 5

INVARIANT GRADIENTS AND ROTATION TANGENTS

The goal of this chapter is to develop a mathematical and computational framework for decomposing the gradient (the first derivative) of the tensor field by its projection onto shape and orientation changes. In this context, we will use "change" to refer to the infinitesimal differences in tensor value associated with the gradient, which may be differentiated with respect to either the space of tensor values, Sym_3, or the spatial domain of the tensor field $\mathbb{R}^3$, depending on context. We will make heavy use of the fact that the space of symmetric tensor values is a vector space (Section 2.1.5). The tensor-valued gradients of invariants (Section 2.3.2) generate a basis for changes in shape, and the tangents to the group action ψ (Section 2.1.7) will generate a basis for changes in orientation. Figure 5.1 provides a schematic view. The mathematical notation will be introduced in this paragraph, and fully defined in the following sections. Around a given tensor value $\mathbf{D}$, there are three mutually orthogonal directions ($\widehat{\boldsymbol{\nabla}}\mu_1$, $\widehat{\boldsymbol{\nabla}}\mu_2$, and $\widehat{\boldsymbol{\nabla}}\alpha_3$) along which the tensor *shape* changes. These are computed from the tensor-valued gradients of tensor invariants. Orthogonal to these directions, and also mutually orthogonal, are three directions ($\widehat{\boldsymbol{\Phi}}_1$, $\widehat{\boldsymbol{\Phi}}_2$, and $\widehat{\boldsymbol{\Phi}}_3$) along which the tensor *orientation* changes. These are all tangents to the orbit $SO_3(\mathbf{D})$ of the group action ψ. Together, these six tensor-valued directions form an *orthonormal* basis for the vector space of symmetric tensors, and therefore any change in tensor value can be analyzed by projections onto the subspaces of shape and orientation change. The following sections develop these concepts

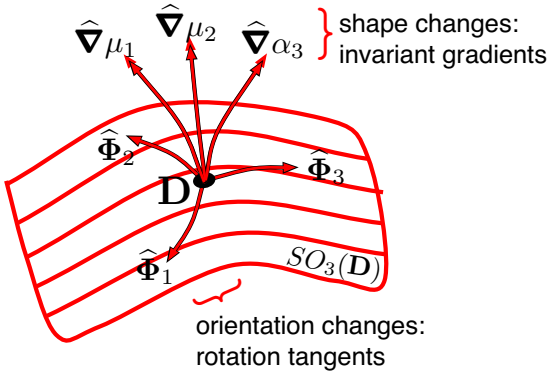


Figure 5.1. Schematic view of shape and orientation changes around a given tensor value $\mathbf{D}$. The orbit $SO_3(\mathbf{D})$ contains orientation changes, and shape changes are all orthogonal to the orbit.

in greater detail.

This chapter ties together and builds on previous work from three separate areas: diffusion tensor imaging, continuum mechanics, and image processing. Previous work by Pajevic et al. [140] in developing a continuous tensor field from a discretely sampled volume included a description of the tensor gradient, a third order tensor, and a means of decomposing the tensor gradient into two parts: the isotropic component, and everything else, called the deviator. Our decomposition of the gradient of a tensor field is more detailed, accounting for three variables of shape and three of orientation. The three variables of tensor shape are the eigenvalue statistics (mean, variance, and skewness), defined and described in Section 2.3.2. Bahn used the same variables to describe diffusion tensor shape [5]. In the context of continuum mechanics, the tensor-valued *derivatives* of essentially the same three shape variables were employed by Criscione et al. to characterize strain tensors [47]. The differences are that we express shape in terms eigenvalues rather than the *logarithms* of the eigenvalues, and we take special care to construct normalized (unit-magnitude) invariant gradients onto which the spatial gradient of the tensor field is projected. For characterizing changes in tensor orientation, we use $d\psi$, the derivative of the group action ψ defined in Section 2.1.7, to define rotation tangents. In the context of image processing, previous work by Damon [48] used $d\psi$ as part of creating a topological description of Sym_3, the space of three-dimensional symmetric tensors, to build a framework for analyzing the scale-space of two-dimensional images. As with the invariant gradients, our approach here is distinguished by the normalization of the rotation tangents, which in turn enables the definition of a novel orthonormal basis for tensor change around each tensor value. By having an orthonormal basis, the tensor field gradient can be projected in a way which uniformly preserves the relevant gradient magnitudes, around all tensor values.

The structure of this chapter is as follows. First, we describe an orthonormal basis for shape changes, generated by the tensor-valued gradients of tensor invariants (Section 5.1), and then describe a means of ensuring that the basis is defined at all tensor values, including those tensors for which two or more eigenvalues are equal (Section 5.2). Next, we describe how to measure the spatial gradient of a tensor field against this basis for shape change, allowing us to locate boundaries and features of diffusion tensor images in terms of tensor shape (Section 5.3). To characterize changes in tensor orientation, we first review some mathematics associated with derivatives of manifold-valued functions (Section 5.4), and then compute a general expression for rotation tangents based on the derivative of the group action ψ (Section 5.5). From this, we make some qualitative and quantitative observations about the space of tensor orientation (Section 5.6). Next, we determine an orthonormal basis for shape change, and note that the normalized invariant gradients and rotation tangents span all the symmetric tensors (Section 5.7). Finally, we describe how to measure the gradient of the tensor field against the basis of orientation change (Section 5.8).

The primary motivation for this chapter is to create a basis for generalizing basic image processing operations like edge-detection to diffusion tensor images in a manner that facilitates extraction of anatomically significant boundaries. Based on the observations about the relationship between eigenvalue moments (μ_1, μ_2, α_3) and tissue structure given in Section 2.3.3, the basis onto which the tensor gradient is projected will created from the same three degrees of freedom in tensor shape. This allows much greater specificity in responding to different kinds of shape changes than is possible with previous approaches.

The secondary motivation of this chapter is to avoid computing the eigenvectors of a

tensor (as described in Section 3.1.1) whenever possible. Parameterizing shape changes in terms of individual eigenvalues requires knowledge of the tensor's eigenvectors, while using moments of eigenvalues typically requires neither eigenvalues nor eigenvectors. Notably, we show in Section 5.8 that the overall magnitude of orientation change can be efficiently computed without knowing the tensor eigenvectors, even though tensor orientation is entirely determined by the eigenvectors.

5.1 Orthonormal Invariant Gradients

If ρ is a scalar-valued function over $\mathbb{R}^3$, its gradient $\nabla\rho$ is a vector-valued function mapping from $\mathbb{R}^3$ to $\mathbb{R}^3$. Analogously, a tensor invariant is a scalar-valued function J over a vector space (of symmetric tensor values) Sym_3, so its gradient maps from Sym_3 to Sym_3. We notate the gradient of a tensor invariant J as $\boldsymbol{\nabla} J$, preferring this over the conventional notations "grad J" and "$\partial J/\partial \mathbf{D}$" for its brevity. It is important to keep in mind that the domain and range of $\boldsymbol{\nabla} J$ are not $\mathbb{R}^3$ (the spatial domain over which the tensor field is defined), but are Sym_3 (the six-dimensional space of symmetric tensor values). The boldface "$\boldsymbol{\nabla}$" (in contrast to "∇") is a reminder of this. The gradient $\boldsymbol{\nabla} J$ of invariant J is defined as:

$$\begin{aligned} \boldsymbol{\nabla} J : Sym_3 &\mapsto Sym_3 \\ \boldsymbol{\nabla} J(\mathbf{D}_0) &\equiv \left.\frac{\partial J}{\partial \mathbf{D}}\right|_{\mathbf{D}=\mathbf{D}_0} . \end{aligned}$$

Recall that the spectral decomposition (Equation 2.13)

$$\mathbf{D} = \lambda_1(\mathbf{e}_1 \otimes \mathbf{e}_1) + \lambda_2(\mathbf{e}_2 \otimes \mathbf{e}_2) + \lambda_3(\mathbf{e}_3 \otimes \mathbf{e}_3) ,$$

expresses a tensor value in terms of its eigenvalues λ_i and eigenvectors $\mathbf{e}_i$. This does not require or assume that the eigenvalues are sorted. By Equation 2.15, the component tensors $\mathbf{e}_i \otimes \mathbf{e}_i$ are mutually orthogonal:

$$\mathbf{e}_i \otimes \mathbf{e}_i : \mathbf{e}_j \otimes \mathbf{e}_j = \delta_{ij}\delta_{ij} = \delta_{ij} . \tag{5.1}$$

Recalling that the gradient of a function gives the direction along which the function increases fastest, it is clear from the spectral decomposition that adding to $\mathbf{D}$ some fraction of $(\mathbf{e}_1 \otimes \mathbf{e}_1)$ will increase the value of eigenvalue λ_1, and, by Equation 5.1, it will not change the value of λ_2 or λ_3. In other words:

$$\boldsymbol{\nabla}\lambda_i = \mathbf{e}_i \otimes \mathbf{e}_i \tag{5.2}$$

Equation 5.1 also shows that $\boldsymbol{\nabla}\lambda_i$ has unit norm:

$$\|\boldsymbol{\nabla}\lambda_i\| = \sqrt{\boldsymbol{\nabla}\lambda_i : \boldsymbol{\nabla}\lambda_i} = \sqrt{\delta_{ii}} = 1 . \tag{5.3}$$

Thus $\{\boldsymbol{\nabla}\lambda_1, \boldsymbol{\nabla}\lambda_2, \boldsymbol{\nabla}\lambda_3\}$ is a local orthonormal basis for the vector space of shape change around tensor $\mathbf{D}$. Note that because this basis always has exactly three elements, the space of shape change is always three-dimensional, regardless of tensor value.

This basis has two drawbacks, however. First is that its use requires the computation of the tensor eigenvalues and eigenvectors. Second is the weaker anatomical relevance of the individual eigenvalues, as compared to the eigenvalue moments (μ_1, μ_2, α_3) which have

a close connection to the definition of anatomical features in a tensor field: Section 2.3.3 detailed the relationship between μ_1, μ_2, α_3 and structures associated with cerebro-spinal fluid, gray matter, white matter, fiber tracts and their crossing, and the consequences of partial volume.

Recall from Section 2.3.1 that the J_i invariants are defined as:

$$\begin{aligned} J_1(\mathbf{D}) &= \mathrm{tr}(\mathbf{D}) \\ J_2(\mathbf{D}) &= (\mathrm{tr}(\mathbf{D})^2 - \mathrm{tr}(\mathbf{D}^2))/2 \\ J_3(\mathbf{D}) &= \det(\mathbf{D}) \\ J_4(\mathbf{D}) &= \|\mathbf{D}\|^2 \, . \end{aligned}$$

Tensor analysis permits coordinate-free expressions of their gradients as [86]:

$$\begin{aligned} \boldsymbol{\nabla} J_1(\mathbf{D}) &= \mathbf{I} \\ \boldsymbol{\nabla} J_2(\mathbf{D}) &= \mathrm{tr}(\mathbf{D})\mathbf{I} - \mathbf{D} \\ \boldsymbol{\nabla} J_3(\mathbf{D}) &= \det(\mathbf{D})\mathbf{D}^{-1} \\ \boldsymbol{\nabla} J_4(\mathbf{D}) &= 2\mathbf{D} \, . \end{aligned} \tag{5.4}$$

Appendix D explains how these expressions are derived from the Taylor expansion of $J_i(\mathbf{D} + \boldsymbol{\epsilon})$. Note that $\boldsymbol{\nabla} J_1(\mathbf{D}) = \mathbf{I}$ is the only invariant gradient that is constant. All the other gradients vary as the function of the tensor value they are evaluated at.

Equation 5.4 makes it straight-forward to compute the gradients of the J_i invariants for a given tensor $\mathbf{D}$, in the laboratory frame. For example,

$$[\boldsymbol{\nabla} J_2(\mathbf{D})]_{\mathcal{L}} = [\mathrm{tr}(\mathbf{D})\mathbf{I} - \mathbf{D}]_{\mathcal{L}} = \begin{bmatrix} D_{yy} + D_{zz} & -D_{xy} & -D_{xz} \\ & D_{xx} + D_{zz} & -D_{yz} \\ (Sym) & & D_{yy} + D_{zz} \end{bmatrix} .$$

The gradient of the determinant $\boldsymbol{\nabla} J_3(\mathbf{D})$ appears unwieldy by its use of the tensor inverse, however for 3×3 matrices we know from Cramer's rule that $\det(\mathbf{D})\mathbf{D}^{-1}$ is just the matrix of cofactors (determinants of 2×2 matrix minors). Thus:

$$[\boldsymbol{\nabla} J_3(\mathbf{D})]_{\mathcal{L}} = \begin{bmatrix} D_{yy}D_{zz} - D_{yz}^2 & D_{xz}D_{yz} - D_{xy}D_{zz} & D_{xy}D_{yz} - D_{xz}D_{yy} \\ & D_{xx}D_{zz} - D_{xz}^2 & D_{xy}D_{xz} - D_{yz}D_{xx} \\ (Sym) & & D_{xx}D_{yy} - D_{xy}^2 \end{bmatrix} .$$

Because the goal is to create a local orthonormal basis for shape change from the eigenvalue moments, we have to know how to evaluate the (tensor) inner product between any two invariant gradients. The chain rule allows us to convert the spectral decomposition of a tensor into a spectral decomposition of an invariant gradient:

$$\boldsymbol{\nabla} J = \frac{\partial J}{\partial \mathbf{D}} = \sum_i \frac{\partial J}{\partial \lambda_i} \frac{\partial \lambda_i}{\partial \mathbf{D}} = \sum_i \frac{\partial J}{\partial \lambda_i} \boldsymbol{\nabla} \lambda_i = \sum_i \frac{\partial J}{\partial \lambda_i} (\mathbf{e}_i \otimes \mathbf{e}_i) \, . \tag{5.5}$$

The utility of this decomposition is in reducing the inner product (the double contraction) between $\boldsymbol{\nabla} J$ and $\boldsymbol{\nabla} K$ into a standard vector dot product between $\nabla_\lambda J$ and $\nabla_\lambda K$. Recall

from Section 2.3.2 that $\nabla_\lambda J$ is the gradient of invariant J viewed as a function over the space of $(\lambda_1, \lambda_2, \lambda_3)$ triples:

$$\begin{aligned}
\boldsymbol{\nabla} J : \boldsymbol{\nabla} K &= \left(\sum_i \frac{\partial J}{\partial \lambda_i} (\mathbf{e}_i \otimes \mathbf{e}_i) \right) : \left(\sum_j \frac{\partial K}{\partial \lambda_j} (\mathbf{e}_j \otimes \mathbf{e}_j) \right) \\
&= \sum_{ij} \frac{\partial J}{\partial \lambda_i} \frac{\partial K}{\partial \lambda_j} (\mathbf{e}_i \otimes \mathbf{e}_i) : (\mathbf{e}_j \otimes \mathbf{e}_j) \\
&= \sum_{ij} \frac{\partial J}{\partial \lambda_i} \frac{\partial K}{\partial \lambda_j} \delta_{ij} \\
&= \sum_i \frac{\partial J}{\partial \lambda_i} \frac{\partial K}{\partial \lambda_i} \\
&= \nabla_\lambda J \cdot \nabla_\lambda K \; . \qquad (5.6)
\end{aligned}$$

Appendix D demonstrates that the gradients of the eigenvalue statistics μ_1, μ_2, and α_3 are orthogonal over the space of eigenvalue triples:

$$\begin{aligned}
\nabla_\lambda \mu_1 \cdot \nabla_\lambda \mu_2 &= 0 \\
\nabla_\lambda \mu_2 \cdot \nabla_\lambda \alpha_3 &= 0 \\
\nabla_\lambda \mu_2 \cdot \nabla_\lambda \alpha_3 &= 0 \; .
\end{aligned}$$

Then Equation 5.6 establishes that

$$\begin{aligned}
\boldsymbol{\nabla} \mu_1 : \boldsymbol{\nabla} \mu_2 &= 0 \\
\boldsymbol{\nabla} \mu_2 : \boldsymbol{\nabla} \alpha_3 &= 0 \qquad (5.7) \\
\boldsymbol{\nabla} \mu_2 : \boldsymbol{\nabla} \alpha_3 &= 0 \; .
\end{aligned}$$

That is, $\{\boldsymbol{\nabla}\mu_1, \boldsymbol{\nabla}\mu_2, \boldsymbol{\nabla}\alpha_3\}$ is an orthogonal set of gradients for all tensor values. Aside from the minor difference between describing the statistics of eigenvalues versus statistics of logarithms of eigenvalues, these invariant gradients are essentially the same as the tensor-valued gradients of the K_1, K_2, K_3 response functions defined by Criscione et al. [47]. In the following, we first describe how to efficiently compute these gradients, and then we address how to normalize their magnitude, in order to produce the desired orthonormal basis of tensor shape change.

Knowing the definitions for μ_1 and μ_2 in terms of J_i (Equations 2.30, 2.31, 2.26), we can express $\boldsymbol{\nabla}\mu_1$ and $\boldsymbol{\nabla}\mu_2$ in terms of $\boldsymbol{\nabla} J_i$ as:

$$\mu_1 = J_1/3 \quad \Rightarrow \quad \boldsymbol{\nabla}\mu_1 = \boldsymbol{\nabla} J_1/3 \qquad (5.8)$$

$$\mu_2 = 2Q = 2(J_1^2 - 3J_2)/9 \quad \Rightarrow \quad \boldsymbol{\nabla}\mu_2 = 2(2J_1 \boldsymbol{\nabla} J_1 - 3\boldsymbol{\nabla} J_2)/9 \qquad (5.9)$$

Dividing the gradient of a invariant by its norm forms a unit-length tensor, provided that the gradient is nonzero. We notate such a normalized tensor-valued gradient by $\widehat{\boldsymbol{\nabla}}$.

$$\begin{aligned}
\widehat{\boldsymbol{\nabla}} J : Sym_3 &\mapsto Sym_3 \\
\widehat{\boldsymbol{\nabla}} J &\equiv \boldsymbol{\nabla} J / \| \boldsymbol{\nabla} J \| \qquad (5.10)
\end{aligned}$$

The gradient of the eigenvalue mean is constant: $\widehat{\boldsymbol{\nabla}}\mu_1 = \mathbf{I}/\sqrt{3}$. The variance of the eigenvalues μ_2 is nonnegative. When μ_2 attains its minimum value of 0, then $\boldsymbol{\nabla}\mu_2 = 0$, in which case $\widehat{\boldsymbol{\nabla}}\mu_2$ is undefined. In contrast, by Equation 5.3:

$$\widehat{\boldsymbol{\nabla}}\lambda_i = \boldsymbol{\nabla}\lambda_i \; , \qquad (5.11)$$

so all three eigenvalue gradients $\widehat{\boldsymbol{\nabla}}\lambda_i$ are always defined, though their values depend on the eigenvectors.

Given $\widehat{\boldsymbol{\nabla}}\mu_1$ and $\widehat{\boldsymbol{\nabla}}\mu_2$ (where defined), one could complete the local basis of shape variation by directly calculating $\widehat{\boldsymbol{\nabla}}\alpha_3$, but this is a rather complicated expression. Instead, we approach $\widehat{\boldsymbol{\nabla}}\alpha_3$ by taking the gradient of the determinant, $\boldsymbol{\nabla}J_3$, and subtract out the components parallel to $\widehat{\boldsymbol{\nabla}}\mu_1$ and $\widehat{\boldsymbol{\nabla}}\mu_2$:

$$\mathbf{G} = \boldsymbol{\nabla}J_3 - (\boldsymbol{\nabla}J_3 : \widehat{\boldsymbol{\nabla}}\mu_1)\widehat{\boldsymbol{\nabla}}\mu_1 - (\boldsymbol{\nabla}J_3 : \widehat{\boldsymbol{\nabla}}\mu_2)\widehat{\boldsymbol{\nabla}}\mu_2 \; . \tag{5.12}$$

When $\mathbf{G}$ is nonzero, it must be parallel to $\widehat{\boldsymbol{\nabla}}\alpha_3$, by the orthogonality of $\boldsymbol{\nabla}\mu_1$, $\boldsymbol{\nabla}\mu_2$, and $\boldsymbol{\nabla}\alpha_3$, and the fact that they span the three-dimensional space of shape changes. On the other hand, when $\mathbf{G}$ is zero, there must be linear dependence among $\boldsymbol{\nabla}J_3$, $\boldsymbol{\nabla}\mu_1$, and $\boldsymbol{\nabla}\mu_2$. By Equation 5.6, we can examine this condition in the simpler space of $(\lambda_1, \lambda_2, \lambda_3)$ triples, using the invariants expressed in terms of eigenvalues (Equation 2.25):

$$\begin{aligned}
\boldsymbol{\nabla}J_3 &= \gamma\boldsymbol{\nabla}\mu_1 + \beta\boldsymbol{\nabla}\mu_2 \\
\Rightarrow \nabla_\lambda J_3 &= \gamma\nabla_\lambda\mu_1 + \beta\nabla_\lambda\mu_2 \\
\Rightarrow \begin{bmatrix} \lambda_2\lambda_3 \\ \lambda_1\lambda_3 \\ \lambda_1\lambda_2 \end{bmatrix} &= \gamma \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \beta \begin{bmatrix} \lambda_2+\lambda_3 \\ \lambda_1+\lambda_3 \\ \lambda_1+\lambda_2 \end{bmatrix} \\
& \qquad \lambda_2\lambda_3 - \beta\lambda_2 - \beta\lambda_3 = \gamma \\
& \Rightarrow \quad \lambda_1\lambda_3 - \beta\lambda_1 - \beta\lambda_3 = \gamma \\
& \qquad \lambda_1\lambda_2 - \beta\lambda_1 - \beta\lambda_2 = \gamma \\
& \qquad (\lambda_2-\lambda_3)(\lambda_1-\beta) = 0 \\
& \Rightarrow \quad (\lambda_1-\lambda_3)(\lambda_2-\beta) = 0 \\
& \qquad (\lambda_1-\lambda_2)(\lambda_3-\beta) = 0 \; .
\end{aligned}$$

The last set of equations is formed by pair-wise differences of the previous set. The last set of equations is satisfied if $\lambda_1 = \lambda_2 = \lambda_3 = \beta$, in which case $\mu_2 = 0$ and α_3 itself is undefined. If the eigenvalues are not all equal, then without loss of generality let $\lambda_1 \neq \beta$, which implies $\lambda_2 = \lambda_3 = \beta$. As seen in Section 2.3.2, this means that eigenvalues skewness α_3 is at an extremum. This implies that $\mathbf{G}$ vanishes when α_3 is undefined or at an extremum, which are precisely the conditions under which $\boldsymbol{\nabla}\alpha_3$ is undefined or vanishes. Thus, without loss of generality we can define:

$$\widehat{\boldsymbol{\nabla}}\alpha_3 = \mathbf{G}/\|\mathbf{G}\| \; . \tag{5.13}$$

Together with $\widehat{\boldsymbol{\nabla}}\mu_1$ and $\widehat{\boldsymbol{\nabla}}\mu_2$ (Equations 5.8, 5.9, 5.10) this completes an orthonormal basis for shape change based on the eigenvalue moments, defined at those tensors with three distinct eigenvalues.

The connection between eigenvalue statistics $\{\mu_1, \mu_2, \alpha_3\}$ and biological features, described in Section 2.3.3, carries over to the gradients $\{\widehat{\boldsymbol{\nabla}}\mu_1, \widehat{\boldsymbol{\nabla}}\mu_2, \widehat{\boldsymbol{\nabla}}\alpha_3\}$. The eigenvalue mean gradient $\widehat{\boldsymbol{\nabla}}\mu_1$ signifies the direction (in the space of tensor values) along which tensors get bigger and bulk mean diffusivity increases, as with a transition into cerebrospinal fluid. The variance gradient $\widehat{\boldsymbol{\nabla}}\mu_2$ indicates the direction along which tensors become less spherical and anisotropy increases, as with a transition from gray to white matter. Finally, the skewness gradient $\widehat{\boldsymbol{\nabla}}\alpha_3$ is the direction along which the shape of anisotropy changes from planar to linear, as from a decrease in the voxel-level mixing of heterogeneous fiber directions.

5.2 Extended Definition of $\{\widehat{\boldsymbol{\nabla}}\mu_1, \widehat{\boldsymbol{\nabla}}\mu_2, \widehat{\boldsymbol{\nabla}}\alpha_3\}$

The significant drawback to using $\{\widehat{\boldsymbol{\nabla}}\mu_1, \widehat{\boldsymbol{\nabla}}\mu_2, \widehat{\boldsymbol{\nabla}}\alpha_3\}$ is that $\widehat{\boldsymbol{\nabla}}\mu_2$ and $\widehat{\boldsymbol{\nabla}}\alpha_3$ are not always defined. Unlike the eigenvalues themselves, the eigenvalue variance μ_2 and skewness α_3 are bounded, so their gradients can vanish, and the normalized gradients become undefined. The failure of the shape gradients to consistently span the three-dimensional space of shape changes can be explained in terms of permutation symmetries in the set of eigenvalues. When exactly two eigenvalues are equal, say $\lambda_1 > \lambda_2 = \lambda_3$, no invariant gradient can point in a "direction" which separates the values of λ_2 and λ_3, since this would violate the symmetry of the invariants with respect to permutation of eigenvalues. Invariants must exhibit this permutation symmetry because, when evaluated on diagonal matrices, the invariant is by definition fixed with respect to the changes of basis which permute the axes of the coordinate system in which the tensor is expressed.

Our current approach to "fixing" the shape gradients is to revert to a modification of the $\{\widehat{\boldsymbol{\nabla}}\lambda_1, \widehat{\boldsymbol{\nabla}}\lambda_2, \widehat{\boldsymbol{\nabla}}\lambda_3\}$ basis (as defined below) whenever the $\{\widehat{\boldsymbol{\nabla}}\mu_1, \widehat{\boldsymbol{\nabla}}\mu_2, \widehat{\boldsymbol{\nabla}}\alpha_3\}$ basis is not completely defined. Using $\widehat{\boldsymbol{\nabla}}\lambda_i$ requires tensor diagonalization, which entails more computational expense, so we use the values of $\|\mathbf{G}\|$ (Equation 5.12) and $\|\boldsymbol{\nabla}\mu_2\|$ to determine exactly when diagonalization is needed. If $\|\boldsymbol{\nabla}\mu_2\| = 0$, then all the eigenvalues are equal, and the tensor is already diagonal. When $\|\boldsymbol{\nabla}\mu_2\| > 0$ but $\|\mathbf{G}\| = 0$, then exactly two eigenvalues are equal, and diagonalization is required. If $\|\boldsymbol{\nabla}\mu_2\| > 0$ and $\|\mathbf{G}\| > 0$, then the $\{\widehat{\boldsymbol{\nabla}}\mu_1, \widehat{\boldsymbol{\nabla}}\mu_2, \widehat{\boldsymbol{\nabla}}\alpha_3\}$ basis is completely defined as is. By slightly abusing our previous notation, we give here extended definitions for $\widehat{\boldsymbol{\nabla}}\mu_2$ and $\widehat{\boldsymbol{\nabla}}\alpha_3$ which handle the locations where $\|\boldsymbol{\nabla}\mu_2\| = 0$ or $\|\mathbf{G}\| = 0$:

$$\widehat{\boldsymbol{\nabla}}\mu_2 = \begin{cases} \boldsymbol{\nabla}\mu_2/\|\boldsymbol{\nabla}\mu_2\| & \text{if } \|\boldsymbol{\nabla}\mu_2\| > 0 \quad \text{(a)} \\ \begin{matrix} \sqrt{2/3}(\mathbf{b}_1 \otimes \mathbf{b}_1) \\ -\sqrt{1/6}(\mathbf{b}_2 \otimes \mathbf{b}_2) \\ -\sqrt{1/6}(\mathbf{b}_3 \otimes \mathbf{b}_3) \end{matrix} & \text{otherwise} \quad \text{(b)} \end{cases} \tag{5.14}$$

$$\widehat{\boldsymbol{\nabla}}\alpha_3 = \begin{cases} \mathbf{G}/\|\mathbf{G}\| & \text{if } \|\mathbf{G}\| > 0 \quad \text{(a)} \\ \begin{matrix} \sqrt{1/2}(\mathbf{e}_2 \otimes \mathbf{e}_2) \\ -\sqrt{1/2}(\mathbf{e}_3 \otimes \mathbf{e}_3) \end{matrix} & \text{if } \|\mathbf{G}\| = 0 \text{ and } \lambda_1 > \lambda_2 = \lambda_3 \quad \text{(b)} \\ \begin{matrix} \sqrt{1/2}(\mathbf{e}_1 \otimes \mathbf{e}_1) \\ -\sqrt{1/2}(\mathbf{e}_2 \otimes \mathbf{e}_2) \end{matrix} & \text{if } \|\mathbf{G}\| = 0 \text{ and } \lambda_1 = \lambda_2 > \lambda_3 \quad \text{(c)} \\ \begin{matrix} \sqrt{1/2}(\mathbf{b}_2 \otimes \mathbf{b}_2) \\ -\sqrt{1/2}(\mathbf{b}_3 \otimes \mathbf{b}_3) \end{matrix} & \text{otherwise } (\|\boldsymbol{\nabla}\mu_2\| = 0) \quad \text{(d) .} \end{cases} \tag{5.15}$$

Equations 5.15(b) and 5.15(c) involve arbitrary decisions about how to separate the two equal eigenvalues. The arbitrariness of this decision is effectively imposed by the numerical properties of whatever procedure is used to pick two orthogonal eigenvectors to span the two-dimensional eigenspace associated with the double eigenvalue. The differential changes of tensor value along the $\widehat{\boldsymbol{\nabla}}\alpha_3$ as defined by Equations 5.15(b) and 5.15(c) will not change the value of μ_1 or μ_2. Considering Equation 5.15(b) (in which $\lambda_2 = \lambda_3$):

$$
\begin{aligned}
\mu_1(\mathbf{D} + \epsilon\sqrt{2}\widehat{\boldsymbol{\nabla}}\alpha_3) &= \frac{1}{3}(\lambda_1 + (1+\epsilon)\lambda_2 + (1-\epsilon)\lambda_3) \\
&= \frac{1}{3}(\lambda_1 + \lambda_2 + \lambda_3) \\
&= \mu_1(\mathbf{D}) \\
\Rightarrow \left.\frac{d}{d\epsilon}\mu_1(\mathbf{D} + \epsilon\widehat{\boldsymbol{\nabla}}\alpha_3)\right|_{\epsilon=0} &= 0 \\
\mu_2(\mathbf{D} + \epsilon\sqrt{2}\widehat{\boldsymbol{\nabla}}\alpha_3) &= \frac{1}{3}((\lambda_1 - \langle\lambda\rangle)^2 + ((1+\epsilon)\lambda_2 - \langle\lambda\rangle)^2 + ((1-\epsilon)\lambda_3 - \langle\lambda\rangle)^2) \\
&= \frac{1}{3}((\lambda_1 - \langle\lambda\rangle)^2 + (\lambda_2 - \langle\lambda\rangle)^2 + \epsilon^2\lambda_2^2 - 2\epsilon\lambda_2\langle\lambda\rangle \\
&\qquad + (\lambda_3 - \langle\lambda\rangle)^2 + \epsilon^2\lambda_3^2 + 2\epsilon\lambda_3\langle\lambda\rangle) \\
&= \mu_2(\mathbf{D}) + 2\epsilon^2\lambda_2^2 \\
\Rightarrow \left.\frac{d}{d\epsilon}\mu_2(\mathbf{D} + \epsilon\widehat{\boldsymbol{\nabla}}\alpha_3)\right|_{\epsilon=0} &= 0\ .
\end{aligned}
$$

Identical reasoning applies to Equation 5.15(c) (in which $\lambda_1 = \lambda_2$) to show that μ_1 and μ_2 do not change along $\widehat{\boldsymbol{\nabla}}\alpha_3$.

When all three eigenvalues are equal, we must pick two directions that are both orthogonal to $\widehat{\boldsymbol{\nabla}}\mu_1$. Equation 5.15(d) reuses the same gradient as 5.15(b), and Equation 5.14(b) is chosen so as to make $\widehat{\boldsymbol{\nabla}}\mu_2$ orthogonal to $\widehat{\boldsymbol{\nabla}}\alpha_3$. With these extended definitions, we now have $\widehat{\boldsymbol{\nabla}}\mu_1$, $\widehat{\boldsymbol{\nabla}}\mu_2$, and $\widehat{\boldsymbol{\nabla}}\alpha_3$, which are always defined and mutually orthogonal.

In light of Equations 5.14 and 5.15 above, two important caveats accompany the adoption of a coordinate system of shape based on $\widehat{\boldsymbol{\nabla}}\mu_1$, $\widehat{\boldsymbol{\nabla}}\mu_2$, and $\widehat{\boldsymbol{\nabla}}\alpha_3$. First, all analysis (filtering, edge detection, conductance functions, etc.) must treat $\widehat{\boldsymbol{\nabla}}\alpha_3$ the same as $-\widehat{\boldsymbol{\nabla}}\alpha_3$, since the sign is chosen arbitrarily when $\alpha_3 = 0$. Second, the analysis should recognize that $\widehat{\boldsymbol{\nabla}}\alpha_3$ becomes less and less meaningful as μ_2 approaches zero. In fact, at $\mu_2 = 0$, $\widehat{\boldsymbol{\nabla}}\alpha_3$ and $\widehat{\boldsymbol{\nabla}}\mu_2$ are interchangeable and arbitrary directions that can only be trusted for their orthogonality to $\widehat{\boldsymbol{\nabla}}\mu_1$. Ideally, image processing methods which employ the $\{\widehat{\boldsymbol{\nabla}}\mu_1, \widehat{\boldsymbol{\nabla}}\mu_2, \widehat{\boldsymbol{\nabla}}\alpha_3\}$ basis should gracefully make the transition from $\widehat{\boldsymbol{\nabla}}\alpha_3$ and $\widehat{\boldsymbol{\nabla}}\mu_2$ being distinctly significant (when $\mu_2 \gg 0$) to being interchangeable (when $\mu_2 = 0$).

5.3 Measuring Shape Gradients in Tensor Fields

In practice, one applies $\widehat{\boldsymbol{\nabla}}\mu_1$, $\widehat{\boldsymbol{\nabla}}\mu_2$, and $\widehat{\boldsymbol{\nabla}}\alpha_3$ to a tensor image by looking at the relationship between the *spatial* derivatives of the tensor (differentiating with respect to the image domain coordinates) and *tensor-valued* invariant gradients. We start by assuming that the tensor dataset is a smooth function $\mathbf{D}$ mapping from the image domain $\mathbb{R}^3$ to tensor values:

$$\mathbf{D} : \mathbb{R}^3 \quad \mapsto \quad Sym_3\ .$$

In practice, the tensor field will be only discretely sampled; we use separable convolution with continuous kernels (Section 2.1.8) to generate a continuous field. Then, the gradient of $\mathbf{D}$, $\nabla\mathbf{D}$, is a *third-order* tensor (see Section 2.1.6). $\nabla\mathbf{D}$ can be thought of as a

vector of tensors, each of which is the partial derivative of $\mathbf{D}$ along an image coordinate. The index notation for $\nabla\mathbf{D}$ involves all 27 tensor products of triples of basis vectors:

$$\begin{aligned} \nabla\mathbf{D} : \mathbb{R}^3 &\mapsto Sym_3^3 \\ \nabla\mathbf{D} = \frac{\partial\mathbf{D}}{\partial\mathbf{x}} &= \frac{\partial D_{jk}}{\partial x_i}\mathbf{b}_i \otimes \mathbf{b}_j \otimes \mathbf{b}_k \qquad (5.16) \\ (\nabla\mathbf{D})_{ijk} &= \frac{\partial D_{jk}}{\partial x_i} \, . \qquad (5.17) \end{aligned}$$

Previous work by Pajevic et al. introduced $\nabla\mathbf{D}$ to the DT-MRI literature [140], also in the context of measuring local changes in the tensor field. We will decompose $\nabla\mathbf{D}$ in greater detail, by using the invariant gradients defined in the previous section.

By differentiating every component of $\mathbf{D}$ along every axis, $\nabla\mathbf{D}$ measures all aspects of the first-order change in $\mathbf{D}$ at a given point. Following the definitions of vector and second-order tensor magnitude:

$$\begin{aligned} |\mathbf{v}| &= \sqrt{\mathbf{v}\cdot\mathbf{v}} = \sqrt{v_i v_i} \\ \|\mathbf{D}\| &= \sqrt{\mathbf{D} : \mathbf{D}} = \sqrt{D_{ij} D_{ij}} \end{aligned}$$

the magnitude of $\nabla\mathbf{D}$ may be measured as:

$$\|\nabla\mathbf{D}\| = \sqrt{\frac{\partial D_{jk}}{\partial i}\frac{\partial D_{jk}}{\partial i}} \, . \qquad (5.18)$$

One may directly verify the rotational invariance of $\|\nabla\mathbf{D}\|$, analogously to how Appendix A shows the invariance of $\mathbf{v}\cdot\mathbf{u}$.

Our interest is in the extent to which tensor change lies along a particular invariant gradient. Because of the difference between the un-normalized and normalized invariant gradients, for an invariant J we define two distinct *spatial* invariant gradients for a given tensor field $\mathbf{D}$:

$$\begin{aligned} \nabla J \, , \, \nabla J : \mathbb{R}^3 &\mapsto \mathbb{R}^3 \\ \nabla J(\mathbf{x}) &\equiv \boldsymbol{\nabla} J(\mathbf{D}(\mathbf{x})) : \nabla\mathbf{D}(\mathbf{x}) \qquad (5.19) \\ \widehat{\nabla} J(\mathbf{x}) &\equiv \widehat{\boldsymbol{\nabla}} J(\mathbf{D}(\mathbf{x})) : \nabla\mathbf{D}(\mathbf{x}) \, . \qquad (5.20) \end{aligned}$$

As described in Section 2.1.5, the double contractions between a second-order tensor (such as $\boldsymbol{\nabla} J$ or $\widehat{\boldsymbol{\nabla}} J$) and a third-order tensor (such as $\nabla\mathbf{D}$) is a first-order tensor, that is, a vector in $\mathbf{R}^3$. Then ∇J and $\widehat{\nabla} J$ are vectors, just like the gradient of any scalar field. There is an important difference between ∇J and $\widehat{\nabla} J$. Note that ∇J (Equation 5.19) is simply the chain rule applied to $J(\mathbf{D})$, as can be seen in the laboratory frame (using Equation 5.17):

$$\begin{aligned} ([\nabla J]_{\mathcal{L}})_k &= ([\boldsymbol{\nabla} J : \nabla\mathbf{D}]_{\mathcal{L}})_k \\ &= ([\boldsymbol{\nabla} J]_{\mathcal{L}})_{ij} ([\nabla\mathbf{D}]_{\mathcal{L}})_{ijk} \\ &= \frac{\partial J}{\partial D_{ij}}\frac{\partial D_{ij}}{\partial x_k} \, . \end{aligned}$$

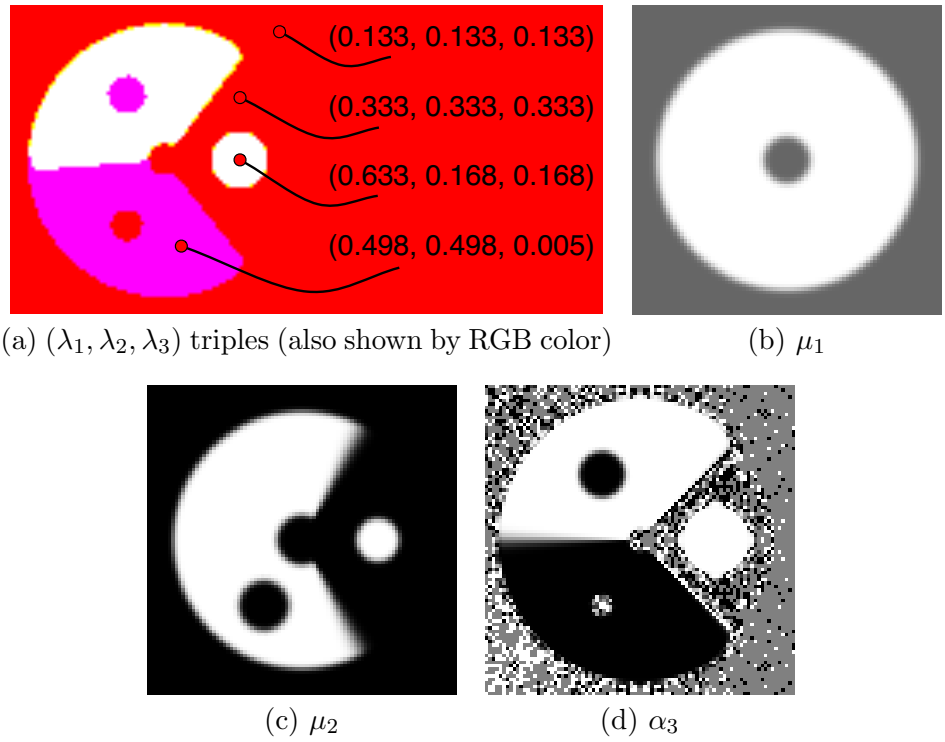
If $J(\mathbf{D}(\mathbf{x}))$ is at an extremum of J, then $\boldsymbol{\nabla} J$ is zero, which means $\nabla J(\mathbf{D}(\mathbf{x}))$ will be zero, *regardless* of $\nabla\mathbf{D}$. This is unfortunate: the original idea was to project changes in tensor

value onto the space of shape change, but ∇J additionally scales the magnitude of $\nabla \mathbf{D}$ as part of the projection. The definition of $\widehat{\nabla}$ remedies this, by projecting $\nabla \mathbf{D}$ onto the *span* of ∇J, rather than ∇J itself. Section 5.2 gave a definition for $\widehat{\boldsymbol{\nabla}} J$ for all tensor values, including values for which $\|\boldsymbol{\nabla} J\| = 0$.

Because $\|\boldsymbol{\nabla}\lambda_i\| = 1$, any component of $\nabla \mathbf{D}$ which represents a change in shape will have a nonzero projection onto $\{\boldsymbol{\nabla}\lambda_1, \boldsymbol{\nabla}\lambda_2, \boldsymbol{\nabla}\lambda_3\}$. The spatial gradients of individual eigenvalues ($\nabla\lambda_1$, $\nabla\lambda_2$, and $\nabla\lambda_3$, as defined by Equation 5.19) collectively indicate any and all spatial changes in tensor shape. The same cannot be said of $\nabla\mu_1$, $\nabla\mu_2$, and $\nabla\alpha_3$, because at some tensor values $\|\boldsymbol{\nabla}\mu_2\| = 0$ and/or $\|\boldsymbol{\nabla}\alpha_3\| = 0$. For example, $\nabla\mu_2$ will be zero at an isotropic point in a tensor field, around which anisotropy is increasing. Worse, $\nabla\alpha_3$ will be zero at linearly anisotropic points in a field, even if the shape of anisotropy is changing. Because both $\{\widehat{\boldsymbol{\nabla}}\mu_1, \widehat{\boldsymbol{\nabla}}\mu_2, \widehat{\boldsymbol{\nabla}}\alpha_3\}$ and $\{\boldsymbol{\nabla}\lambda_1, \boldsymbol{\nabla}\lambda_2, \boldsymbol{\nabla}\lambda_3\}$ are orthonormal bases for the same space of tensor shape change, projection of $\nabla \mathbf{D}$ onto the *span* of either basis is identical, but the individual components of the projection onto the $\{\widehat{\boldsymbol{\nabla}}\mu_1, \widehat{\boldsymbol{\nabla}}\mu_2, \widehat{\boldsymbol{\nabla}}\alpha_3\}$ basis have the anatomical relevance described in Section 2.3.3.

Figures 5.2 and 5.3 and show a synthetic tensor image constructed to demonstrate the invariant gradients described above. There are four distinct kinds of materials visible in Figure 5.2(a): isotropic with low diffusivity (dark gray), isotropic with high diffusivity (light gray), anisotropic with linear shape (red), and anisotropic with planar shape (yellow). There are boundaries between every pair of materials. As suggested by Figure 5.2(c), the design of the dataset ensures that the eigenvalue variance μ_2 is constant in the transition from linear to planar anisotropy. As seen in Figure 5.3, the orientation of the tensors also changes smoothly within the anisotropic regions.

Figure 5.4 displays various gradient magnitude measures applied to the synthetic tensor image. Note in Figure 5.4(a) that $\|\nabla \mathbf{D}\|$ is responding, as expected, to changes in orientation which are not accompanied by any changes in shape. In Figures 5.4(b) and 5.4(c), however, only the shape changes are being detected, and they are being detected equally by the $\{\widehat{\boldsymbol{\nabla}}\lambda_1, \widehat{\boldsymbol{\nabla}}\lambda_2, \widehat{\boldsymbol{\nabla}}\lambda_3\}$ and $\{\widehat{\boldsymbol{\nabla}}\mu_1, \widehat{\boldsymbol{\nabla}}\mu_2, \widehat{\boldsymbol{\nabla}}\alpha_3\}$ bases. Figures 5.4(d), 5.4(e), and 5.4(f) show how the tensors change with respect to individual eigenvalues, which does not facilitate distinction between, for example, the transition from low to high diffusivity, versus low to high anisotropy. Figures 5.4(g), 5.4(h), and 5.4(i), on the other hand, show that $\nabla\mu_1$, $\nabla\mu_2$, and $\nabla\alpha_3$ successfully respond in isolation to the different changes in shape. This kind of specificity in measuring different aspects of shape change, combined with the sensitivity of the $\{\widehat{\boldsymbol{\nabla}}\lambda_1, \widehat{\boldsymbol{\nabla}}\lambda_2, \widehat{\boldsymbol{\nabla}}\lambda_3\}$ shape change basis, has been the overall goal of the mathematics in this and the previous sections.



(a) $(\lambda_1, \lambda_2, \lambda_3)$ triples (also shown by RGB color) (b) μ_1

(c) μ_2 (d) α_3

Figure 5.2. Eigenvalues and invariants in 90×90 tensor synthetic image. The value of α_3 is essentially noise when μ_2 is at or near zero.

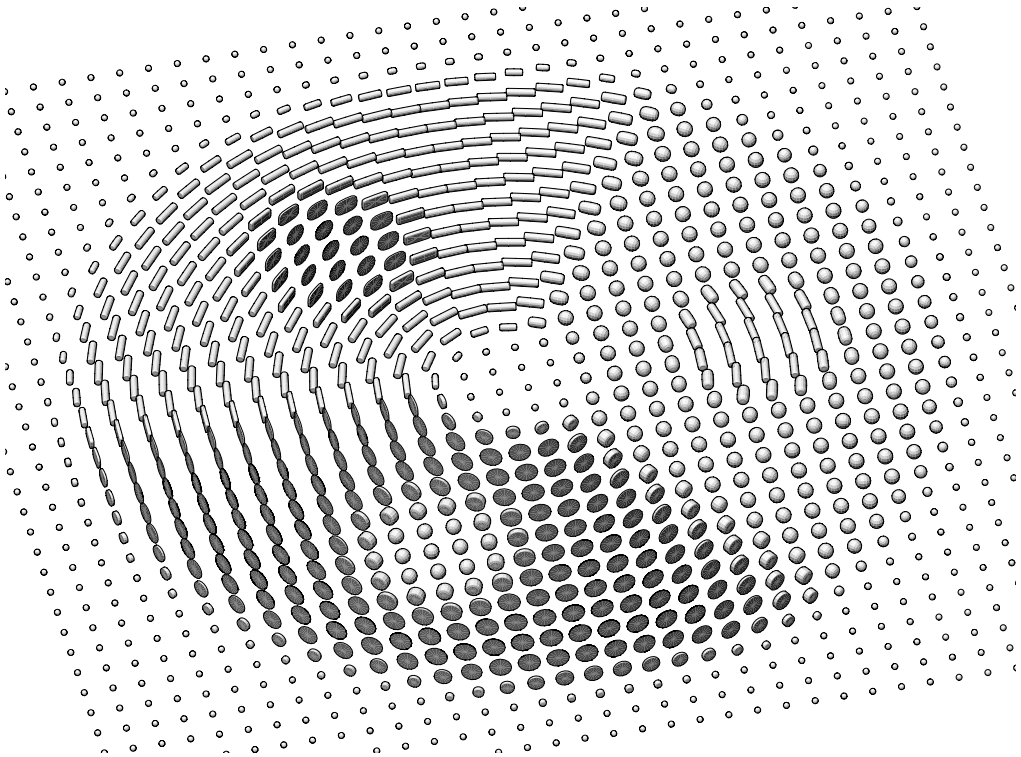
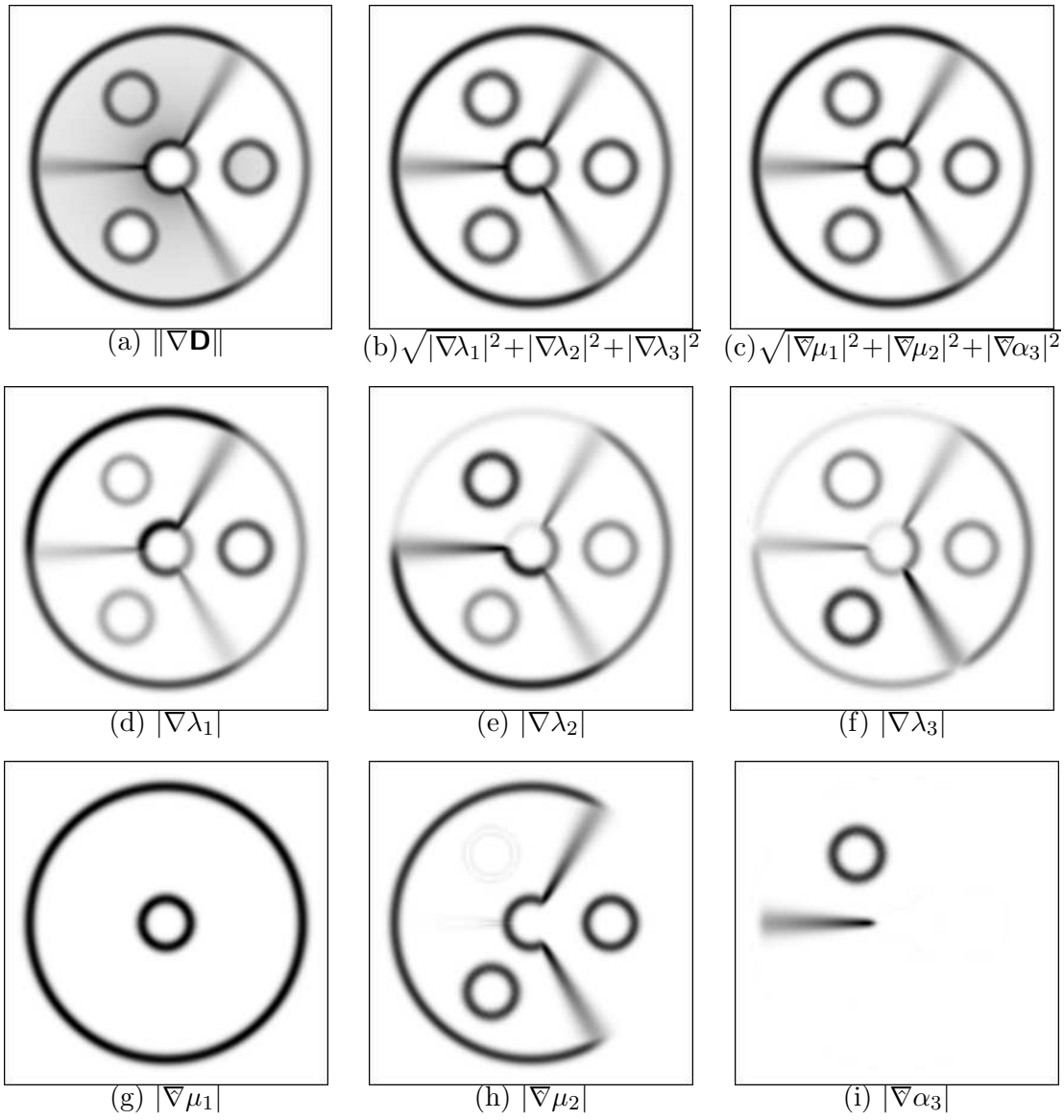


Figure 5.3. Display of synthetic tensor image with superquadric glyphs. Image was downsampled to 36×36 tensor samples for better visibility of individual glyphs. Viewpoint is off-axis for better depiction of the difference between linear and planar anisotropy, as well as glyph orientation.



(a) $\|\nabla \mathbf{D}\|$ (b) $\sqrt{|\nabla\lambda_1|^2+|\nabla\lambda_2|^2+|\nabla\lambda_3|^2}$ (c) $\sqrt{|\nabla\mu_1|^2+|\nabla\mu_2|^2+|\nabla\alpha_3|^2}$

(d) $|\nabla\lambda_1|$ (e) $|\nabla\lambda_2|$ (f) $|\nabla\lambda_3|$

(g) $|\nabla\mu_1|$ (h) $|\nabla\mu_2|$ (i) $|\nabla\alpha_3|$

Figure 5.4. Gradient magnitude in synthetic tensor image, measured in different ways. The grayscale values have been inverted (darker means higher gradient magnitude) for better display of features at low gradient magnitudes.

Figure 5.4(h) shows an example of how slight artifacts may be created by derivative measurement. Even though eigenvalue variance μ_2 is exactly constant at all tensor samples in the synthetic image, the small amount of blurring inherent in discrete differentiation causes slight changes in μ_2 near large changes in eigenvalue skewness α_3. One way to understand this is that in the space of tensor values, isosurfaces of μ_2 (unlike those of μ_1) are curved. Averaging between two tensors with equal μ_2 but different α_3 will artifactually lower μ_2, leading to faint edges in Figure 5.4(h) at the locations where $\nabla\alpha_3$ is largest. Such averaging is an unavoidable consequence of differentiation by convolution, although the choice of kernel plays a role in accuracy. A careful analysis of filter accuracy in the context of tensor processing (extending the work on scalar fields by Möller et al. [130] to tensor fields) is beyond the scope of this dissertation, but is an interesting direction for future work.

5.4 Rotations and Tangents to SO_3

The previous sections in this chapter described how to construct, using the normalized gradients of tensor invariants, a local coordinate system that captures all possible variations in tensor *shape* around a given tensor value. These are the directions indicated in the upper part of Figure 5.1. The following sections will complete the local coordinate system of tensor change with a basis spanning changes in tensor *orientation*. These directions are associated with rotations of the tensor, as with orientation changes along a twisting fiber tract in the brain. We term these tensor-valued directions "rotation tangents", because they are tangents to the orbits of the group action ψ (defined in Section 2.1.7), shown in the lower part of Figure 5.1. Before describing how to compute ψ, this section provides background on tangent spaces to manifolds, and the role of antisymmetric tensors. This type of material is covered in textbooks on theoretical kinematics [127].

The tangent space of a n-dimensional manifold M at a particular point $\mathbf{x} \in M$, notated $T_{\mathbf{x}}(M)$, can be thought of as a copy of $\mathbb{R}^n$ attached to M at $\mathbf{x}$. Suppose that $\boldsymbol{\sigma}(t)$ is a smooth path mapping $\mathbb{R}$ to M, and suppose that $\boldsymbol{\sigma}(0) = \mathbf{x}$. Then $\boldsymbol{\sigma}'(0)$ is an element of the tangent space $T_{\mathbf{x}}(M)$. In fact, $T_{\mathbf{x}}(M)$ can be constructed as the union of the tangents (at $\mathbf{x}$) to all smooth paths through $\mathbf{x}$.

By definition, a rotation $\mathbf{R} \in SO_3$ has the property that $\mathbf{R}^{\mathsf{t}} = \mathbf{R}^{-1}$. If $\boldsymbol{\sigma}(t)$ is a smooth path within SO_3, then $\boldsymbol{\sigma}(t)^{\mathsf{t}}\boldsymbol{\sigma}(t) = \boldsymbol{\sigma}(t)^{-1}\boldsymbol{\sigma}(t) = \mathbf{I}$ for all t. If $\boldsymbol{\sigma}(0) = \mathbf{I}$, then differentiating $\boldsymbol{\sigma}(t)^{\mathsf{t}}\boldsymbol{\sigma}(t) = \mathbf{I}$ at $t = 0$ gives:

$$\boldsymbol{\sigma}'(0)^{\mathsf{t}}\boldsymbol{\sigma}(0) + \boldsymbol{\sigma}(0)^{\mathsf{t}}\boldsymbol{\sigma}'(0) = \boldsymbol{\sigma}'(0)^{\mathsf{t}}\mathbf{I} + \mathbf{I}^{\mathsf{t}}\boldsymbol{\sigma}'(0) = \boldsymbol{\sigma}'(0)^{\mathsf{t}} + \boldsymbol{\sigma}'(0) = \mathbf{0} \; .$$

Thus, $\boldsymbol{\sigma}'(0)$ is an antisymmetric tensor. The set of all antisymmetric tensors is notated so_3. Because the path σ was arbitrary, we have:

$$T_{\mathbf{I}}(SO_3) = so_3 \; . \tag{5.21}$$

Considering that a rotation in SO_3 very near the identity $\mathbf{I}$ can be approximated by $\mathbf{I}+\boldsymbol{\omega}$ for a small magnitude $\boldsymbol{\omega} \in so_3$, we will sometimes refer to $\mathbf{I} + \boldsymbol{\omega}$ as a "small rotation".

If $\boldsymbol{\sigma}(t)$ is a smooth path in SO_3 for which $\boldsymbol{\sigma}(0) = \mathbf{R} \neq \mathbf{I}$, then we can define

$$\boldsymbol{\sigma}_0(t) = \mathbf{R}^{\mathsf{t}}\boldsymbol{\sigma}(t) \; ,$$

so that $\boldsymbol{\sigma}_0(0) = \mathbf{I}$. Then, $\boldsymbol{\sigma}_0'(0)$ is an element of so_3 (it is antisymmetric), and

$$\boldsymbol{\sigma}'(0) = \mathbf{R}\boldsymbol{\sigma}_0'(0) \Rightarrow \boldsymbol{\sigma}'(0) \in \mathbf{R}so_3 = \{\mathbf{R}\boldsymbol{\omega} | \boldsymbol{\omega} \in so_3\} \ .$$

In fact, as a consequence of the so-called "Lie group" structure of SO_3, one can show that [120, 35]

$$T_{\mathbf{R}}(SO_3) = \mathbf{R}so_3 \tag{5.22}$$

which generalizes Equation 5.21. That is, small changes in a rotation $\mathbf{R}$ can be represented by adding to $\mathbf{R}$ a tensor of the form $\mathbf{R}\boldsymbol{\omega}$, for a $\boldsymbol{\omega} \in so_3$.

An especially useful example of a smooth path through SO_3 is a uniformly parameterized rotation around a given fixed vector $\mathbf{a}$. Around $\mathbf{a}$ we can create a right-handed orthonormal basis $\mathcal{A} = \{\mathbf{u}, \mathbf{v}, \mathbf{a}\}$, so that $\mathbf{a} = \mathbf{u} \times \mathbf{v}$. Rotation around $\mathbf{a}$ by angle ϕ can be parameterized by $\mathbf{R}_{\mathbf{a}}(\phi)$, defined as:

$$[\mathbf{R}_{\mathbf{a}}(\phi)]_{\mathcal{A}} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} .$$

This generates (by Equation 2.14) a coordinate-free representation:

$$\mathbf{R}_{\mathbf{a}}(\phi) = \cos(\phi)(\mathbf{u} \otimes \mathbf{u} + \mathbf{v} \otimes \mathbf{v}) + \sin(\phi)(\mathbf{v} \otimes \mathbf{u} - \mathbf{u} \otimes \mathbf{v}) + \mathbf{a} \otimes \mathbf{a} \ .$$

Then by differentiating with respect to ϕ:

$$\mathbf{R}_{\mathbf{a}}'(0) = \mathbf{v} \otimes \mathbf{u} - \mathbf{u} \otimes \mathbf{v} \ .$$

To better understand the antisymmetric structure of $\mathbf{R}_{\mathbf{a}}'(0)$, we can determine its matrix components in an arbitrary orthonormal basis $\mathcal{B} = \{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_2\}$, in which $\mathbf{u} = u_i\mathbf{b}_i$, $\mathbf{v} = v_i\mathbf{b}_i$, and $\mathbf{a} = a_i\mathbf{b}_i$:

$$\begin{aligned} ([\mathbf{R}_{\mathbf{a}}'(0)]_{\mathcal{B}})_{ij} &= \mathbf{b}_i \cdot (\mathbf{v} \otimes \mathbf{u} - \mathbf{u} \otimes \mathbf{v})\mathbf{b}_j \\ &= \mathbf{b}_i \cdot (\mathbf{v}(\mathbf{u} \cdot \mathbf{b}_j) - \mathbf{u}(\mathbf{v} \cdot \mathbf{b}_j)) \\ &= (\mathbf{b}_i \cdot \mathbf{v})(\mathbf{u} \cdot \mathbf{b}_j) - (\mathbf{b}_i \cdot \mathbf{u})(\mathbf{v} \cdot \mathbf{b}_j) \\ &= u_j v_i - u_i v_j \ . \end{aligned} \tag{5.23}$$

Recall the definition of the vector cross-product $\mathbf{u} \times \mathbf{v}$:

$$\mathbf{a} = \mathbf{u} \times \mathbf{v} = \begin{vmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 \\ u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \end{vmatrix} \quad \Rightarrow \quad \begin{aligned} a_1 &= u_2v_3 - u_3v_2 \\ a_2 &= u_3v_1 - u_1v_3 \\ a_3 &= u_1v_2 - u_2v_1 \end{aligned} \ .$$

From this, and Equation 5.23, we find:

$$[\mathbf{R}_{\mathbf{a}}'(0)]_{\mathcal{B}} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} . \tag{5.24}$$

Multiplying $\mathbf{R}_{\mathbf{a}}'(0)$ by an arbitrary vector $\mathbf{b}$ shows that $\mathbf{R}_{\mathbf{a}}'(0)$ has an interesting property:

$$[\mathbf{R}_{\mathbf{a}}'(0)\mathbf{b}]_{\mathcal{B}} = [\mathbf{R}_{\mathbf{a}}'(0)]_{\mathcal{B}}[\mathbf{b}]_{\mathcal{B}} = \begin{bmatrix} a_2b_3 - a_3b_2 \\ a_3b_1 - a_1b_3 \\ a_1b_2 - a_2b_1 \end{bmatrix} = [\mathbf{a} \times \mathbf{b}]_{\mathcal{B}} \ .$$

That is, multiplying by $\mathbf{R}'_{\mathbf{a}}(0)$ has the same effect as taking the vector cross-product with $\mathbf{a}$. The tensor that effectively performs the vector cross-product with a vector $\mathbf{a}$ can be notated $\mathbf{X}(\mathbf{a})$[1]:

$$\mathbf{X}(\mathbf{a})\mathbf{b} = \mathbf{a} \times \mathbf{b} \quad \text{for all } \mathbf{b} \; . \tag{5.25}$$

To summarize, the derivative of a smooth rotation around $\mathbf{a}$ is an antisymmetric tensor which has the same multiplicative effect as taking the cross-product with $\mathbf{a}$:

$$\mathbf{R}'_{\mathbf{a}}(0) = \mathbf{X}(\mathbf{a}) \; .$$

On the other hand, from Equations 5.24 and 5.25, one can show that *any* antisymmetric tensor $\boldsymbol{\omega}$ can be expressed as $\mathbf{X}(\mathbf{w})$ for some vector $\mathbf{w}$.

5.5 Derivatives of ψ: Rotation Tangents

The background material in the previous section provides the basis for discussing derivatives of the group action ψ. The derivative of a map between manifolds is a linear map from tangent spaces at points in the domain to tangent spaces at points in the range. Recall from the definition of ψ (Equation 2.16) that ψ maps from a rotation in SO_3 and a tensor in Sym_3 to another tensor in Sym_3:

$$\begin{aligned} \psi : SO_3 \times Sym_3 &\mapsto Sym_3 \\ \psi(\mathbf{R}, \mathbf{D}) &= \mathbf{R}\mathbf{D}\mathbf{R}^{\mathsf{t}} \; . \end{aligned}$$

The derivative of ψ is a linear map between tangent spaces of SO_3 and Sym_3:

$$d\psi_{(\mathbf{R},\mathbf{D})} : T_{\mathbf{R}}(SO_3) \times T_{\mathbf{D}}(Sym_3) \mapsto T_{\psi(\mathbf{R},\mathbf{D})}(Sym_3) \; .$$

If $\boldsymbol{\sigma}(t)$ is a smooth path in the vector space Sym_3, then the tangents $\boldsymbol{\sigma}'(t)$ are also contained in the same vector space Sym_3, so Sym_3 is identified with its tangent spaces:

$$T_{\mathbf{D}}(Sym_3) = Sym_3 \; \forall \; \mathbf{D} \in Sym_3 \; .$$

Recall from the previous section that:

$$T_{\mathbf{R}}(SO_3) = \mathbf{R}so_3 \; .$$

Then, $d\psi$ can be more simply described as:

$$d\psi_{(\mathbf{R},\mathbf{D})} : \mathbf{R}so_3 \times Sym_3 \mapsto Sym_3 \; .$$

Following the approach of Damon [48], we calculate $d\psi$ at $(\mathbf{R}\boldsymbol{\omega}, \boldsymbol{\epsilon})$ by evaluating $\psi(\mathbf{R} + t\mathbf{R}\boldsymbol{\omega}, \mathbf{D} + t\boldsymbol{\epsilon})$ and extracting the terms linear in t:

$$\begin{aligned} \psi(\mathbf{R} + t\mathbf{R}\boldsymbol{\omega}, \mathbf{D} + t\boldsymbol{\epsilon}) &= (\mathbf{R} + t\mathbf{R}\boldsymbol{\omega})(\mathbf{D} + t\boldsymbol{\epsilon})(\mathbf{R} + t\mathbf{R}\boldsymbol{\omega})^{\mathsf{t}} \\ &= \mathbf{R}(\mathbf{I} + t\boldsymbol{\omega})(\mathbf{D} + t\boldsymbol{\epsilon})(\mathbf{I} + t\boldsymbol{\omega})^{\mathsf{t}}\mathbf{R}^{\mathsf{t}} \\ &= \mathbf{R}(\mathbf{I} + t\boldsymbol{\omega})(\mathbf{D} + t\boldsymbol{\epsilon})(\mathbf{I} - t\boldsymbol{\omega})\mathbf{R}^{\mathsf{t}} \\ &= \mathbf{R}(\mathbf{I} + t\boldsymbol{\omega})(\mathbf{D} + t\boldsymbol{\epsilon} - t\mathbf{D}\boldsymbol{\omega} - t^2\boldsymbol{\epsilon}\boldsymbol{\omega})\mathbf{R}^{\mathsf{t}} \\ &= \mathbf{R}(\mathbf{D} + t\boldsymbol{\epsilon} - t\mathbf{D}\boldsymbol{\omega} - t^2\boldsymbol{\epsilon}\boldsymbol{\omega} + t\boldsymbol{\omega}\mathbf{D} + t^2\boldsymbol{\omega}\boldsymbol{\epsilon} - t^2\boldsymbol{\omega}\mathbf{D}\boldsymbol{\omega} - t^3\boldsymbol{\omega}\boldsymbol{\epsilon}\boldsymbol{\omega})\mathbf{R}^{\mathsf{t}} \\ \Rightarrow d\psi_{(\mathbf{R},\mathbf{D})}(\mathbf{R}\boldsymbol{\omega}, \boldsymbol{\epsilon}) &= \mathbf{R}(\boldsymbol{\omega}\mathbf{D} - \mathbf{D}\boldsymbol{\omega} + \boldsymbol{\epsilon})\mathbf{R}^{\mathsf{t}} \; . \end{aligned} \tag{5.26}$$

Equation 5.26 says that a change $\boldsymbol{\epsilon}$ in $\mathbf{D}$ produces a change $\mathbf{R}\boldsymbol{\epsilon}\mathbf{R}^{\mathsf{t}}$ in $\psi(\mathbf{R}, \mathbf{D})$, while a small rotation $\mathbf{R}(\mathbf{I} + \boldsymbol{\omega})$ produces a change $\mathbf{R}(\boldsymbol{\omega}\mathbf{D} - \mathbf{D}\boldsymbol{\omega})\mathbf{R}^{\mathsf{t}}$. Because we intend to focus

[1] The kinematics literature does not seem to have standard notation for this function [85].

on the differential structure of changes in $\mathbf{D} = \psi(\mathbf{I}, \mathbf{D})$ rather than in $\mathbf{RDR}^{\mathsf{t}} = \psi(\mathbf{R}, \mathbf{D})$, we can set $\mathbf{R} = \mathbf{I}$. Also, for the purposes of this section (the calculation of rotation tangents), we need only consider *rotations* of $\mathbf{D}$, so we can set $\boldsymbol{\epsilon} = \mathbf{0}$. With these simplifications:

$$d\psi_{(\mathbf{I},\mathbf{D})}(\boldsymbol{\omega}, \mathbf{0}) = \boldsymbol{\omega}\mathbf{D} - \mathbf{D}\boldsymbol{\omega} \; .$$

At this point, we may as well drop from $d\psi$ the $(\mathbf{I}, \mathbf{D})$ subscript, and the second argument $\mathbf{0}$, since these will be fixed from now on:

$$d\psi(\boldsymbol{\omega}) = d\psi_{(\mathbf{I},\mathbf{D})}(\boldsymbol{\omega}, \mathbf{0}) = \boldsymbol{\omega}\mathbf{D} - \mathbf{D}\boldsymbol{\omega} \; . \tag{5.27}$$

Equation 5.27 is a general expression for all possible rotation tangents. To show a concrete example of a rotation tangent, recall that the principal frame $\mathcal{E} = \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ is the basis for $\mathbb{R}^3$ spanned by eigenvectors of a given tensor $\mathbf{D}$. Consider the image, under $d\psi$, of the tangent associated with rotation around $\mathbf{e}_1$, namely $\mathbf{R}'_{\mathbf{e}_1}(0) = \mathbf{X}(\mathbf{e}_1)$:

$$\begin{aligned}
[d\psi(\mathbf{X}(\mathbf{e}_1))]_{\mathcal{E}} &= [\mathbf{X}(\mathbf{e}_1)\mathbf{D} - \mathbf{D}\mathbf{X}(\mathbf{e}_1)]_{\mathcal{E}} \\
&= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} - \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \lambda_2 - \lambda_3 \\ 0 & \lambda_2 - \lambda_3 & 0 \end{bmatrix} \; .
\end{aligned}$$

Then by Equation 2.14:

$$d\psi(\mathbf{X}(\mathbf{e}_1)) = (\lambda_2 - \lambda_3)(\mathbf{e}_2 \otimes \mathbf{e}_3 + \mathbf{e}_3 \otimes \mathbf{e}_2) \; .$$

This equation describes exactly how the tensor value $\mathbf{D}$ changes as a result of small rotations of the tensor around $\mathbf{e}_1$. By similar reasoning one can derive expressions for the rotation tangents associated with $\mathbf{e}_2$ and $\mathbf{e}_3$:

$$\begin{aligned}
d\psi(\mathbf{X}(\mathbf{e}_2)) &= (\lambda_3 - \lambda_1)(\mathbf{e}_3 \otimes \mathbf{e}_1 + \mathbf{e}_1 \otimes \mathbf{e}_3) \\
d\psi(\mathbf{X}(\mathbf{e}_3)) &= (\lambda_1 - \lambda_2)(\mathbf{e}_1 \otimes \mathbf{e}_2 + \mathbf{e}_2 \otimes \mathbf{e}_1) \; .
\end{aligned}$$

We define *principal rotation tangents* $\boldsymbol{\Phi}_i$ according to the expressions above:

$$\begin{aligned}
\boldsymbol{\Phi}_1 &= d\psi(\mathbf{X}(\mathbf{e}_1)) = (\lambda_2 - \lambda_3)(\mathbf{e}_2 \otimes \mathbf{e}_3 + \mathbf{e}_3 \otimes \mathbf{e}_2) \\
\boldsymbol{\Phi}_2 &= d\psi(\mathbf{X}(\mathbf{e}_2)) = (\lambda_3 - \lambda_1)(\mathbf{e}_3 \otimes \mathbf{e}_1 + \mathbf{e}_1 \otimes \mathbf{e}_3) \\
\boldsymbol{\Phi}_3 &= d\psi(\mathbf{X}(\mathbf{e}_3)) = (\lambda_1 - \lambda_2)(\mathbf{e}_1 \otimes \mathbf{e}_2 + \mathbf{e}_2 \otimes \mathbf{e}_1)
\end{aligned} \; . \tag{5.28}$$

Equation 2.15 states that:

$$(\mathbf{e}_i \otimes \mathbf{e}_j) : (\mathbf{e}_k \otimes \mathbf{e}_l) = \delta_{ik}\delta_{jl} \; .$$

This implies the principal rotation tangents $\boldsymbol{\Phi}_i$ are mutually orthogonal. For example,

$$\begin{aligned}
\boldsymbol{\Phi}_1 : \boldsymbol{\Phi}_2 &= (\lambda_2 - \lambda_3)(\mathbf{e}_2 \otimes \mathbf{e}_3 + \mathbf{e}_3 \otimes \mathbf{e}_2) : (\lambda_3 - \lambda_1)(\mathbf{e}_3 \otimes \mathbf{e}_1 + \mathbf{e}_1 \otimes \mathbf{e}_3) \\
&= (\lambda_2 - \lambda_3)(\lambda_3 - \lambda_1)\begin{pmatrix} \mathbf{e}_2 \otimes \mathbf{e}_3 : \mathbf{e}_3 \otimes \mathbf{e}_1 \\ + \, \mathbf{e}_3 \otimes \mathbf{e}_2 : \mathbf{e}_3 \otimes \mathbf{e}_1 \\ + \, \mathbf{e}_2 \otimes \mathbf{e}_3 : \mathbf{e}_1 \otimes \mathbf{e}_3 \\ + \, \mathbf{e}_3 \otimes \mathbf{e}_2 : \mathbf{e}_1 \otimes \mathbf{e}_3 \end{pmatrix} \\
&= (\lambda_2 - \lambda_3)(\lambda_3 - \lambda_1)(\delta_{23}\delta_{31} + \delta_{33}\delta_{21} + \delta_{21}\delta_{33} + \delta_{31}\delta_{23}) \\
&= 0 \; .
\end{aligned}$$

Similar reasoning applies for $\mathbf{\Phi}_1 : \mathbf{\Phi}_3$ and $\mathbf{\Phi}_2 : \mathbf{\Phi}_3$. Then:

$$\begin{aligned} \mathbf{\Phi}_1 : \mathbf{\Phi}_2 &= 0 \\ \mathbf{\Phi}_1 : \mathbf{\Phi}_3 &= 0 \\ \mathbf{\Phi}_2 : \mathbf{\Phi}_3 &= 0 \,. \end{aligned} \tag{5.29}$$

Although the $\mathbf{\Phi}_i$ are mutually orthogonal, they are not *orthonormal*, because they do not have unit norm:

$$\begin{aligned} \|\mathbf{\Phi}_1\| &= \sqrt{2}\,|\lambda_2 - \lambda_3| \\ \|\mathbf{\Phi}_2\| &= \sqrt{2}\,|\lambda_3 - \lambda_1| \\ \|\mathbf{\Phi}_3\| &= \sqrt{2}\,|\lambda_1 - \lambda_2| \,. \end{aligned}$$

It turns out at the $\mathbf{\Phi}_i$ defined above support something akin to a spectral decomposition of arbitrary rotation tangents. Like the spectral decomposition of a tensor value, the spectral decomposition of rotation tangents unfortunately incurs the computation cost of determining the eigensystem of the tensor. Given an arbitrary axis of rotation $\mathbf{a}$, we can determine its coefficients a_i in the principal frame: $\mathbf{a} = a_i \mathbf{e}_i$. Recall from Section 5.4 that $\mathbf{R}_\mathbf{a}(\phi)$ is the rotation by ϕ around axis $\mathbf{a}$. We are interested in how tensor $\mathbf{D}$ changes as result of small rotations around $\mathbf{a}$, in other words, how $\psi(\mathbf{R}_\mathbf{a}(\phi), \mathbf{D})$ changes for small ϕ. The spectral decomposition of the corresponding rotation tangent is:

$$\begin{aligned} \left.\frac{d}{d\phi}\psi(\mathbf{R}_\mathbf{a}(\phi), \mathbf{D})\right|_{\phi=0} &= d\psi(\mathbf{R}'_\mathbf{a}(0)) \\ &= d\psi(\mathbf{X}(\mathbf{a})) \\ &= a_i \, d\psi(\mathbf{X}(\mathbf{e}_i)) \\ &= a_i \mathbf{\Phi}_i \,. \end{aligned}$$

With this, we can compute the inner product between two arbitrary rotations tangents, associated with rotations around $\mathbf{a} = a_i \mathbf{e}_i$ and $\mathbf{b} = b_i \mathbf{e}_i$:

$$\begin{aligned} d\psi(\mathbf{X}(\mathbf{a})) : d\psi(\mathbf{X}(\mathbf{b})) &= a_i \mathbf{\Phi}_i : b_j \mathbf{\Phi}_j \\ &= a_i b_j (\mathbf{\Phi}_i : \mathbf{\Phi}_j) \\ &= \textstyle\sum_i a_i b_i \|\mathbf{\Phi}_i\|^2 \\ &= 2(a_1 b_1 (\lambda_2 - \lambda_3)^2 + a_2 b_2 (\lambda_3 - \lambda_1)^2 + a_3 b_3 (\lambda_1 - \lambda_2)^2). \end{aligned} \tag{5.30}$$

Recall from Equation 5.5 that:

$$\boldsymbol{\nabla} J = \frac{\partial J}{\partial \lambda_1}(\mathbf{e}_1 \otimes \mathbf{e}_1) + \frac{\partial J}{\partial \lambda_2}(\mathbf{e}_2 \otimes \mathbf{e}_2) + \frac{\partial J}{\partial \lambda_3}(\mathbf{e}_3 \otimes \mathbf{e}_3) \,.$$

Comparing this with the expressions for $\mathbf{\Phi}_i$ (Equation 5.28), and again recalling Equation 2.15, it follows that all rotation tangents are orthogonal to all invariant gradients:

$$d\psi(\mathbf{R}'_\mathbf{a}(0)) : \boldsymbol{\nabla} J = 0 \,. \tag{5.31}$$

The unavoidable sign ambiguity of the eigenvectors $\mathbf{e}_i$ causes a sign ambiguity of $\mathbf{\Phi}_i$. Any subsequent processing or analysis based on computing $\mathbf{\Phi}_i$ needs to be insensitive to the difference between $\mathbf{\Phi}_i$ and $-\mathbf{\Phi}_i$.

5.6 Space of Tensor Orientation

Some insight into the space of tensor orientation can be derived from considering the norm of an arbitrary rotation tangent. Given an arbitrary axis of rotation $\mathbf{a}$, $\|d\psi(\mathbf{X}(\mathbf{a}))\|$ can be found directly from Equation 5.30:

$$\begin{aligned} \|d\psi(\mathbf{X}(\mathbf{a}))\| &= \sqrt{d\psi(\mathbf{X}(\mathbf{a})) : d\psi(\mathbf{X}(\mathbf{a}))} \\ &= \sqrt{2(a_1^2(\lambda_2-\lambda_3)^2 + a_2^2(\lambda_3-\lambda_1)^2 + a_3^2(\lambda_1-\lambda_2)^2)}\,. \end{aligned} \tag{5.32}$$

Equation 5.32 quantifies the intuitive notion that rotations create change only in the presence of anisotropy, by showing that the magnitude of a rotation tangent is determined by the differences between eigenvalues, as well as the components of the rotation axis in the principal frame. A connection to tensor anisotropy can be made more precise by calculating the expected value of $\|d\psi(\mathbf{X}(\mathbf{a}))\|^2$ over the sphere of unit-length rotation axes $\mathbf{a}$:

$$\begin{aligned} \left\langle \|d\psi(\mathbf{X}(\mathbf{a}))\|^2 \right\rangle &= \frac{1}{4\pi}\int_0^{\pi}\int_0^{2\pi} 2 \begin{pmatrix} \cos^2(\theta)\sin^2(\phi)(\lambda_2-\lambda_3)^2 \\ +\sin^2(\theta)\sin^2(\phi)(\lambda_3-\lambda_1)^2 \\ +\cos^2(\phi)(\lambda_1-\lambda_2)^2 \end{pmatrix} \sin(\phi)\,d\theta\,d\phi \\ &= 2((\lambda_2-\lambda_3)^2 + (\lambda_1-\lambda_3)^2 + (\lambda_1-\lambda_2)^2)/3 \\ &= 6\mu_2\,. \end{aligned}$$

Another way to express this is that the RMS value of $\|d\psi(\mathbf{X}(\mathbf{a}))\|$ is proportional to $\sqrt{\mu_2}$, the standard deviation of the eigenvalues. Equations 2.35 and 2.36 show that fractional anisotropy and relative anisotropy also vary linearly with $\sqrt{\mu_2}$, though they are normalized by different invariants measuring overall size.

Equation 5.32 highlights a fundamental difference between the space of shape changes and the space of orientation changes. The dimension of the space of orientation change around a given tensor $\mathbf{D}$ is exactly the dimension of $SO_3(\mathbf{D})$, the orbit of the group action ψ containing $\mathbf{D}$. The orbit dimension can be determined by counting the number of orthogonal nonzero tangents to it. If $\lambda_1 = \lambda_2 = \lambda_3$, then $\|\mathbf{\Phi}_1\| = \|\mathbf{\Phi}_2\| = \|\mathbf{\Phi}_3\| = 0$, so the dimension of the orbit is zero. If exactly two eigenvalues are equal, then exactly one of the $\mathbf{\Phi}_i$ is zero-magnitude, so the dimension of the orbit is two. If all the eigenvalues are distinct, then all $\mathbf{\Phi}_i$ are nonzero, and the orbit is three-dimensional.

Figure 5.5 illustrates how dimension of the orientation space changes gradually, in a way governed by eigenvalue variance μ_2 and skewness α_3. Figure 5.5(a) shows a triangular isosurface of μ_1 in the three-dimensional space of tensors that are diagonal in the laboratory frame. On the μ_1 isosurface, μ_2 and α_3 are plotted in Figures 5.5(b) and 5.5(c). Figure 5.5(d) displays the tensors sampled on the same triangular domain, using superquadric glyphs. Figure 5.5(e) aims to visualize the space of tensor orientation, based on tensors defined in terms of the principal rotation tangent magnitudes. For each tensor $\mathbf{D}$ shown in Figure 5.5(d), where

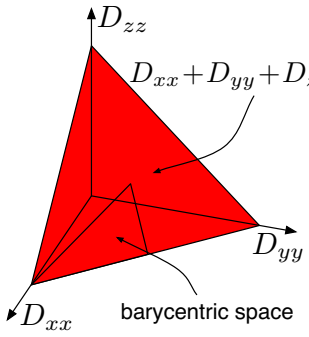
$$\mathbf{D} = \textstyle\sum_i \lambda_i (\mathbf{e}_i \otimes \mathbf{e}_i)$$

we define a new tensor $\mathbf{P}$ by:

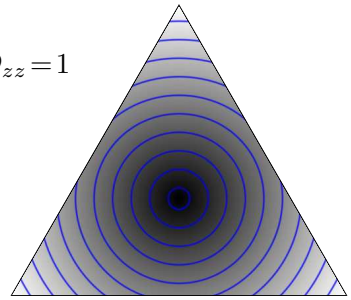
$$\mathbf{P} = \textstyle\sum_i \|\mathbf{\Phi}_i\| (\mathbf{e}_i \otimes \mathbf{e}_i)$$

The tensors $\mathbf{P}$ and $\mathbf{D}$ share the same eigenvectors, but the eigenvalue of $\mathbf{P}$ along $\mathbf{e}_i$ is the magnitude of the rotation tangent associated with $\mathbf{e}_i$. When $\mathbf{D}$ has axial symmetry,

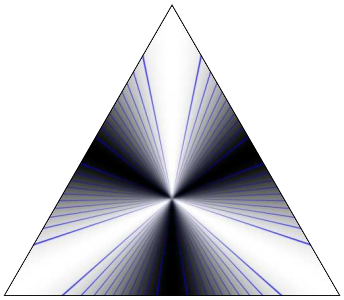
due to either linear or planar anisotropy, the orientation space is two-dimensional, and one eigenvalue of **P** is zero, thus the glyph representation is a flat disc. Figure 5.5(e) shows how the glyphs of **P** gradually flatten to discs as the eigenvalue skewness α_3 approaches both its minimum value ($-1/\sqrt{2}$, planar anisotropy) and maximum value ($1/\sqrt{2}$, linear anisotropy). The overall size of **P** varies linearly with the distance from the center of the triangular domain, that is, the eigenvalue standard deviation. Interestingly, the only point at which **P** is isotropic is when **P** is zero.



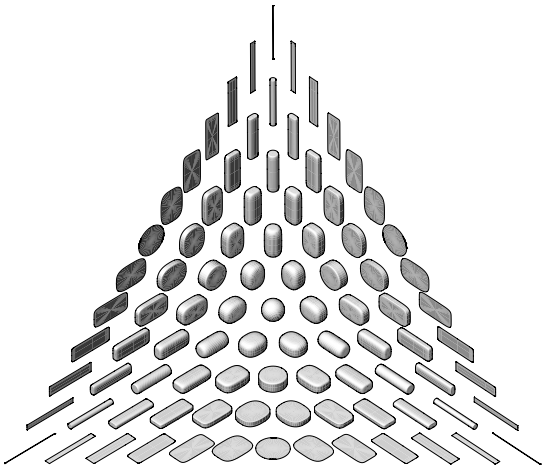
(a) Isosurface of μ_1



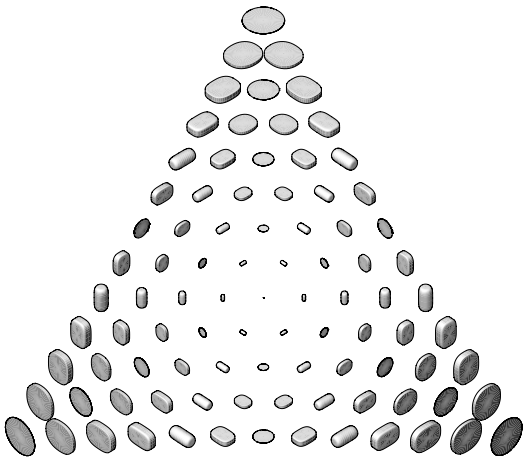
(b) Contour plot of $\sqrt{\mu_2}$



(c) Contour plot of α_3



(d) Glyphs of **D**



(e) Glyphs of **P**

Figure 5.5. The space of orientation changes, as represented by **P**, is reduced from three to two dimensions when eigenvalue skewness α_3 is at extremum, and is reduced to zero dimensions when eigenvalue variance μ_2 is zero.

5.7 Orthonormal Rotation Tangents

Section 5.2 described how to define a local orthonormal basis for shape change at all tensor values. This section will do the same for orientation changes. The principal rotation tangents $\mathbf{\Phi}_i$ previously defined in Section 5.5 (Equation 5.28) are orthogonal but not orthonormal. The normalization is straightforward:

$$\begin{aligned} \widehat{\mathbf{\Phi}}_1 &= (\mathbf{e}_2 \otimes \mathbf{e}_3 + \mathbf{e}_3 \otimes \mathbf{e}_2)/\sqrt{2} \\ \widehat{\mathbf{\Phi}}_2 &= (\mathbf{e}_3 \otimes \mathbf{e}_1 + \mathbf{e}_1 \otimes \mathbf{e}_3)/\sqrt{2} \\ \widehat{\mathbf{\Phi}}_3 &= (\mathbf{e}_1 \otimes \mathbf{e}_2 + \mathbf{e}_2 \otimes \mathbf{e}_1)/\sqrt{2}\,. \end{aligned} \tag{5.33}$$

The simplicity of the definitions of the orthonormal rotation tangents $\widehat{\mathbf{\Phi}}_i$, as compared to the normalized invariant gradients $\widehat{\boldsymbol{\nabla}}\mu_2$ and $\widehat{\boldsymbol{\nabla}}\alpha_3$ (Equations 5.14 and 5.15), is directly due to the fact that $\widehat{\mathbf{\Phi}}_i$ are defined in terms of eigenvectors. This simplicity is therefore analogous to that of the eigenvalue gradient $\widehat{\boldsymbol{\nabla}}\lambda_i$ definitions (Equations 5.2 and 5.11). In both cases, the apparent simplicity of the *mathematical* definition is actually benefiting from the greater *computational* effort required to determine the eigenvectors (Sections 2.3.1 and 3.1.1).

We are currently not aware of a method to efficiently compute orthonormal rotation tangents without first computing eigenvectors. One possible approach is to start with the rotation tangents associated with the basis vectors of the laboratory frame $\mathcal{L} = \{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$: $\{d\psi(\mathbf{X}(\mathbf{x})), d\psi(\mathbf{X}(\mathbf{y})), d\psi(\mathbf{X}(\mathbf{z}))\}$. However, according to Equation 5.30, these are not necessarily orthogonal, nor are they unit magnitude. A procedure such as Principal Component Analysis may be applied to find an orthogonal set of vectors in the span of $\{d\psi(\mathbf{X}(\mathbf{x})), d\psi(\mathbf{X}(\mathbf{y})), d\psi(\mathbf{X}(\mathbf{z}))\}$, but the computational expense of this is not trivial, comparable to the eigensystem computation which is currently required.

It should be noted that the normalized invariant gradients $\{\widehat{\boldsymbol{\nabla}}\mu_1, \widehat{\boldsymbol{\nabla}}\mu_2, \widehat{\boldsymbol{\nabla}}\alpha_3\}$ and the normalized rotation tangents $\{\widehat{\mathbf{\Phi}}_1, \widehat{\mathbf{\Phi}}_2, \widehat{\mathbf{\Phi}}_3\}$ do in fact span the entire space of symmetric tensors. Equation 5.7 stated that $\{\boldsymbol{\nabla}\mu_1, \boldsymbol{\nabla}\mu_2, \boldsymbol{\nabla}\alpha_3\}$ were orthogonal, and the definitions of $\widehat{\boldsymbol{\nabla}}\mu_2$ and $\widehat{\boldsymbol{\nabla}}\alpha_3$ (Equations 5.14 and 5.15) were careful to preserve this. Equation 5.31 stated that all invariant gradients are orthogonal to all rotation tangents. Equation 5.28 defined three rotation tangents which are mutually orthogonal, a property unchanged by their normalization in Equation 5.33. Thus

$$\{\widehat{\boldsymbol{\nabla}}\mu_1, \widehat{\boldsymbol{\nabla}}\mu_2, \widehat{\boldsymbol{\nabla}}\alpha_3, \widehat{\mathbf{\Phi}}_1, \widehat{\mathbf{\Phi}}_2, \widehat{\mathbf{\Phi}}_3\}$$

is an orthonormal set of tensors with a six-dimensional span. Because everything in the span is a symmetric tensor, and the space of symmetric tensors Sym_3 is six dimensional, the normalized invariant gradients and rotation tangents span all of Sym_3 at *all* tensor values, including places where two or more eigenvalues are equal (as per the extended definitions of invariant gradients in Section 5.2).

5.8 Measuring Orientation Changes in Tensor Fields

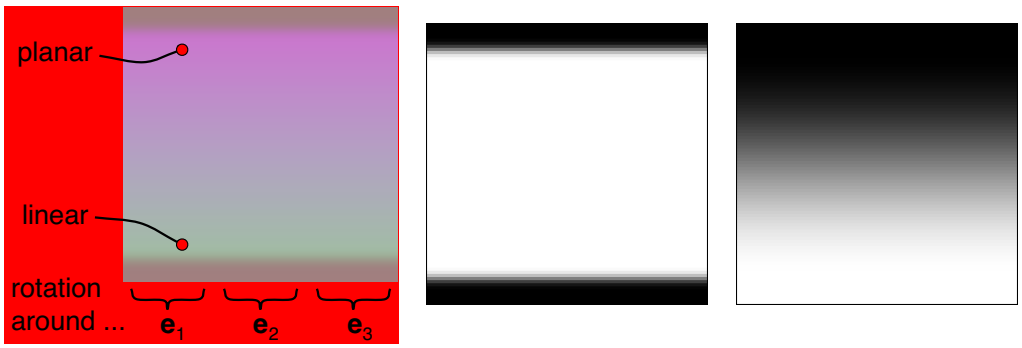
As with the normalized invariant gradients of Section 5.2, the practical utility of the normalized rotation tangents $\widehat{\mathbf{\Phi}}_i$ is in measuring their alignment with the *spatial* gradient

of the tensor field, $\nabla \mathbf{D}$, which is a third-order tensor. Mimicking the notation of ∇J and ∇J from Section 5.3, we define two different measures of spatial orientation change:

$$\begin{aligned} \nabla\phi_i \, , \, \nabla\phi_i : \mathbb{R}^3 &\mapsto \mathbb{R}^3 \\ \nabla\phi_i(\mathbf{x}) &\equiv \mathbf{\Phi}_i(\mathbf{D}(\mathbf{x})) : \nabla\mathbf{D}(\mathbf{x}) \qquad (5.34) \\ \nabla\phi_i(\mathbf{x}) &\equiv \widehat{\mathbf{\Phi}}_i(\mathbf{D}(\mathbf{x})) : \nabla\mathbf{D}(\mathbf{x}) \; . \qquad (5.35) \end{aligned}$$

Note that the notation "$\nabla\phi_i$" and "$\nabla\phi_i$" is intended to be suggestive rather than literal. We define no scalar function ϕ_i which is differentiated to form $\nabla\phi_i$. Rather, $\nabla\phi_i$ indicates the spatial direction along which the tensor values rotate around eigenvector $\mathbf{e}_i$ fastest, based on the notion that "ϕ_i" represents the angle of the rotation. $\nabla\phi_i$ is the analogous function defined to ensure equal sensitivity to all orientation changes, regardless of tensor value.

Figure 5.6 shows the overall structure of a synthetic tensor image, designed to demonstrate the detection of tensor change along rotation tangents. The eigenvalue mean μ_1 is held constant throughout the image, and the variance μ_2 is constant except for bands at the top and bottom of the image. The eigenvalue skewness α_3 varies smoothly from the top to the bottom of the high μ_2 region, covering the full range of shapes from planar to linear anisotropy. Horizontally, the image is divided into three regions of rotation, around each of the eigenvectors. Figure 5.7 displays how the rotation affects the different glyph shapes, including the rotation symmetries of linear and planar anisotropy.



(a) Image layout; $(\lambda_1, \lambda_2, \lambda_3)$ triples shown by RGB color

(b) μ_2

(c) α_3

Figure 5.6. Layout and invariants in 90×90 tensor synthetic image. The range of α_3 along vertical axis is $-1/\sqrt{2}$ to $1/\sqrt{2}$, as determined by linearly varying Θ from $\pi/3$ to 0. Rotation around eigenvectors occurs in three distinct regions.

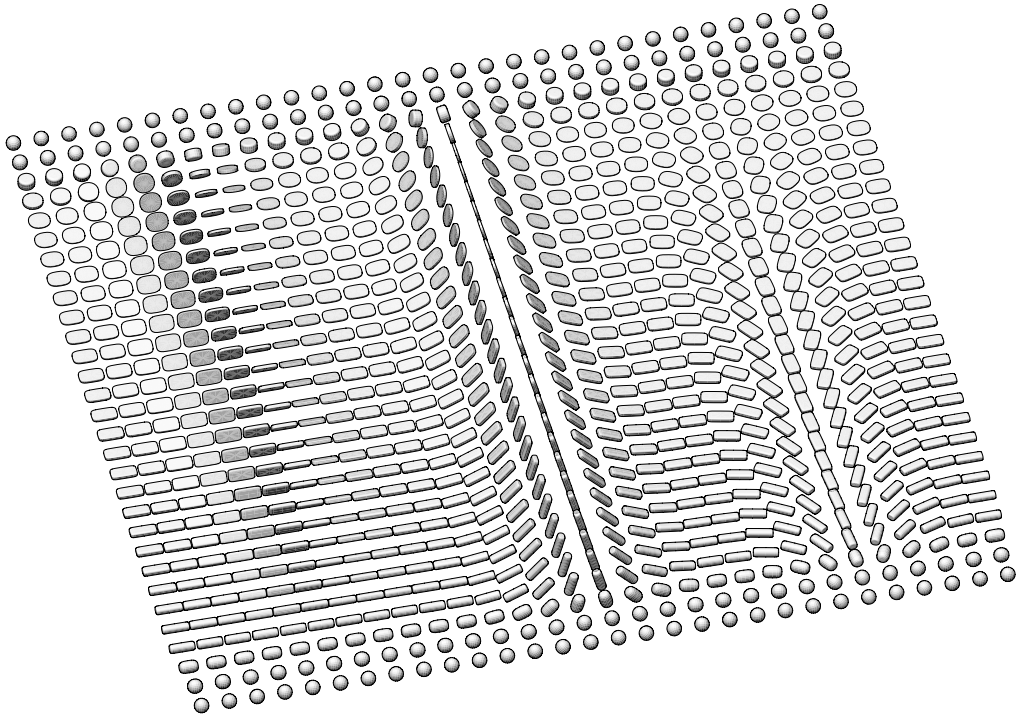
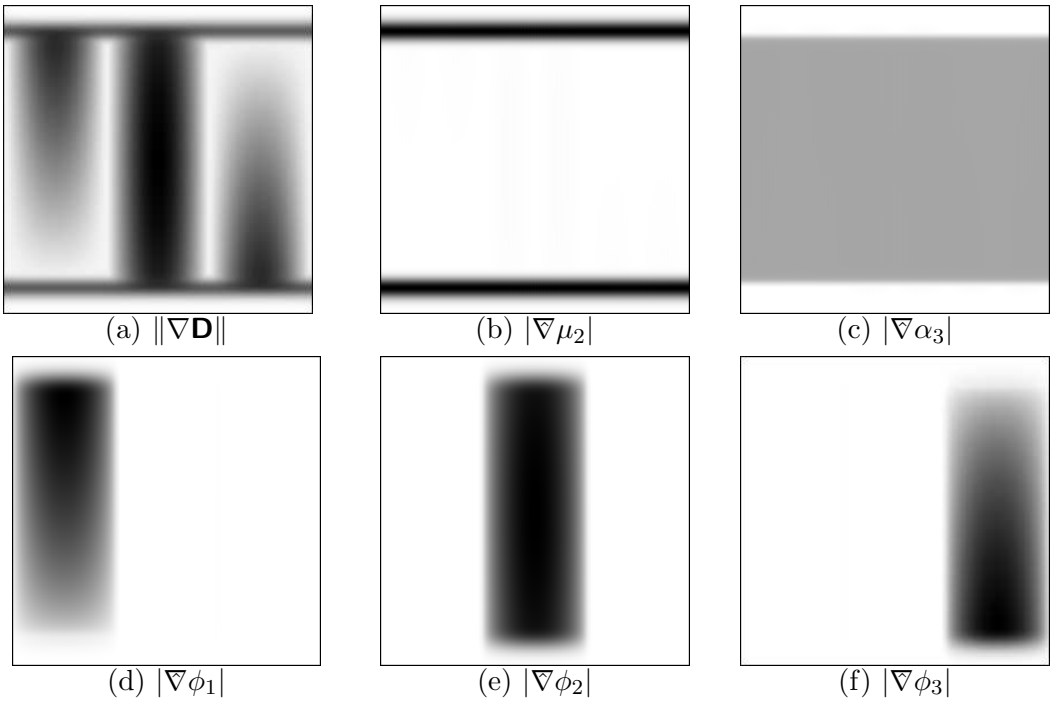


Figure 5.7. Display of synthetic tensor image with superquadric glyphs. Image was downsampled to 30×30 tensor samples for better visibility of individual glyphs.

Figure 5.8 shows different gradient magnitude measures applied to the synthetic image. Figure 5.8(a) shows how both shape and orientation changes contribute to $\|\nabla \mathbf{D}\|$, while Figures 5.8(b) and 5.8(c) show how the shape changes are decomposed into $|\nabla \mu_2|$ and $|\nabla \alpha_3|$. Figures 5.8(d), 5.8(e), and 5.8(f) show how $|\nabla \phi_1|$, $|\nabla \phi_2|$, and $|\nabla \phi_3|$ can respond in isolation to rotations around the eigenvectors $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$, respectively. These images also confirm that $|\nabla \phi_i| = 0$ when equality among eigenvalues creates axial symmetry around eigenvector $\mathbf{e}_i$.

Section 2.3.3 described the anatomical relevance of the eigenvalue moments μ_1, μ_2, and α_3, which gave the spatial gradients $\nabla \mu_1$, $\nabla \mu_2$, and $\nabla \alpha_3$ important roles in detecting the boundaries of specific anatomical features. The anatomical connection to tensor orientation change is also important for diffusion tensor field analysis. The eigenvector $\mathbf{e}_1$ of a diffusion tensor $\mathbf{D}$ associated with the largest eigenvalue is the direction with the highest apparent diffusion coefficient (λ_1), or the "principal diffusivity direction." The direction of $\mathbf{e}_1$ is of such basic importance that it is sometimes the only quantity extracted and analyzed from a measured diffusion tensor image. In nervous tissue, $\mathbf{e}_1$ is aligned with the direction of the white matter fiber tracts [153, 24, 14], which is the basis of fiber tracking algorithms [41, 20, 133, 206, 203].

The rate of change in $\mathbf{e}_1$ is therefore a useful indicator of tissue organization. Because the direction of $\mathbf{e}_1$ will be affected by rotations around $\mathbf{e}_2$ and $\mathbf{e}_3$, changes in $\mathbf{e}_1$ can be measured by $\sqrt{|\nabla \phi_2|^2 + |\nabla \phi_3|^2}$. Twisting around $\mathbf{e}_1$ (as indicated by $|\nabla \phi_1|$) is thereby ignored. Conversely, if the structural feature of interest is characterized by *planar* anisotropy, then its rate of orientation change is assessed by $\sqrt{|\nabla \phi_1|^2 + |\nabla \phi_2|^2}$.



(a) $\|\nabla \mathbf{D}\|$ (b) $|\nabla \mu_2|$ (c) $|\nabla \alpha_3|$

(d) $|\nabla \phi_1|$ (e) $|\nabla \phi_2|$ (f) $|\nabla \phi_3|$

Figure 5.8. Gradient magnitudes in synthetic tensor image. Darker grays indicate higher values. Quantization ranges for (d), (e), and (f) are equal, to permit comparison of the different magnitudes by their gray values.

Finally, we note that if the magnitude of orientation change can be acceptably measured with $\sqrt{|\nabla\phi_1|^2 + |\nabla\phi_2|^2 + |\nabla\phi_2|^3}$, then this quantity can be measured efficiently, without computing eigenvectors, even though orientation is defined in terms of eigenvectors. Because the normalized invariant gradients and rotation tangents form a basis for Sym_3, and because the spaces of shape and orientation changes are orthogonal, the magnitude of orientation change can be found from the magnitude of shape change, and the overall gradient magnitude, using the Pythagorean theorem:

$$\sqrt{|\nabla\phi_1|^2 + |\nabla\phi_2|^2 + |\nabla\phi_2|^3} = \sqrt{\|\nabla\mathbf{D}\|^2 - |\nabla\mu_1|^2 - |\nabla\mu_2|^2 - |\nabla\alpha_3|^2}\ .$$

5.9 Discussion

This chapter has described a mathematical approach to characterizing changes in tensor shape and changes in tensor orientation. Changes in shape are characterized in terms of the tensor-valued gradients of three tensor invariants: the mean, variance, and skewness of the eigenvalues. These are the same shape variables governing the geometry of the eigenvalue wheel (Section 2.3.2). Changes in orientation are characterized in terms of the derivatives of the group action ψ introduced in Section 2.1.7. Together, the normalized invariant gradients and rotation tangents form an orthonormal basis for all symmetric tensors. The tensor field gradient, a third-order tensor, can then be projected onto the basis vectors (by double contraction) to quantify the components of tensor change at a given point in field.

Projection by double contraction is only one of many possible ways of analyzing the tensor field gradient. Another approach could be based on the invariants computed from the third-order tensor defined by the gradient itself. That is, it may be that a few scalar attributes of the tensor gradient suffice to characterize certain feature boundaries, which may be faster to compute that the magnitudes of various tensor gradient projections. Analogies to previous work include characterizing the eigenvalues of the Jacobian of a vector field (the first derivative of the the vector field is a second-order tensor) [92], or the eigenvalues and eigenvectors of the Hessian of the scalar field (another second-order tensor), including its use to measure implicit surface curvature [131, 89, 26, 105].

As this chapter is more theoretical in nature than the others, the determination of its "success" rests primarily in the soundness of the mathematical exposition, as well as in the longer-term adoption of this framework by other researchers in tensor image processing. More immediately, Figures 5.4 and 5.8 demonstrate that the mathematics support a more nuanced approach to edge detection: three kinds of shape edges can be detected, as well as three kinds of orientation edges. To our knowledge this kind of specificity has not been described before in the context of tensor image analysis. More generally, any edge or feature boundary indicated by a high magnitude of $\nabla\mathbf{D}$ can, with the mathematics described, be broken down into its constituent shape and orientation changes. This is the basis of feature detection described in Chapter 7.

CHAPTER 6

DIRECT VOLUME RENDERING

Direct volume rendering is an established tool for visualizing scalar volume data, and is valued for its ability to display structures without relying on a prior segmentation or surface extraction step. This chapter extends direct volume rendering to visualization of diffusion tensor data. Recall that the idea behind the glyph-based methods of Chapter 3 was to display all the tensor information at a *discrete* set of restricted locations, selected either by a two-dimensional slice through the data, or by an anisotropy threshold in three dimensions. In contrast, the idea of the volume renderings presented here is to chose particular attributes of the data, such as those related to tensor shape (Section 2.3), and display the *continuous* structures formed by the variation of the attributes within the continuous tensor field. To create such visualizations, the basic ingredients of direct volume rendering– color, opacity, and shading– must be determined from the data. This chapter describes how transfer functions can be defined to specify colors and opacities, and describes a new method of shading based on analytical gradients of tensor invariants.

Note: Some of this material was previously published in [102], and the extended version of the same [103]. The unpublished material is Section 6.4, describing analytical shading based on the invariant gradients presented in Section 5.3.

6.1 Introduction and Related Work

The premise of direct volume rendering is that features of interest within a dataset can be shown on the basis of one or more locally measured data attributes. The *transfer function* then maps these attributes to renderable properties like color and opacity. Renderings are formed by casting rays through the volume, measuring these attributes, mapping them through the transfer function, and compositing the results along rays to determine pixel values. The canonical references for direct volume rendering describe how this method works in scalar fields [116, 56, 160]. Transfer functions for volume rendering are typically either univariate (mapping the scalar data value) [56] or bivariate (mapping data value and its gradient magnitude) [116, 101, 99]. Often, the visual effect is of a slightly "fuzzy" isosurface, rather than one explicitly formed of triangles, as by Marching Cubes [121].

There is a small but growing amount of work on volume rendering nonscalar data. An early example by Frühauf used raycasting and directional colormaps to display large-scale patters in vector fields [66]. When rendering multiscalar fields, such as those from multi-valued MRI or photographic cryosection, transfer functions have been implemented as a two-dimensional lookup table indexed by the constituent scalar values [109, 108]. Other work has converted vector field visualization into a scalar volume rendering problem, such as the Flow Volume method of Max et al. [125]. Similarly, previous work in using volume visualization of DT-MRI data has rasterized fiber tractography paths to a scalar

volume, which is rendered along with a precomputed tensor attributes such as mean bulk diffusivity [193].

Section 2.3.3 described the role that tensor shape has in describing the underlying anatomical structure, so we often use shape information from tensor invariants as the basis for our transfer functions. At every step along each ray cast, the tensor components are reconstructed by separable convolution (Section 2.1.8), from which the required invariants are computed. In volume rendering scalar fields, shading plays the role clarifying the shape of structures, and is often based on the gradient of the scalar field. In tensor volume rendering, shading is more complicated, as there is no single gradient, so we compute the the spatial invariant gradients based on the gradient of the tensor field, a third-order tensor (Section 5.3).

6.2 Barycentric Color and Opacity Maps

One method for assigning colors and opacities in the volume rendering is based on the barycentric space of tensor shape presented in Section 2.3.4. Because of the intuitive nature of this space, it is fairly easy to design transfer functions which reflect interest in particular ranges of tensor shape.

Figure 6.1 demonstrates some barycentric opacity maps. Each opacity map is depicted by gray-scale representation: brighter regions in the triangle correspond to higher opacity assignment. For the purposes of this figure, the effect of the opacity map is demonstrated by applying the map to the the tensor dataset, resulting in a *scalar* volume of opacity values. This new scalar volume is visualized with a *linear* opacity function, and shaded according to the gradient of opacity values. One can see that appropriately chosen opacity functions allow one to see the form of structures in the dataset that have one predominant type of anisotropy.

Because of its expressive power, the barycentric space also makes sense as the domain of the color function, which assigns color to each sample point in the volume rendering according to its anisotropy. Most importantly, different kinds of anisotropy receiving equal opacity can be disambiguated by assigning different colors. Also, to the extent that various classes of white-matter tissue are found to have a characteristic anisotropy throughout the volume, they can be color-coded with an appropriate barycentric color map. Volume renderings made with both barycentric opacity and color maps allow an extra dimension of information about the diffusion tensor to be represented in the volume rendering. Figure 6.2 shows two examples of these.

6.3 Lit-Tensors

Streamlines used in vector visualization are sometimes hard to interpret because they lack the shading cues which we are accustomed to seeing on surfaces. However, illuminated streamlines ("lit-lines") have shading and highlights which give information about their direction and curvature, creating the appearance of shiny filaments [209, 175]. In the case of diffusion tensor visualization, we have made tensors opaque and colored based on their anisotropy. The shading used in Figures 6.1 and 6.2 is based on the gradient of the precomputed opacity field (essentially preclassified volume rendering [119]). Here, we describe a shading technique termed *lit-tensors* which seeks to indicate the type and orientation of anisotropy.

Lit-tensors are designed according to following certain constraints:

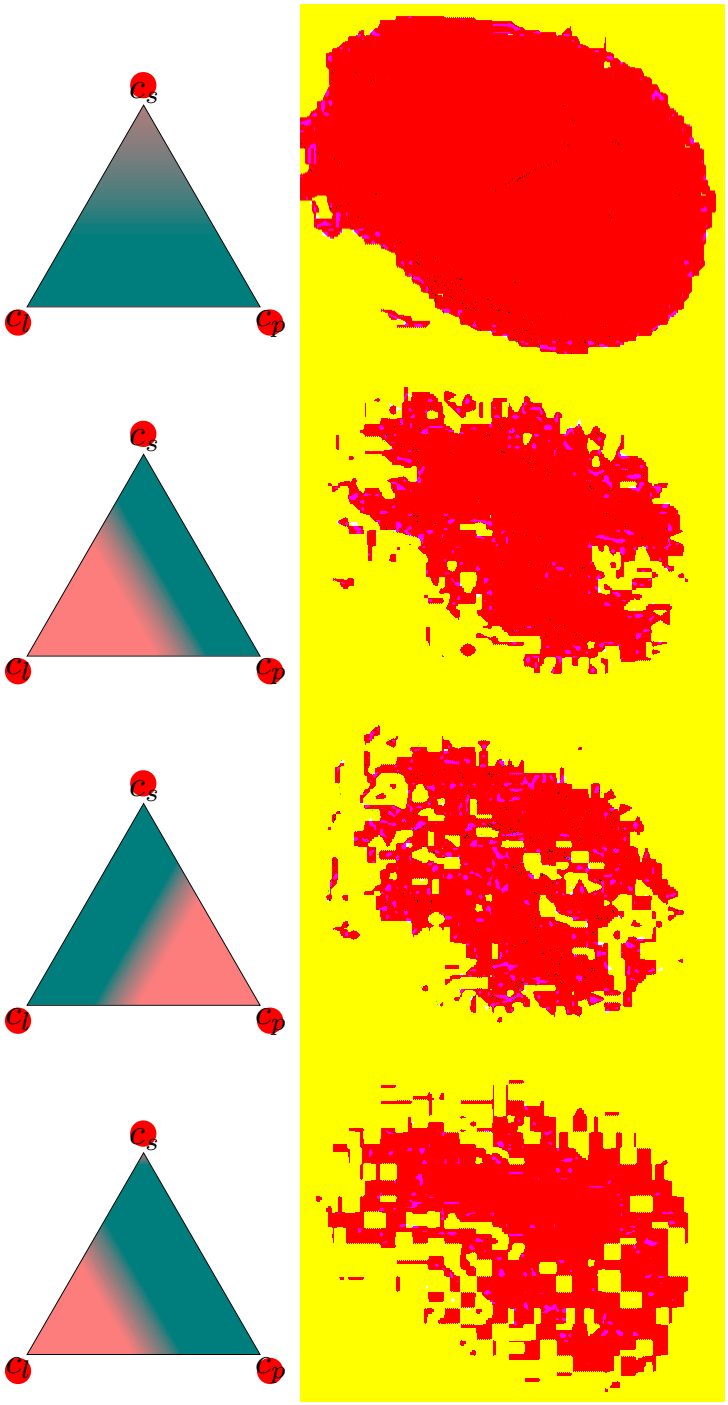


Figure 6.1. Examples of barycentric opacity maps and resulting volumes.

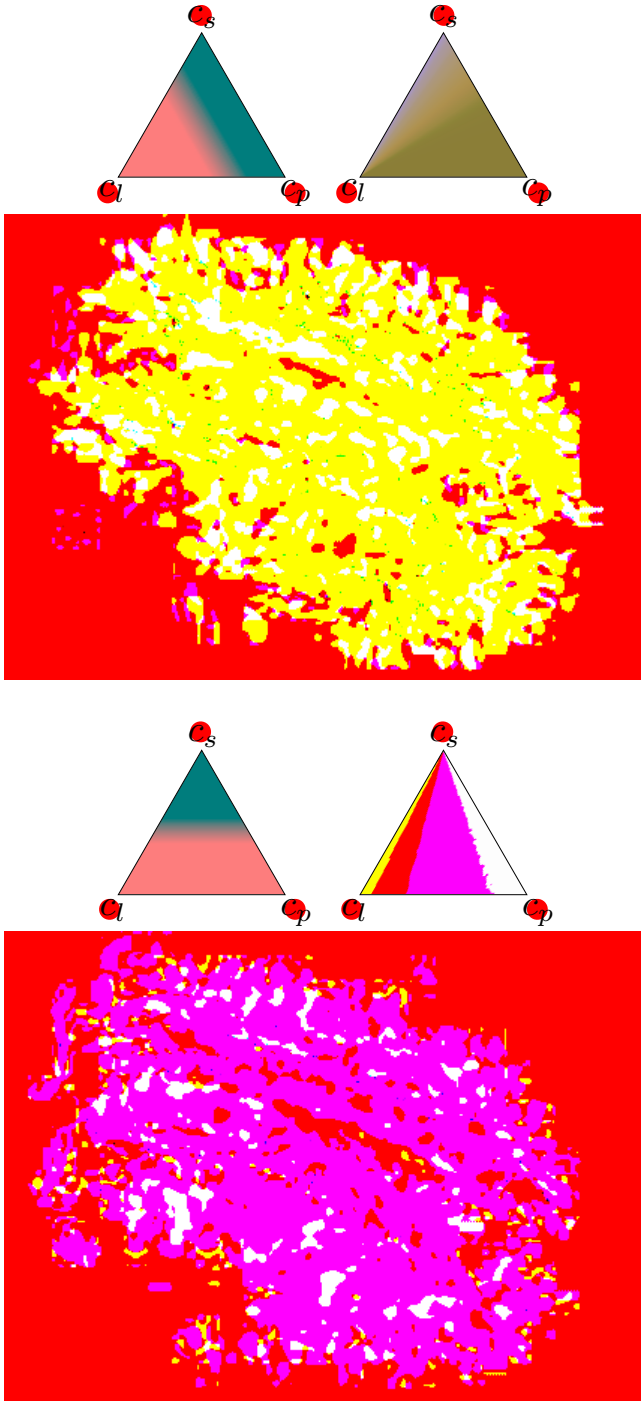


Figure 6.2. Examples of barycentric color maps and resulting renderings.

1. In regions of complete linear anisotropy, the lighting model should be identical to that of illuminated streamlines. Complete linear anisotropy means movement by diffusion is constrained to one dimension, so it is sensible for the lighting model to degenerate to one already developed for vector fields.

2. In regions of complete planar anisotropy, the lighting model should be the same as with traditional surface rendering. The obvious choice for the "surface normal" for a planar anisotropic tensor is the third eigenvector, perpendicular to the plane formed by the span of the first two eigenvectors (associated with the largest two eigenvalues).

3. There has to be a smooth interpolation between these two extremes. Because tensor data can exhibit a wide variety of anisotropies, allowing small variations in anisotropy to lead to large changes in shading will probably create a more confusing image.

This can be seen as a problem of how to interpolate illumination between different codimensions. The codimension of the diffusion tensor's representative ellipsoid is two in the linear anisotropy case, and one with planar anisotropy. Previous work [8] has rigorously developed illumination methods for general manifold dimension and codimension, but did not cover cases part-way between different codimensions. Unlike that work, no claim to physical accuracy or plausibility is made for the model presented here; it is just one simple way of satisfying the constraints above.

We take as our starting point the Blinn-Phong lighting model [34]:

$$\begin{aligned} I &= I_{\text{ambient}} + I_{\text{diffuse}} + I_{\text{specular}} \\ &= k_{\text{a}} A_\lambda O_\lambda + I_\lambda (k_{\text{d}} O_\lambda \mathbf{l} \cdot \mathbf{n} + k_{\text{s}} (\mathbf{h} \cdot \mathbf{n})^s) \end{aligned} \tag{6.1}$$

k_{a}, k_{d}, and k_{s} control the contributions of ambient, diffuse, and specular reflection to the final image. Following Foley et al. [63], we add the subscript λ to those variables which vary according to color. For example, there are separate values I_{r}, I_{g}, I_{b}, for the red, green, and blue components of the directional light source. The ambient light color is A_λ. Instead of representing the intrinsic object color with different k_{a} and k_{d} for red, green, and blue, we use O_λ for object color and keep k_{a} and k_{d} as separate controls. In our case, the intrinsic object color is determined by any of the methods described in this paper (barycentric maps, hue-balls, or reaction-diffusion textures). $\mathbf{l}$ is the vector pointing towards the directional light source, $\mathbf{v}$ points towards the eye, and $\mathbf{n}$ is the surface normal. We following the convention for computing the specular term introduced by Blinn [34], whereby instead of using the reflection $\mathbf{r}$ of $\mathbf{l}$ across $\mathbf{n}$, we use the "half-way" vector $\mathbf{h}$, computed as the normalized average of $\mathbf{l}$ and $\mathbf{v}$. The size of the specular highlight is controlled by s, the shininess exponent.

Because a streamline is one-dimensional, at any given point along it there is an infinite set of normals, all perpendicular to the tangent direction $\mathbf{t}$, radiating outwards in a circle. If naively using Equation 6.1 to illuminate a streamline, one must find the normal which is in the plane spanned by $\mathbf{l}$ and $\mathbf{t}$ to evaluate $\mathbf{l} \cdot \mathbf{n}$. Similarly, another specific normal must be found to evaluate $\mathbf{h} \cdot \mathbf{n}$. The insight which makes lit-lines simple is that one does not need to actually find a specific normal in order to evaluate a dot product with it. With the Pythagorean theorem, the dot product with $\mathbf{n}$ can be expressed in terms of the tangent $\mathbf{t}$:

$$\mathbf{u} \cdot \mathbf{n} = \sqrt{1 - (\mathbf{u} \cdot \mathbf{t})^2} \tag{6.2}$$

where $\mathbf{u}$ is either $\mathbf{l}$ or $\mathbf{h}$, for the diffuse and specular terms, respectively.

The relevant property of Equation 6.2 is that the lighting calculation depends on a tangent vector $\mathbf{t}$ that gives the object's direction, instead of its surface normal $\mathbf{n}$. The direction and orientation of a diffusion tensor is determined by not one, but two vectors: the first and second eigenvectors.[1] Both of these could be interpreted as tangents, but their relative importance is determined by the magnitudes of the corresponding eigenvalues. To control the relative importance of the first two eigenvectors in determining the tensor's orientation, we introduce a parameter c_θ which characterizes *anisotropy type*. Assuming that the eigenvalues are ordered $\lambda_1 \geq \lambda_2 \geq \lambda_3$, we define

$$c_\theta = \frac{\pi}{2}\frac{c_p}{c_a} = \frac{\pi(\lambda_2 - \lambda_3)}{\lambda_1 + \lambda_2 - 2\lambda_3} \ . \tag{6.3}$$

As anisotropy varies from completely linear ($c_l = 1$; $c_p = 0$) to completely planar ($c_l = 0$; $c_p = 1$), c_θ varies from 0 to $\frac{\pi}{2}$. The role of c_θ is to control how much the second eigenvector contributes to the lighting of the diffusion tensor. In the linear case, only the first eigenvector determines the tensor orientation, and in the planar case, both the first and second eigenvectors matter equally.

The expression to be used in lieu of dot products with $\mathbf{n}$ is:

$$\text{“}\mathbf{u}\cdot\mathbf{n}\text{”} = \sqrt{1 - (\mathbf{u}\cdot\mathbf{e}_1)^2 - (\mathbf{u}\cdot\mathbf{e}_2\ \sin(c_\theta))^2} \ . \tag{6.4}$$

In the case of linear anisotropy, $\sin(c_\theta) = \sin(0) = 0$, so the contribution from $\mathbf{e}_2$ vanishes, and the expression reduces to the formula for lit-lines (Equation 6.2), with the principal eigenvector $\mathbf{e}_1$ taking the role of the tangent $\mathbf{t}$. This is appropriate, since in linear anisotropy, the principal eigenvector points in the direction of movement, as does a streamline's tangent vector.

In planar anisotropy, $\sin(c_\theta) = \sin(\frac{\pi}{2}) = 1$, and the contributions of the two dot products are equal. This means that for any other vector $\mathbf{w}$ such that

$$(\mathbf{w}\cdot\mathbf{e}_1)^2 + (\mathbf{w}\cdot\mathbf{e}_2)^2 = (\mathbf{u}\cdot\mathbf{e}_1)^2 + (\mathbf{u}\cdot\mathbf{e}_2)^2 \ , \tag{6.5}$$

Equation 6.4 will have the same value. Therefore, in planar anisotropy the lighting model is rotationally symmetric around $\mathbf{e}_3$. Rotational symmetry in this case is actually an important feature of the lighting model. In planar anisotropy, the diffusion tensor ellipsoid degenerates to a disc, and *any* vector in the plane spanned by the disc is an eigenvector. Because of this numerical instability, the calculated directions of the first and second eigenvectors will be essentially random. The illumination should not be sensitive to this arbitrary orientation, and should only be a function of the third eigenvector. In fact, one can use the Pythagorean theorem to show that if $c_\theta = \frac{\pi}{2}$, Equation 6.4 gives an exact formula for $\mathbf{u}\cdot\mathbf{e}_3$. Interpreting both $\mathbf{e}_1$ and $\mathbf{e}_2$ as surface tangents, then the surface normal $\mathbf{n}$ is aligned along $\mathbf{e}_3$. Therefore the model contains standard surface shading as a special case.

To demonstrate lit-tensors, Figure 6.3 shows nine different synthetic diffusion tensor datasets which were direct volume rendered with a fixed viewpoint and light. The anisotropy index c_a of the sphere is also constant in every case, but c_θ is changing. The

[1] Because the eigenvectors always form an orthogonal basis, and because we are adopting two-sided lighting, the third eigenvector does not contribute any additional information.

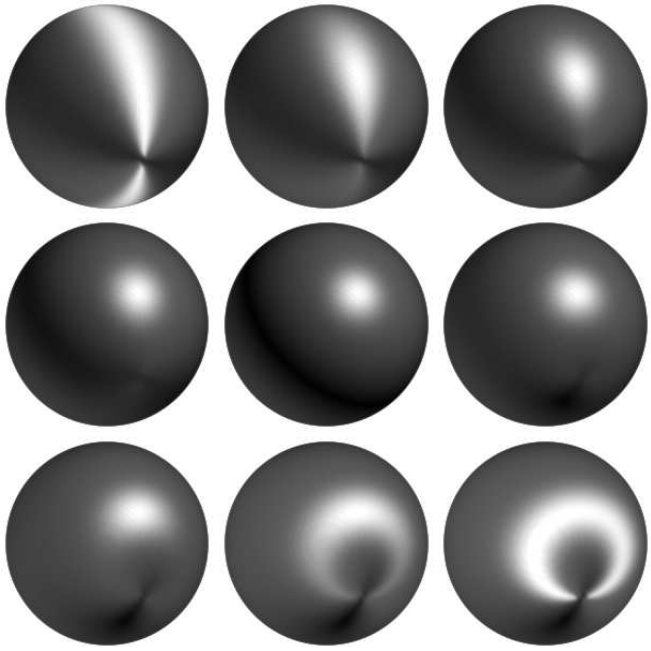


Figure 6.3. Sequence of volumes of differing anisotropy, rendered with lit-tensors. Anisotropy varies gradually between the nine volumes, going in scanline order.

dataset in the upper left has complete linear anisotropy in a concentric circular pattern (along lines of latitude). The dataset in the middle has complete planar anisotropy (and hence looks just like a standard surface rendering). The dataset in the lower right has complete linear anisotropy along lines of longitude, going from pole to pole. The images provide a convincing sense of surface anisotropy, which is not a typical trait in direct volume renderings.

The spheres shown in Figure 6.3 have well-behaved anisotropy in the following sense: the changes in the orientation of anisotropy are directly correlated to the changes in the orientation of the sphere's surface normal. Experience has shown that measured diffusion tensor data is generally not so well-behaved, so that still images created using lit-tensors tend to be confusing[2]. The underlying problem is that lit-tensors were designed to indicate anisotropy type and direction, not the shape of the structure made opaque by the opacity function. On a complex structure with significant self occlusion, the lack of surface shape cues can lead to a rather ambiguous, water-color effect, as seen in Figure 6.4.

Our current solution to this problem is to perform a separate (and significantly simpler) shading calculation, using standard Phong shading with the normalized gradient of opacity serving as the surface normal. This is accomplished by a two-step preprocess: the opacity at each data point is determined, and then the gradient of the opacity field is calculated. The normalized negative gradient of opacity is stored at each sample point, and during volume rendering, these directions are interpolated to determine a surface normal at every point along the ray. The interpolated surface normal is used in the Phong

[2]Animations that vary viewpoint location can disambiguate surface shape while enhancing the effect of lit-tensors by showing motion of the specular highlights.

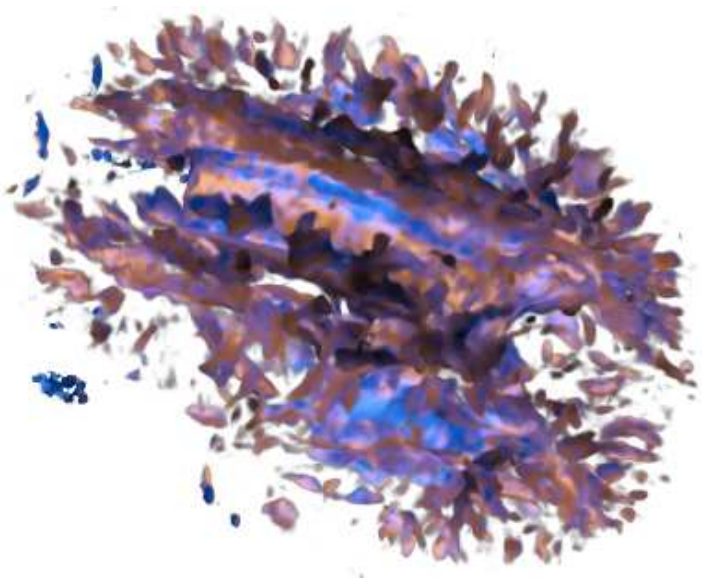


Figure 6.4. Lit-tensor model shading on the whole brain, with the same opacity and light direction as first image in Figure 6.2.

shading equation. This results in an image in which shading is entirely determined by the opacity assignment and the shape of the opaque structures selected by it.

However, it is also possible to arbitrarily mix the results of lit-tensor shading and opacity gradient shading, as shown in Figure 6.5. Both shading calculations are performed, and then the results are mixed on a per-voxel basis by a user-defined parameter. This sort of mixing is quite different than varying the anisotropy type as was done in Figure 6.3. Instead of one specular highlight changing shape gradually, there are two different specular highlights which cross-blend.

The range of possibilities illustrated by Figure 6.5 demonstrates an important difference between scalar and tensor volume rendering. In scalar volume rendering, opacity is nearly always determined as a function of the (scalar) data value, hence the opacity gradient is always aligned with the gradient of original data value. This means that the data value gradient can be computed only once per dataset and used to shade the output of any opacity function. Unfortunately, such a preprocess is not possible with tensor data under barycentric opacity maps, as the domain of the opacity function is a multidimensional space which varies nonlinearly with the tensor matrix component values. On the other hand, given the overall computational expense of tensor volume rendering, we have found the cost of having to compute the opacity gradient once per opacity function to be acceptable.

6.4 Volume Rendering with Tensor Gradients

A significant drawback to the method of rendering and shading presented in Section 6.2 is the use of *preclassification*: the tensor volume was mapped through the opacity function to create a scalar opacity field, which was then shaded as in standard scalar volume rendering (although with color assigned by a transfer function of interpolated tensor values). As with any use of preclassification, especially with transfer functions with

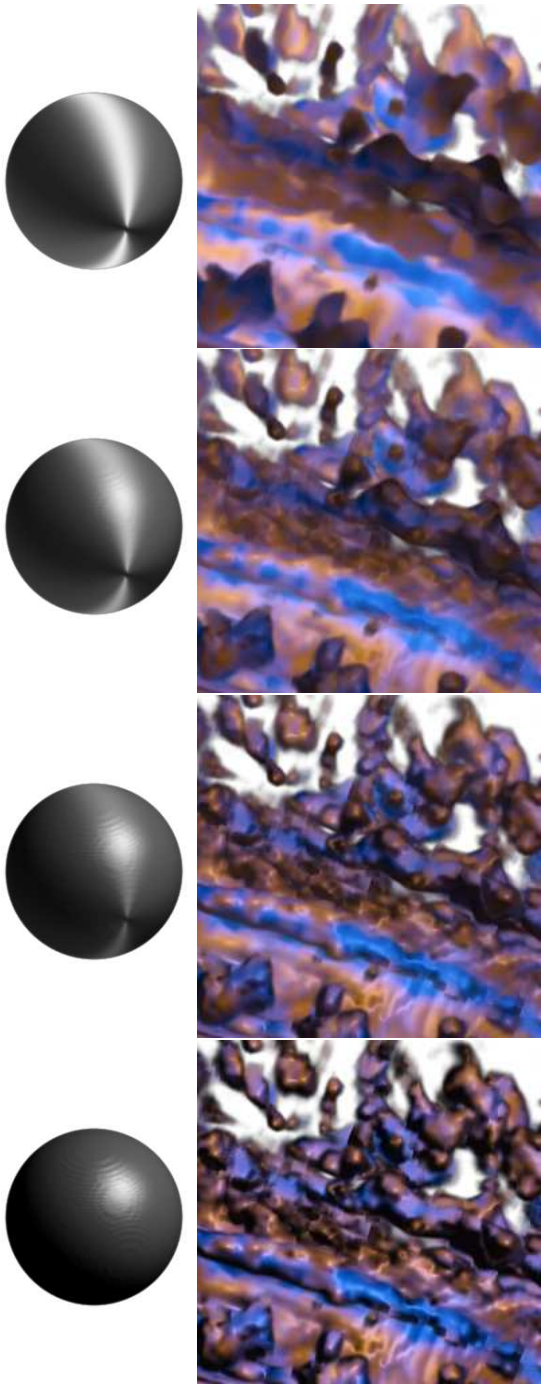


Figure 6.5. Mixing between lit-tensor and opacity gradient shading, for a synthetic sphere (left), and a portion of brain data (right). Going from top to bottom, the contribution of lit-tensors is 1.00, 0.66, 0.33, and 0.00.

sharp transitions, this creates errors in rendering small features [119, 108].

The mathematical results of Chapter 5 allow us to improve upon this. In particular, the ability to project the tensor gradient $\nabla \mathbf{D}$ along some invariant gradient $\boldsymbol{\nabla} J$ allows us to compute the *analytical* spatial gradient of J within the tensor field, once a particular set of reconstruction and derivative kernels has been chosen. Here, the invariant J will be an anisotropy measure such as fractional anisotropy, FA. This work represents the first use of analytical shading in nonscalar volume rendering. The importance of this is two-fold. First, the limited resolution of DT-MRI scans motivates special care in rendering the small features which may be represented by only a few voxels, which preclassification fails to do. Second, the extensive use of anisotropy thresholds in other tensor visualization methods, such as determining which glyphs contribute to a three-dimensional rendering (Section 3.1.5) or the termination criteria for fiber tractography (Section 4.2), suggests that anisotropy isosurfaces are fundamental structures that can be visualized in their own right. Proper shading helps us better understand the shape of anisotropy isosurfaces.

Recall from Equation 2.35 that fractional anisotropy FA can be defined as:

$$\mathrm{FA} = \frac{3}{\sqrt{2}} \sqrt{\frac{\mu_2}{J_4}} \; .$$

Shading in volume rendering is based on the normalized gradient of the underlying scalar, so it does not matter if we take the gradient of FA or FA^2:

$$\boldsymbol{\nabla} (\mathrm{FA}^2) = \frac{9}{2} \frac{J_4 \boldsymbol{\nabla} \mu_2 - \mu_2 \boldsymbol{\nabla} J_4}{J_4^2} \; .$$

Again, the overall magnitude of this tensor is not important, so to define the spatial gradient of FA as the basis of a surface normal for shading, we may say (following the definition of ∇J in Equation 5.19)

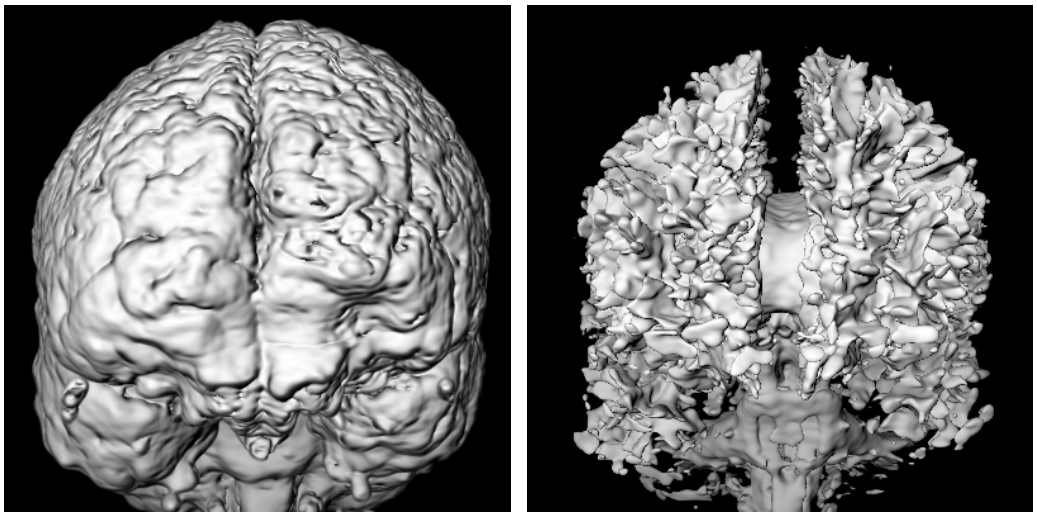
$$\nabla \mathrm{FA} \approx (J_4 \boldsymbol{\nabla} \mu_2 - \mu_2 \boldsymbol{\nabla} J_4) : \nabla \mathbf{D} \; .$$

Equations 5.9 and 5.4 can be used to define $\nabla \mathrm{FA}$ in terms of the principal invariant gradients. Then, shading for volume rendering may be based on a "surface normal":

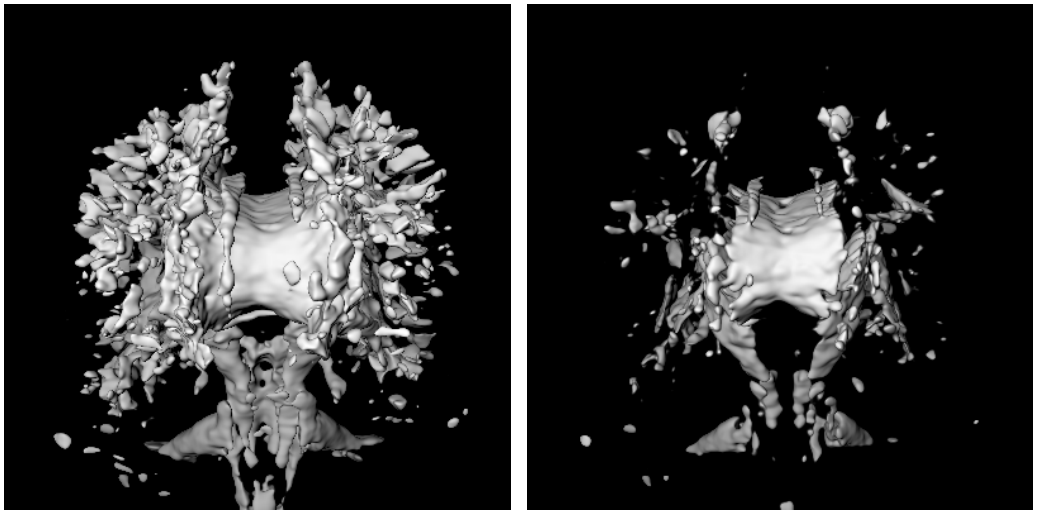
$$\mathbf{n}_{\mathrm{FA}} = -\nabla \mathrm{FA} / |\nabla \mathrm{FA}| \; .$$

Figure 6.6 shows four volume renderings of FA isosurfaces with Lambertian shading [170] based on $\mathbf{n}_{\mathrm{FA}}$. The isosurface is rendered with what is nearly an opacity *step function*: except for a narrow transition, opacity is 0.0 or 1.0 depending on whether FA is below or above the indicated threshold. Also, a high sampling rate along the volume rendering rays (roughly five samples per voxel) avoids aliasing artifacts at the surface. The reconstruction kernel is the cubic B-spline (Figure 2.2(c)), and the derivative kernel is the B-spline derivative (Figure 2.3(c)). The clarity of these images, due in part to the analytical shading of small features, is a significant improvement over Figure 6.1. The drawback, however, is in having a univariate (instead of bivariate) opacity function.

To further demonstrate that volume rendered anisotropy isosurfaces are informative visualizations of major white matter structure, Figures 6.7 and 6.8 show lateral and medial views, respectively, of a volume rendered isosurface of FA, with the RGB color-coding (Equation 3.7) to help identify the major white matter structures, which are labeled in the second half of each figure. As with Figure 6.6, the cubic B-Spline and its derivative

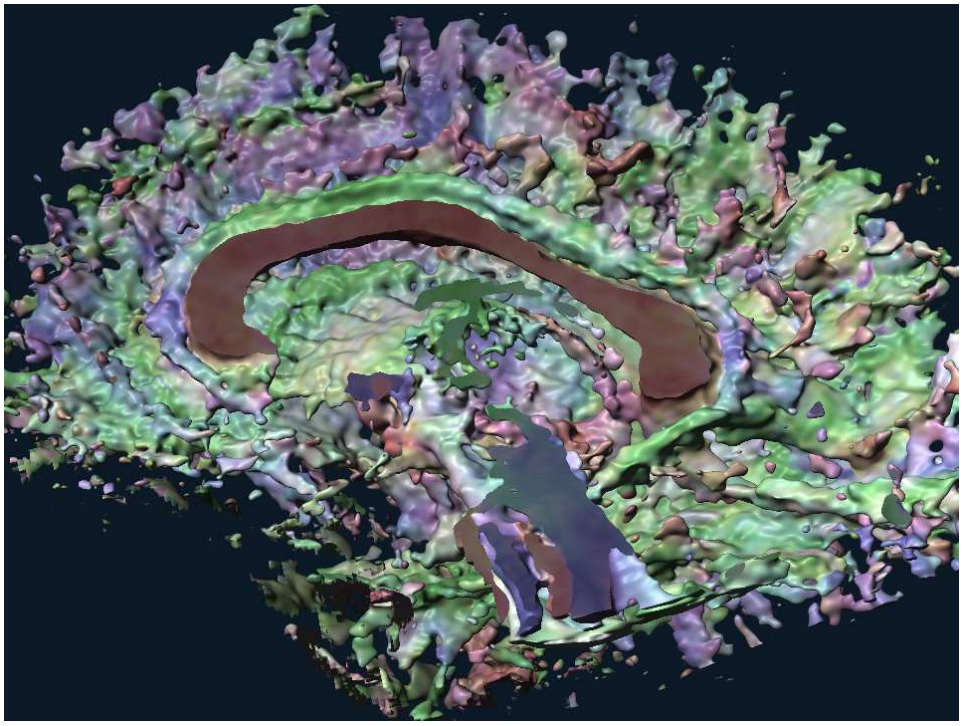


FA = 0.0 FA = 0.3

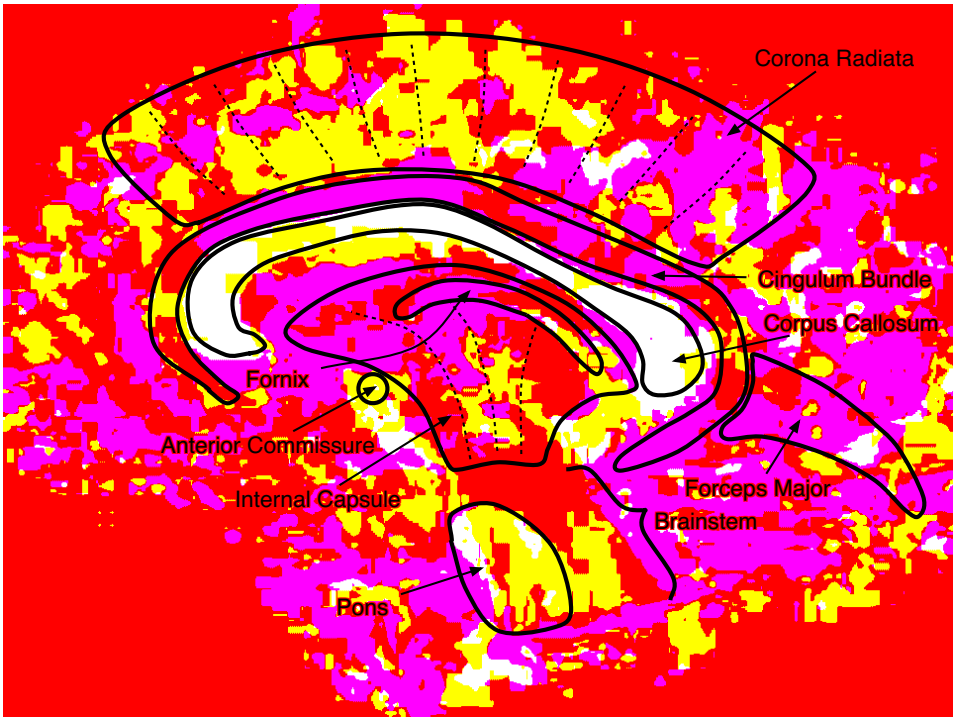


FA = 0.5 FA = 0.65

Figure 6.6. Volume-rendered isosurfaces at a range of FA values show basic 3D structure of white matter in a DTI brain scan.

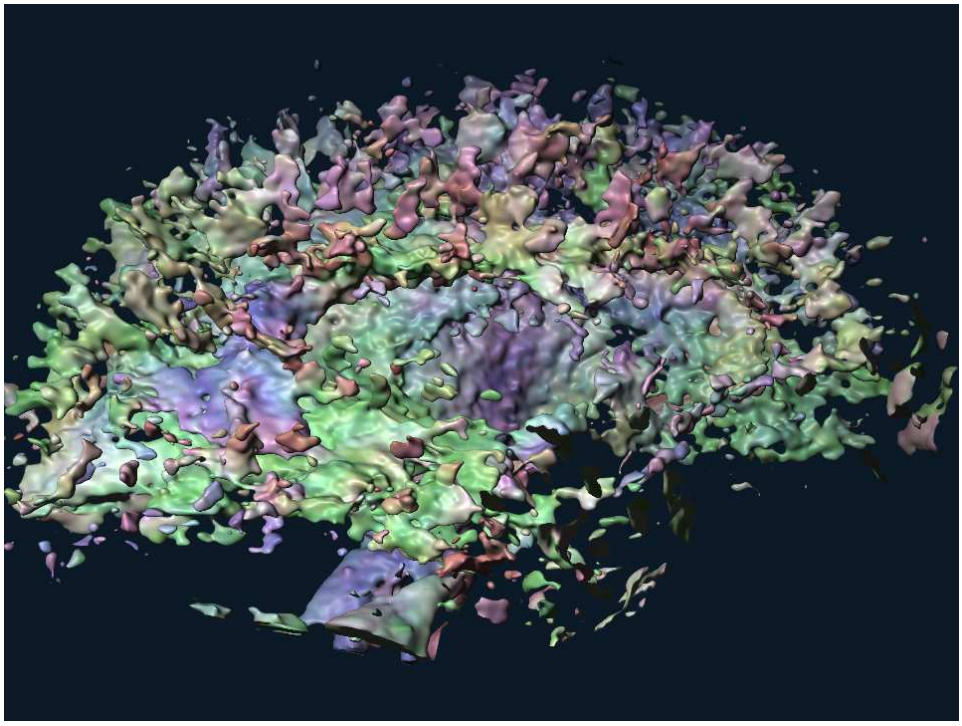


(a) Volume Rendered Isosurface

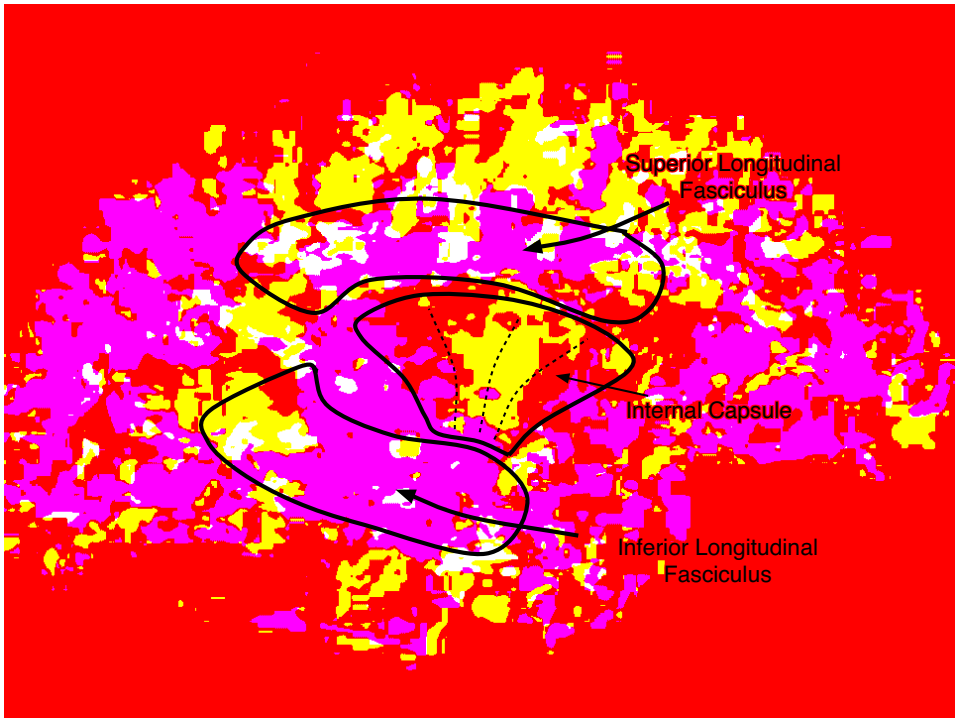


(b) Anatomical Anotations

Figure 6.7. Volume renderings of half a DT-MRI brain scan, with isovalue FA = 0.4, from a medial and slightly inferior view.



(a) Volume Rendered Isosurface



(b) Anatomical Anotations

Figure 6.8. Volume renderings of half a DT-MRI brain scan, with isovalue FA = 0.5, from a lateral and slightly inferior view.

are used for value and derivative reconstruction, respectively. The smooth shading and specular highlights assist in the shape perception of the isosurface. The RGB coloring is the usual colormap of the principal eigenvector (Equation 3.7). These direct volume renderings complement Figure 3.10 (the three-dimensional glyph renderings of the same dataset) because the volume renderings show the features and texture in the continuous FA isosurface that defines the region of discrete sampling visualized by the glyphs.

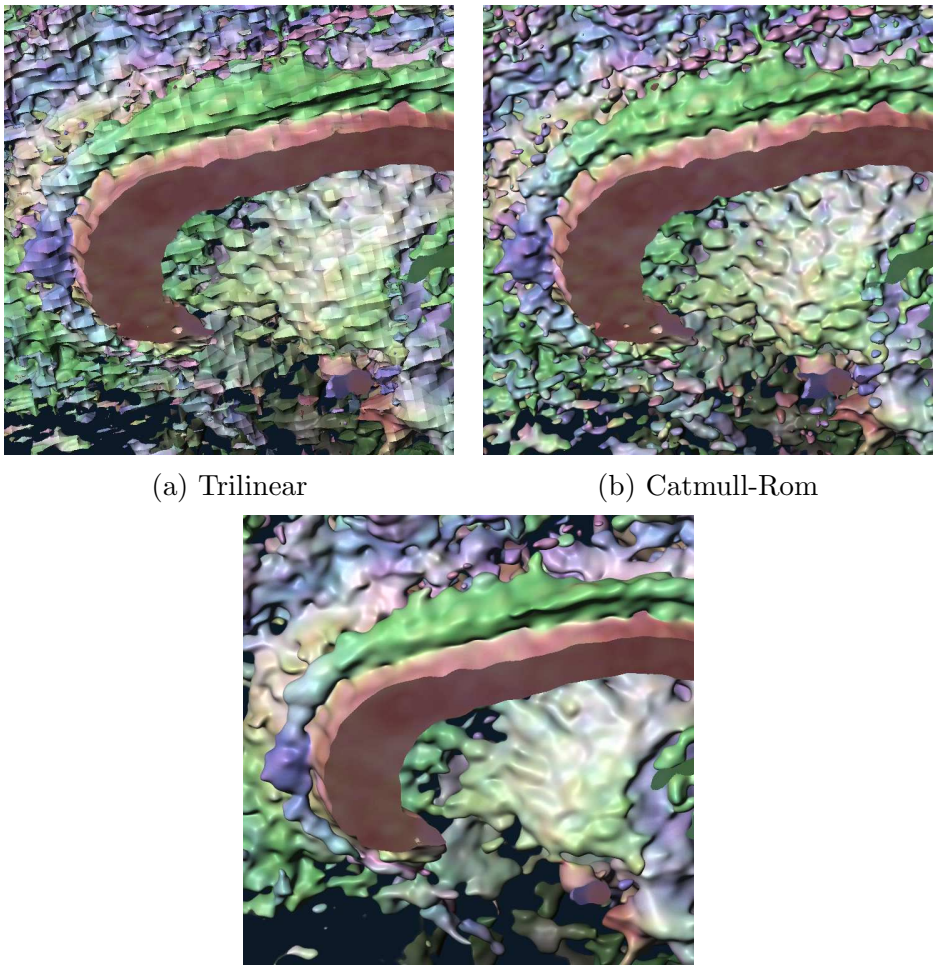
Finally, Figure 6.9 demonstrates the effect of kernel choice on the resulting anisotropy isosurface. Different kernel choices allow different aspects of the field to be seen. Trilinear interpolation is not C^1 continuous, so the sharp edges in Figure 6.9(a) indicate the underlying data resolution. The Catmull-Rom kernel (graphed in Figure 2.2(b)) is a C^1 kernel which interpolates, so any noise in the data is manifest as a more bumpy surface. The B-spline curve is C^2, and not interpolating, so the corresponding isosurface is significantly smoother, allowing the ridged structure of the internal capsule (underneath the red corpus callosum) to be seen clearly.

6.5 Discussion

This chapter has described the generalization of direct volume rendering to diffusion tensor data. The approach of Section 6.4 in particular represents the first time that diffusion tensors have been volume rendered directly from the tensor field, rather than by preprocess conversion to a scalar opacity field (as in Section 6.2). The success of the method is supported by the ability to discern the major white matter structures in, for example, Figure 6.7. The numerical accuracy of the rendering is assured by the small step size (about a fifth the width of a voxel) used in the ray integration, and by the consistent reliance on separable convolution to reconstruct the tensor values and their derivatives. All field information is reconstructed on-the-fly, at each sample point, by convolution, which enables analytical shading based on the gradient of fractional anisotropy.

One aspect of diffusion tensor volume rendering that has not been addressed is the importance of differentiability of the tensor invariants used to define the transfer function. A compelling reason for using fraction anisotropy FA for the demonstrations of analytical shading is that FA is defined in terms of invariants μ_2 and J_4, both of which are differentiable everywhere in Sym_3. This is not true of c_l and c_p, however, because of the eigenvalue sorting inherent in their definition. On the other hand, the barycentric shape space (based on c_l and c_p) provides a vastly more flexible domain for opacity functions. Determining how to combine with the flexibility of barycentric shape space with the differentiability of metrics like FA is an area of future work.

Given the amount of work involved in computing a volume rendering of a tensor field, the term "direct volume rendering" seems to be a misnomer. Multiple convolutions are required to estimate, at each sample point, the tensor components and their derivatives (Section 2.1.8), often the eigenvalues and eigenvectors are computed (Sections 2.3.1 and 3.1.1), and then colors and opacities were determined based on these values. For this reason, tensor volume rendering is currently not interactive. The computational speed and flexibility of modern graphics hardware is increasing at such a rate, however, that this should soon be feasible. For example, volume renderings from two-dimensional transfer functions took many minutes to compute (in software) when introduced in 1988 [116], but can now be rendered at multiple frames per second with commodity graphics hardware [108]. Whether applied to scalar or tensor data, the intrinsically data-parallel nature of direct volume rendering means it is well suited to streaming-based processors found on modern graphics hardware [62].



(a) Trilinear (b) Catmull-Rom

(c) B-Spline

Figure 6.9. Renderings of anterior corpus callosum and cingulum bundle, showing the effect of different kernel choices

CHAPTER 7

ORTHOGONAL TRACT DELINEATION

This brief chapter applies the theoretical framework of Chapter 5 to a particular task of feature analysis in diffusion tensor fields: detecting proximity between two regions of orthogonally oriented linear anisotropy. This configuration was discussed in Section 2.3.3 as source of planar anisotropy due to partial voluming. As noted there, at least two locations in the brain are characterized by this configuration: the anterior-posterior direction of the cingulum bundles directly superior to to the left-right direction of the corpus callosum, and the right-left trans-pontine tracts directly ventral to the inferior-superior corticospinal tracts in the brainstem. The first of these configurations was illustrated in Figure 3.10: the cingulum bundle is represented by the path of green/blue glyphs, and the corpus callosum is represented by the orthogonal oriented red glyphs beneath.

In the context of feature-preserving filtering of diffusion tensor fields, these are anatomical locations where it is especially important to avoid smoothing or blurring, since this would imply additional mixing or communication between the distinct tracts. Unfortunately, this configuration is not easy to characterize in terms of shape or orientation alone: anisotropy is high within both tracts, and the orientation change appears to be discrete, not continuous. The mere presence of planar anisotropy is not a sufficient indicator, because (as was noted in Section 2.3.3) other regions of the brain such as the superior longitudinal fasciculus have high planar anisotropy without touching an orthogonally oriented tract. This chapter presents a novel method of localizing configurations of orthogonal tract proximity based on analysis of the tensor field gradient. We also confirm that the gradients of the eigenvalue mean μ_1 can detect the boundary between gray matter and cerebro-spinal fluid (CSF), and that (to a lesser extent) the gradient of eigenvalue variance μ_2 can detect the boundary of white and gray matter.

7.1 Analysis of Tensor Gradient Components

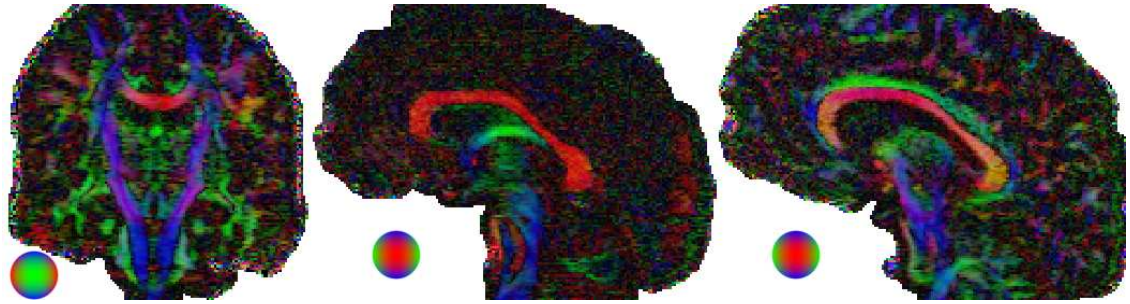
Figure 7.1 shows three slices of a DT-MRI volume used for this analysis, both with the standard RGB colormap (Equation 3.7). On the same slices, Figure 7.2 shows $\|\nabla \mathbf{D}\|$, the overall magnitude of the spatial gradient of the tensor field $\mathbf{D}$ (Equation 5.18). The corpus callosum is the upward red arc in the coronal image, and the downward red arc in the sagittal images. The cingulum bundles are the green/blue structures directly above the corpus callosum in the coronal view, and wrapped around the corpus callosum in the second sagittal view. In the first sagittal image are visible the (red) trans-pontine tracts directly ventral to the inferior-superior (blue) corticospinal tracts in the brainstem. This image also shows the anterior commissure as a small left-right pathway, visible as a red dot below the corpus callosum. The feature detection task of this section is essentially to isolate those regions where tracts with different primary colors in Figure 7.1 are physically touching.

Aside from the RGB colormap images which show the slice at original resolution, all the other images in this section (images of gradient magnitudes) show measurements on a *continuous* tensor field created by convolution with the B-spline filter (Figure 2.2(c)), and up-sampled by a factor of four. The small amount of smoothing helps deal with noise in the original data. Also, as in Figures 5.4 and 5.8, the grayscale mapping of the gradient magnitude images is inverted for better visibility of faint features. To assist in qualitative comparison, the quantization ranges and grayscale mapping is identical for all images of gradient magnitudes.

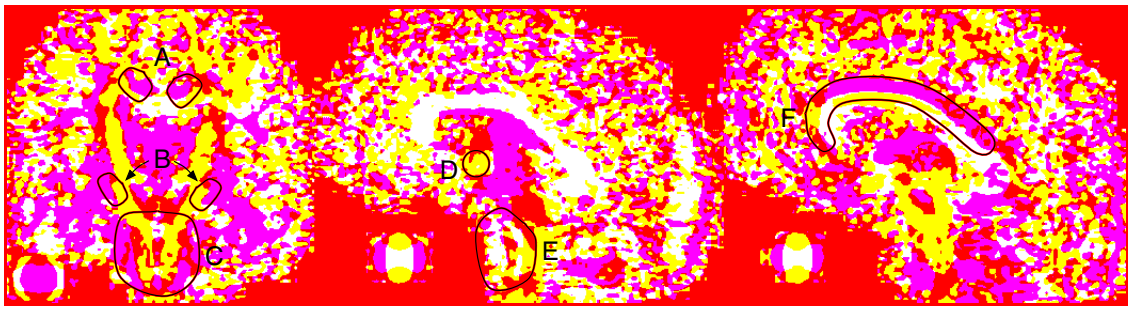
Figure 7.3 decomposes $\|\nabla \mathbf{D}\|$ into the overall magnitudes of shape change and of orientation change. Note that the hard boundary at the surface of the ventricles, indicated by arrows, is present in Figure 7.3(a) but not in Figure 7.3(b), consistent with the idea that the transition between white or gray matter and CSF is indicated by a change in tensor shape, but not in orientation.

Figure 7.4 decomposes the changes in shape along the three orthonormal invariant gradients, as defined in Section 5.3. The features of these images are broadly consistent with the earlier discussion (Section 2.3.3) of the relationship between anatomy and tensor shape. The hard boundary between CSF and the brain is present only in $|\nabla \mu_1|$ (Figure 7.4(a)), and not $|\nabla \mu_2|$ or $|\nabla \alpha_3|$ (Figures 7.4(b) or 7.4(c)), which confirms that the CSF boundary is marked primarily by a change in mean bulk diffusivity. On the other hand, by comparison with the brightly colored regions of Figures 7.1, it is clear that outlines of the white matter structures are visible in Figure 7.4(b) but not Figure 7.4(a), confirming that the contrast between white and gray matter is in the variance μ_2 of the eigenvalues, not the mean μ_1. The structures in the skewness gradient $|\nabla \alpha_3|$ image (Figure 7.4(c)) are less clear, but the boundary between the corpus callosum and the cingulum bundles is being delineated (as indicated). This is consistent with current knowledge of partial voluming in DT-MRI: the voxels in between orthogonally oriented fiber tracts will exhibit planar anisotropy, which has a lower skewness than linear anisotropy (recall Figure 2.7(c)).

However, there is a basic difference between this anatomically-based skewness change and the one created in the synthetic dataset for the sake of Figure 5.4: the continuum of tensor shapes between the corpus callosum and the cingulum bundle ranges from linear, to planar, and back to linear. Skewness reaches a local minimum at the halfway point. Figure 7.5 illustrates this kind of shape variation. Eigenvalue skewness α_3 decreases towards, and increases away from, the planar configuration at the middle of the shape variation, at which skewness is minimal. Because eigenvalue mean and variance



(a) RGB map



(b) RGB map with region labels

Figure 7.1. One coronal (left) slice and two sagittal (right) slices used for inspecting tensor gradient components, shown with the RGB colormap (a), and with anatomical regions of interest labeled (b). A: cingulum bundle (green) superior to corpus callosum (red), B: inferior longitudinal fasciculus (green) lateral to internal capsule (purple), C: cerebellar commissures (green) lateral to cortico-spinal tracts (blue), D: anterior commissure (red), E: trans-pontine tracts (red) amidst cortico-spinal tracts (blue), F: cingulum bundle (green and blue) superior to corpus callosum (red).

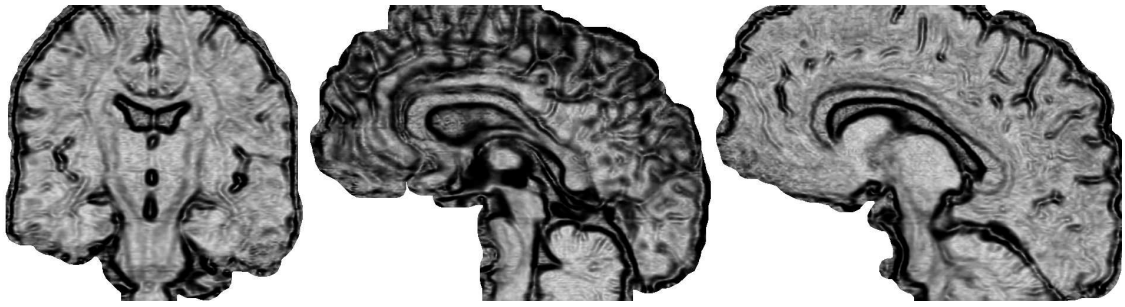
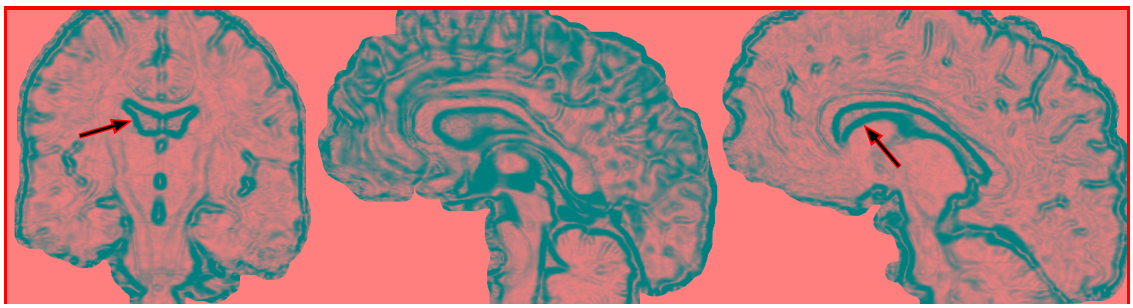
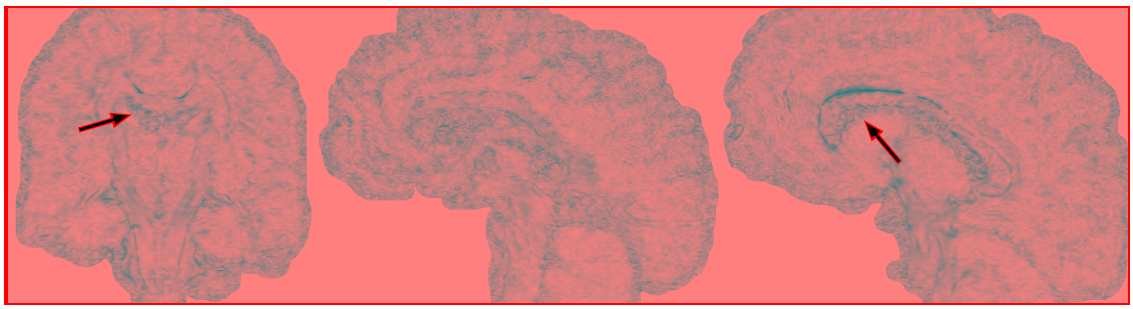


Figure 7.2. Tensor gradient magnitude $\|\nabla \mathbf{D}\|$ on the same dataset slices shown in Figure 7.1.

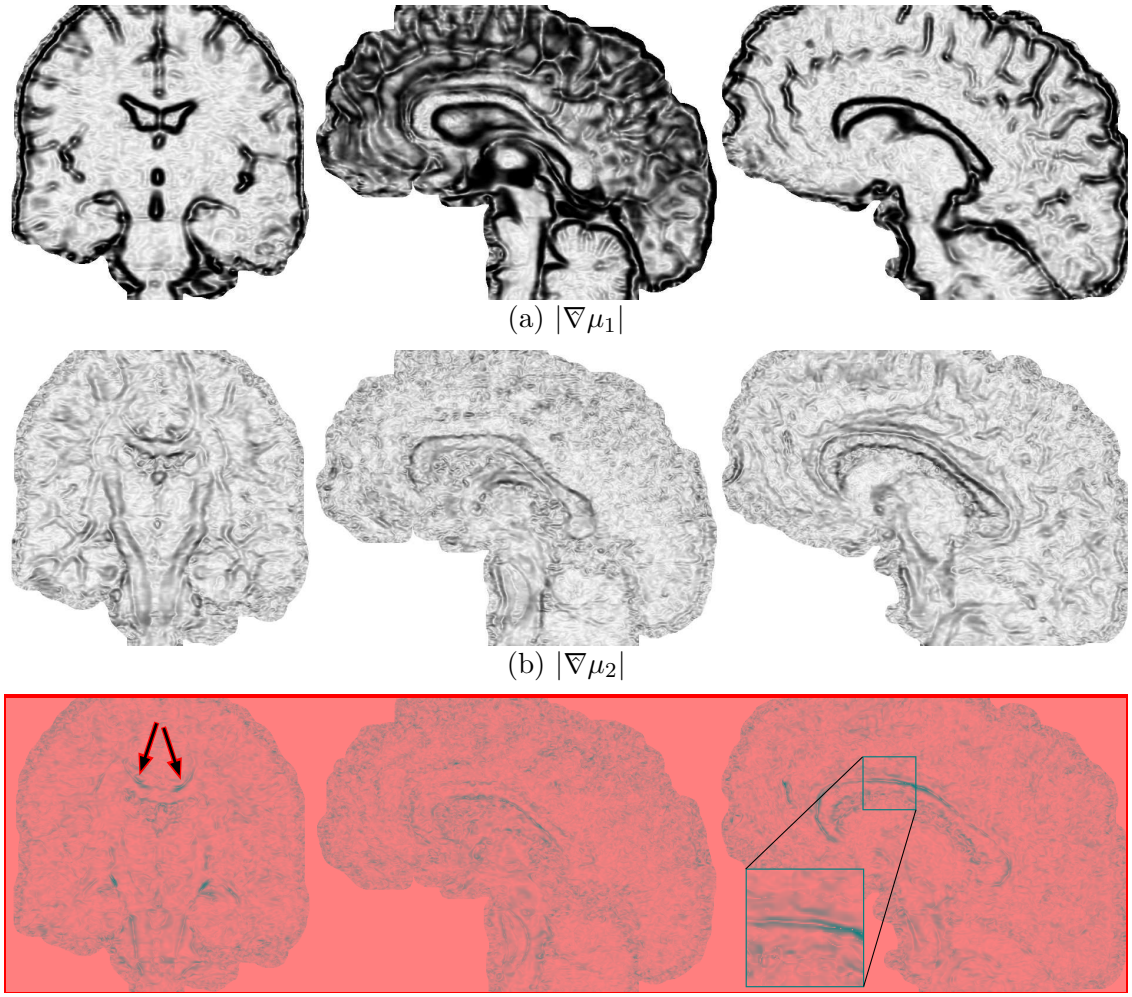


(a) $\sqrt{|\nabla\mu_1|^2 + |\nabla\mu_2|^2 + |\nabla\alpha_3|^2}$



(b) $\sqrt{|\nabla\phi_1|^2 + |\nabla\phi_2|^2 + |\nabla\phi_2|^3}$

Figure 7.3. Images of overall magnitudes of shape change (a) and orientation change (b). Arrows point to boundary of ventricle, which is essentially absent in the orientation change image.



(a) $|\nabla \mu_1|$

(b) $|\nabla \mu_2|$

(c) $|\nabla \alpha_3|$

Figure 7.4. Magnitudes of changes in eigenvalue mean (a), variance (b), and skewness (c). The boundary between cingulum bundle and corpus callosum is indicated by arrows in the coronal (right-most) slice of (c), and by the indicated box (including zoom) in the second sagittal (left-most) slice of (c).

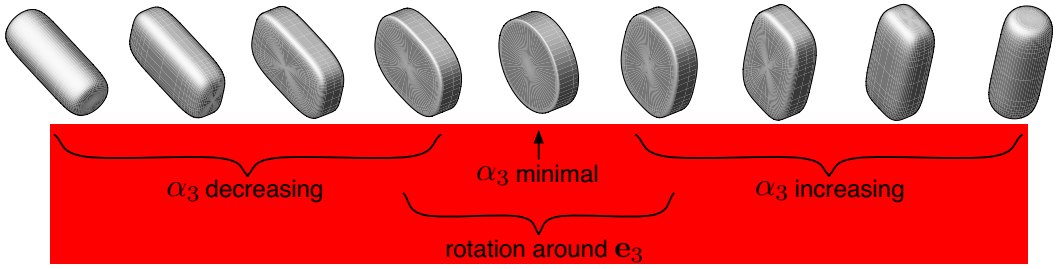
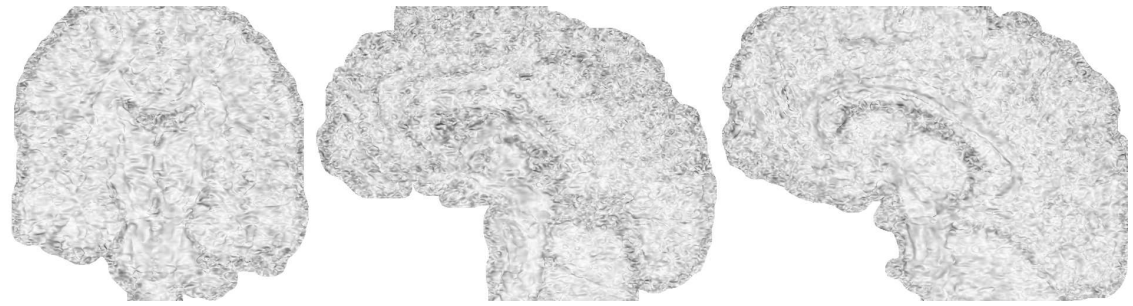


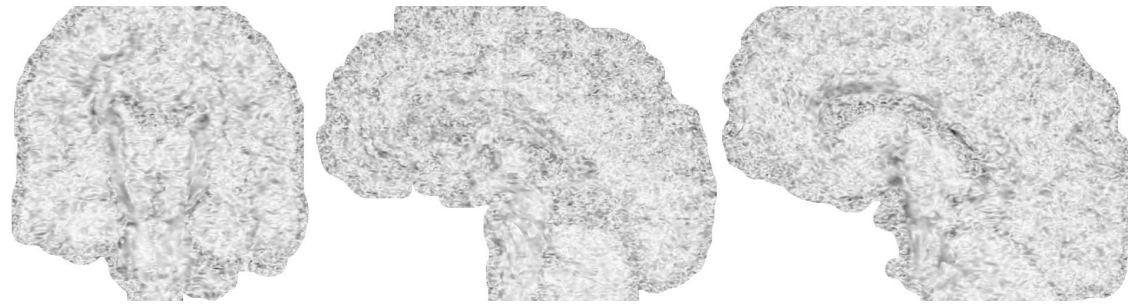
Figure 7.5. Continuous variation between linear, planar, and linear.

is constant throughout this (synthetic) variation, all tensor invariants are fixed at the planar configuration. The difference in orientation which is evident between the initial and final linear shapes is thus concentrated around the planar shape, as a rotation around the minor eigenvector $\mathbf{e}_3$ (the short axis of the planar shape). Consistent with this, close inspection of Figure 7.4(c) reveals isolated pixels of very low magnitude $|\nabla\alpha_3|$: the local change here is in orientation, not shape.

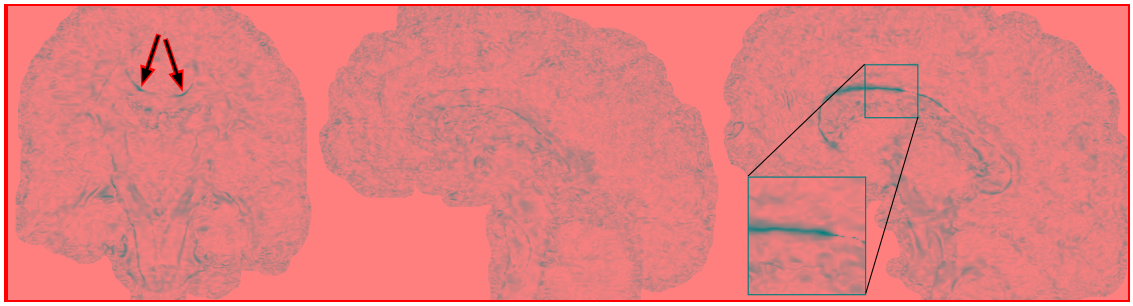
Figure 7.6 helps explain the remaining components of the tensor gradient at this particular anatomical feature. The most striking feature is the clean line in the $|\nabla\phi_3|$ image (Figure 7.6(c)), at a position which separates the corpus callosum and the cingulum bundle, as indicated. This component of the tensor gradient responds to rotations around the minor eigenvector $\mathbf{e}_3$, the eigenvector associated with the lowest diffusivity. A secondary feature of Figure 7.6 is that no white matter structures are clearly visible in



(a) $|\nabla\phi_1|$



(b) $|\nabla\phi_2|$

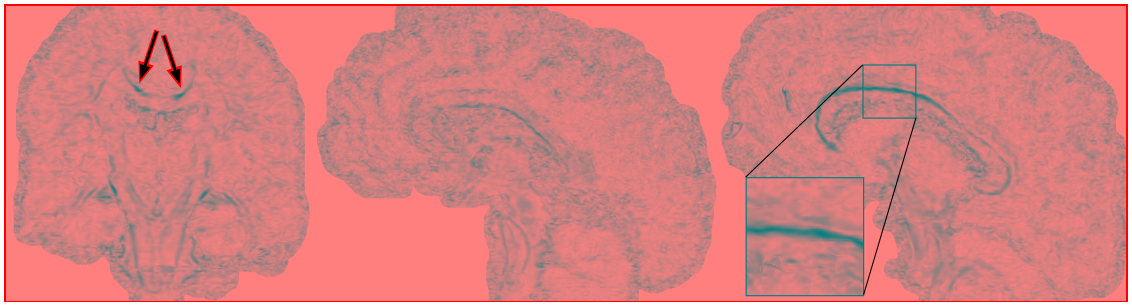


(c) $|\nabla\phi_3|$

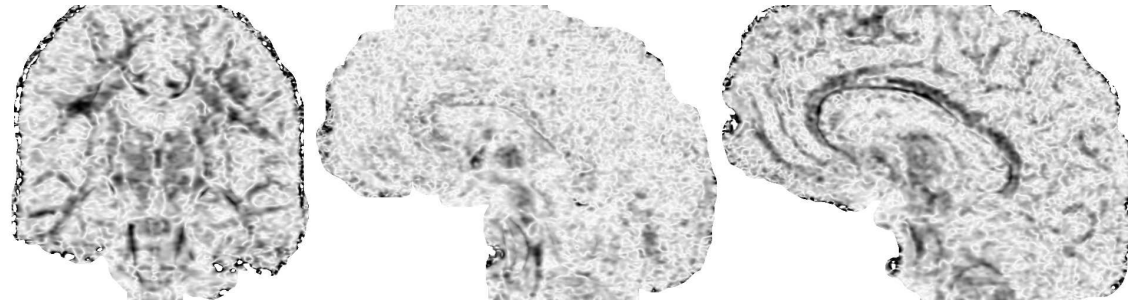
Figure 7.6. Magnitudes of orientation changes measured in DT-MRI slices. The boundary between cingulum bundles and corpus callosum is highlighted in the $|\nabla\phi_3|$ image in (c).

the $|\nabla\phi_1|$ and $|\nabla\phi_2|$ images (Figures 7.6(a) and 7.6(b)). The potential implication is that rotations around the principal eigenvector $\mathbf{e}_1$ and the secondary eigenvector $\mathbf{e}_2$ are not significant indicators of anatomical boundaries, and thus these components of the tensor gradient may be disregarded in the context of feature detection and feature-preserving filtering.

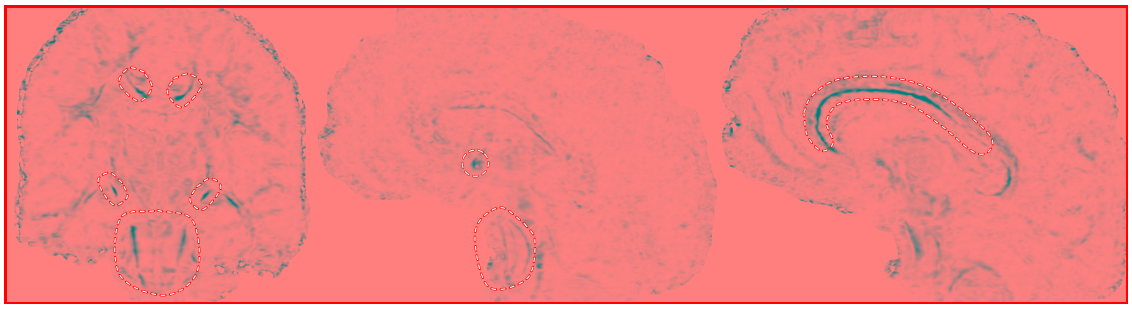
Figure 7.7 shows the two tensor gradient components that successfully delineated the corpus callosum and the cingulum bundle. Recalling that proximity between orthogonally oriented linear anisotropy creates planar anisotropy, Figure 7.7(c) shows that multiplying $\sqrt{|\nabla\alpha_3|^2 + |\nabla\phi_3|^2}$ by c_p (Section 2.3.4) leads to an effective indicator of proximity between orthogonally oriented structures. This is visually confirmed by comparing Figure 7.7(c) with Figure 7.1. The corpus callosum and cingulum bundles are delineated by two dark spots in the coronal slice (left-most image), and by a solid line in the second sagittal image (right-most image). The zoomed-in box regions in Figures 7.4(c),



(a) $\sqrt{|\nabla\alpha_3|^2 + |\nabla\phi_3|^2}$



(b) c_p



(c) $c_p\sqrt{|\nabla\alpha_3|^2 + |\nabla\phi_3|^2}$

Figure 7.7. High values of $\sqrt{|\nabla\alpha_3|^2 + |\nabla\phi_3|^2}$ (a), multiplied by c_p (b), lead to a successful indicator of proximity between orthogonally oriented linear anisotropy (c).

7.6(c), and 7.7(a) in particular demonstrates complementary roles of $|\nabla\alpha_3|$ and $|\nabla\phi_3|$ in detecting the proximity of orthogonal fiber tracts: where one is low the other is high, so that Figure 7.7(a) shows a solid line between the cingulum bundle and corpus callosum as measured by $\sqrt{|\nabla\alpha_3|^2 + |\nabla\phi_3|^2}$.

Figure 7.7(c) outlines the numerous locations of proximal orthogonal tracts shown in Figure 7.1. The success of the detection method is indicated by the presence of dark strips at the centers of the regions indicated by red dashed outlines. The anterior commissure, for example, is visible as a single dark spot in the first sagittal image (middle image). Less striking but also visible are the demarcations of orthogonally oriented pathways in the brainstem. To our knowledge, the sensitivity and specificity of $c_p\sqrt{|\nabla\alpha_3|^2 + |\nabla\phi_3|^2}$ (Figure 7.7(c)) for detecting contact between orthogonal fiber structures has not been previously demonstrated. Feature-preserving filtering of the tensor field can using this information as a means of avoiding inappropriate blurring in these regions.

7.2 Application to a Second Dataset

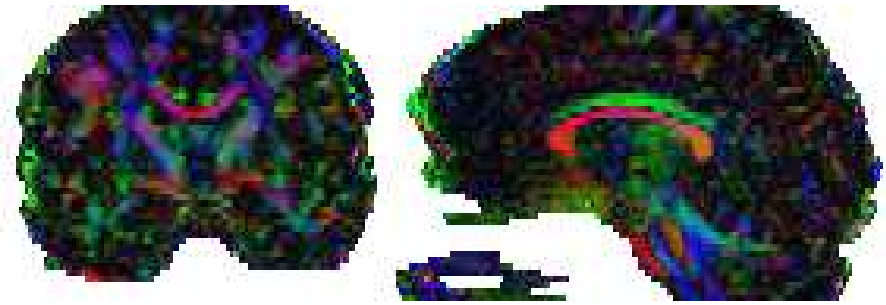
The 1 mm $\times$ 1 mm $\times$ 1 mm $\times$ resolution of the dataset used in the previous section (and elsewhere in the dissertation) is higher than datasets typical in DT-MRI studies. This section applies the method developed in the previous section to such a typical dataset, published online in an effort to standardize comparison of tractography algorithms[1]. The specifics of the pulse sequence are unfortunately not disclosed. It is likely acquired by echo-planar imaging, based on the amount of "ghost" anisotropy (anatomically implausible high anisotropy on the surface of the cortex) caused by mis-registration of the individual diffusion weighted images used for tensor computation [156]. The spatial resolution of the data is 1.25 mm $\times$ 1.25 mm along X and Y, and 2.5 mm along Z. To have the correct aspect ratio, the RGB images below were up-sampled by a factor of two along Z with nearest-neighbor interpolation. All the gradient magnitude images are up-sampled by three along X and Y, and by six along Z, using a cubic spline and its derivative for all measurements. The particular spline is from the (B, C) family of cubic splines, with $(B, C) = (0.5, 0.25)$, which is a compromise between the blurring characteristics of the B-spline ($(B, C) = (1, 0)$) and the interpolation of Catmull-Rom ($(B, C) = (0, 0.5)$) [128].

Figure 7.8 shows a coronal and a sagittal slice of this dataset. Compared to Figure 7.1(a), Figure 7.8(a) makes clear that the number of voxels per anatomical feature is lower in this dataset, especially along the Z direction. The major anatomical structures in Figure 7.8(a) are the same as in Figure 7.1(a), and as before, the main task here will be to delineate the corpus callosum from the cingulum bundles.

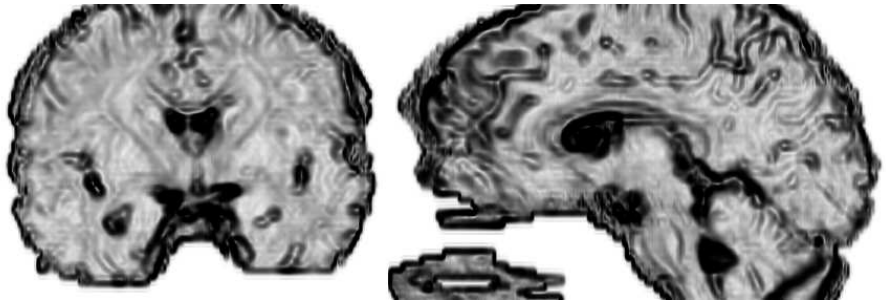
Figure 7.9 shows the decomposition of $\|\nabla\mathbf{D}\|$ into three degrees of freedom in shape change (left side) and orientation change (right side). The patterns described in the previous section are visible here as well: the brain-CSF boundary is most visible in the $|\nabla\mu_1|$ image (Figure 7.9(a)), the interface between white and gray matter is most visible in the $|\nabla\mu_2|$ image (Figure 7.9(c)), while little structure is obvious in the $|\nabla\phi_1|$ and $|\nabla\phi_2|$ images (Figures 7.9(b) and 7.9(d)). However, $|\nabla\alpha_3|$ and $|\nabla\phi_3|$ seem to delineate the cingulum bundles and corpus callosum.

Figure 7.10 separately demonstrates the $\sqrt{|\nabla\alpha_3|^2 + |\nabla\phi_3|^2}$ and c_p factors in the $c_p\sqrt{|\nabla\alpha_3|^2 + |\nabla\phi_3|^2}$ indicator of proximal orthogonal fiber tracts. The cingulum bundle

[1] `http://www.ujf-grenoble.fr/ismrm/Diffusion/DTI/`



(a) RGB map



(b) $\|\nabla \mathbf{D}\|$

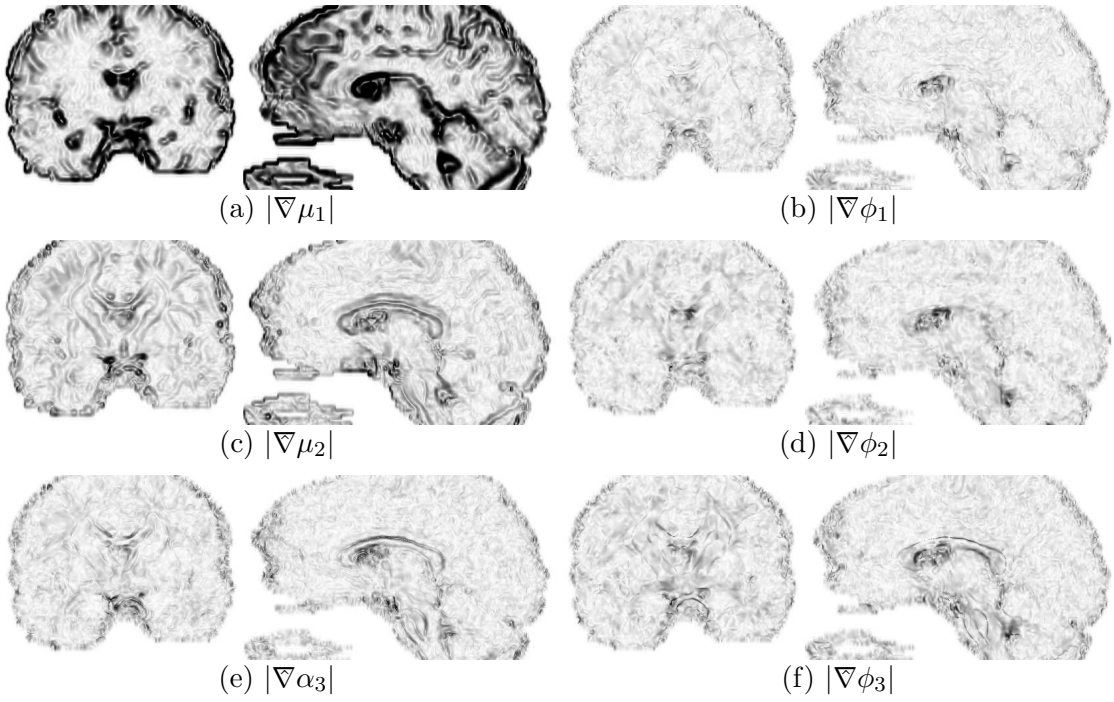
Figure 7.8. Images of RGB colormap (a) and overall gradient magnitude (b) in one coronal (left) and one sagittal (right) slice.

and corpus callosum are well-delineated, and the tracts in the brainstem are actually better indicated in this dataset than in the previous. One notable aspect of Figure 7.10(c) is that the narrowness and smoothness of the tract delineations is in contrast to the coarse data resolution evidenced in Figure 7.8(a). This can be ascribed to the careful use of continuous convolution for all value and derivative measurements, as originally motivated in Section 2.1.8.

7.3 Discussion

This chapter has described preliminary results of feature-detection relevant for feature-preserving filtering of diffusion tensor fields. Based on inspecting images of tensor gradient components, we verified that anatomical boundaries can be characterized by certain gradient components. At the boundary between the brain tissue and the cerebrospinal fluid (CSF) $|\nabla \mu_1|$ is high. At the boundary between gray and white matter, $|\nabla \mu_2|$ is high, but this boundary is not manifest as sharply as the CSF boundary. Both $|\nabla \alpha_3|$ and $|\nabla \phi_3|$ were used to delineate regions where orthogonal fiber tracts touch. Finally, $c_p \sqrt{|\nabla \alpha_3|^2 + |\nabla \phi_3|^2}$ was demonstrated as a novel indicator of orthogonally oriented anisotropy.

More work is required to continue and refine this approach. For example, Figure 7.4(a) convincingly shows that the boundary with CSF may be detected by high values of $|\nabla \mu_1|$, but using $|\nabla \mu_2|$ as an indicator for the gray-matter/white-matter boundary is less clear cut. Looking at Figure 7.1 confirms that not all white matter tracts have equal anisotropy contrast: the boundary of the internal capsule and corticospinal tracts in the brainstem



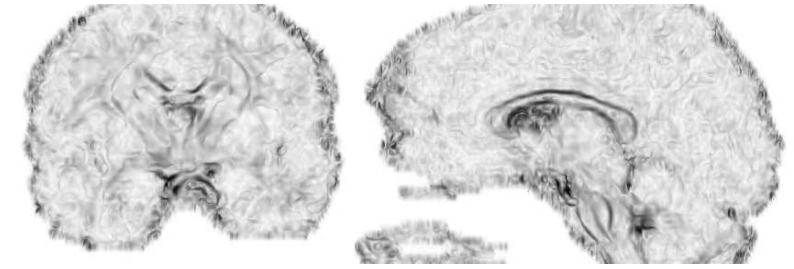
(a) $|\nabla\mu_1|$ (b) $|\nabla\phi_1|$

(c) $|\nabla\mu_2|$ (d) $|\nabla\phi_2|$

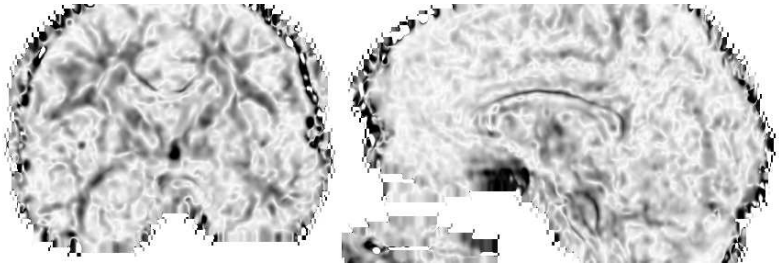
(e) $|\nabla\alpha_3|$ (f) $|\nabla\phi_3|$

Figure 7.9. Magnitudes of three shape and three orientation components of $\nabla\mathbf{D}$.

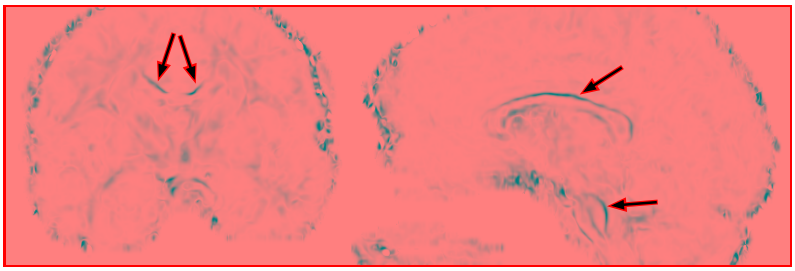
(the blue "V"-shaped configuration in the coronal image) is much sharper than than of the corona radiata, so the values of $|\nabla\mu_2|$ are lower around the corona radiata. It may be that for the purposes of feature-preserving filtering, the most reliable way to isolate the gray-matter/white-matter boundary is not by the derivative of the tensor field, but by choosing a threshold value of fractional anisotropy. Such a method, combined with $|\nabla\mu_1|$ to indicate the CSF boundary, and $c_p\sqrt{|\nabla\alpha_3|^2 + |\nabla\phi_3|^2}$ to delineate orthogonally oriented tracts, should permit feature-preserving filtering of DT-MRI datasets.



(a) $\sqrt{|\nabla\alpha_3|^2 + |\nabla\phi_3|^2}$



(b) c_p



(c) $c_p\sqrt{|\nabla\alpha_3|^2 + |\nabla\phi_3|^2}$

Figure 7.10. Second demonstration of the two factors in detecting proximal orthogonal tracts: $\sqrt{|\nabla\alpha_3|^2 + |\nabla\phi_3|^2}$ (a) detects the linear-planar-linear transition, while the c_p measure (b) highlights regions of planar anisotropy, resulting in a successful delineation (c) of the cingulum bundle and corpus callosum, as well as orthogonal tracts in the brainstem (indicated by arrows).

CHAPTER 8

CONCLUSIONS

This dissertation has presented a diverse combination of novel methods for diffusion tensor visualization and analysis. As mention in Chapter 1, the different visualization methods can be compared in terms of what aspect of the tensor field is shown, how that information is transformed to a visible form, and where are the locations at which it is visualized.

Superquadric glyphs and reactions diffusion textures display all the degrees of freedom in tensor values, at a discretely sampled set of points. By encoding shape information into a variable base geometry, superquadric glyphs avoid the visual ambiguities of standard ellipsoidal glyphs. The packing and spacing of the spots that arise in the reaction-diffusion textures help indicate the form of underlying structures more explicitly than by glyphs on a strictly regular sampling grid. Tensorlines visualize a subset of the tensor information, along a collection of continuous paths initiated by points chosen by the user. Each tensorline conveys the directional information imparted to it by *deflection*, in addition to the the principal eigenvector (the basis of standard tractography). Direct volume rendering, on the other hand, seeks to convey one or two aspects of tensor shape, everywhere, by rendering and shading the continuous form associated with the spatial variation of the shape information through out the field. The mathematical center of the dissertation is Chapter 5, describing a framework for calculating the components of shape change and orientation change in the *gradient* of a tensor field, the first-order differential structure of the tensor field at each point. The tensor gradient mathematics was first applied to create analytical shading of anisotropy surfaces in volume rendering, and to create a novel indicator for proximity of orthogonally oriented fiber tracts.

All the different methods can be characterized in terms of local versus global field information, and in terms of which tensor attributes are embodied. This even includes the mathematics of the tensor gradient analysis (Chapter 5): by definition it is completely local (it describes the field at exactly a point), and Sections 5.3 and 5.8 described how to extract its shape and orientation attributes.

8.1 Future Work

As stated in Chapter 1, this dissertation contributes a discrete set of methods, rather than a unified solution to a single over-arching problem of visualization or analysis. However, the methods described are complementary in their approach and focus. The possible bridges between the methods are natural areas for future work. Rather than visualizing tensorlines (Chapter 4) has a continuous tubular geometry, a string of superquadric glyphs (Section 3.1) could indicate the path of the tensorline, while also conveying all the tensor attributes at regularly spaced points along the path. The tensorlines themselves could be seeded from the centers of the spots created by the reaction-diffusion textures

(Section 3.2). The anisotropy thresholds required for tensorline or glyph renderings could be chosen based on direct volume renderings of the continuous anisotropy isosurface (Section 6.4), and volume renderings could be integrated into tensorline or glyph visualizations to provide further indication of large-scale structures. In particular, the features identified by the methods of Chapter 7 could be visualized and contextualized by a combination of glyphs, tensorlines, and volume rendering.

Although Section 2.2.4 stated that the methods of this dissertation are targeted at diffusion tensor rather than high angular resolution diffusion (HARD) imaging, certain methods may generalize naturally to HARD data. For example, the description of tensor shape based on orthogonal statistical measures of the eigenvalues (Section 2.3) may naturally extent to higher-order moments or cumulants, which may provide a more straight-forward approach to shape description than the spherical harmonics that are currently favored [4, 65]. Although the generalization of glyphs and textures to HARD data is a challenge, the deflection principle behind tensorlines may generalize well, whereby the input direction is simply steered towards the directionally closest peak in diffusivity. One of the most promising visualization methods may actually be volume rendering. Because it is so flexible with respect to which data attributes contribute to the rendering, generalized anisotropy and shape measurements could serve as transfer function domain variables, creating renderings of continuous three-dimensional forms which to date have not been computed from HARD data.

There are numerous possible directions of future work based on the mathematical formalism of Chapter 5. As stated there, the original intent of this approach was the development of feature-preserving nonlinear filtering, akin to the Perona-Malik method for grayscale images. It is this method which motivated the detection of proximity of orthogonal fiber tracts, since these are especially important to keep seperate during filtering. However, the same mathematical vocabulary for decomposing the tensor gradient may allow the detection of other fiber tract features such as the branches. This location is also characterized by a linear-planar-linear transition in the field, but of a different configuration than that illustrated in Figure 7.5. In other case, it would be useful to verify the feature detection based on gradient analysis with HARD imaging, since this provides a more direct validation of the underlying directional composition. Another exciting direction of future work involves using topological methods to decompose the tensor field into spatially discrete regions, bounded by critical points and separating surfaces, within which the structure of the field is topologically constant. Limited versions of these methods have been described for tensor fields [83, 53, 82] and vector fields [77, 68]. The generalizations to tensor fields may detect anatomical landmarks which serve as fiducial points for deformable registration methods.

An important area that this dissertation has left for future work is that of perception and perceptual psychology. Although the efficacy of superquadric glyphs (over ellipsoids) is evident from Figure 3.8, and the improvements from reaction-diffusion textures (over ellipses on a regular grid) is visible in Figure 3.15, rigorous validation of the methods will be found in future user studies. User studies could *quantify* the improvement in the ability to discern the relevant tensor and field properties, as well as the viewing and presentation conditions under which the methods perform best.

Different studies are possible for the different visualization methods. One study could quantify the advantage of superquadrics to convey the orientation of the principal eigenvector (a fundamental task of diffusion tensor visualization), and another could quantify the ability to differentiate the tensor shape variations that occur within the

field. The effectiveness of glyphs also depends on their screen resolution, so another study could determine if the greater surface complexity of superquadric glyphs decreases their legibility as a function of lower screen resolutions. With the reaction diffusion textures, it would be interesting to quantify the extent to which the nonregular placement of the spots allows underlying field properties to be perceived. To validate the volume rendering methods, it would be interesting the compare the volume rendered isosurfaces of anisotropy with a polygonal representation of the same, to determine the improvement in ability to perceive the overall form of the anisotropic regions. To a large extent, the tensorline method has been validated by Lazar et al. [113, 114], although future studies could assess how to best combine the different visualization methods to facilitate particular diagnostic or analysis tasks.

Another related area left largely unaddressed by this dissertation is the issue of interactivity and presentation. The scientific utility of a visualization method is lessened if it can not interactively respond to queries and exploration by a researcher, and if the images are not accessible in a natural context for scientific exploration. These "meta"-visualization issues have been previously addressed for diffusion tensor visualization, although mainly for exploration by tractography [54, 202, 1]. An interesting example of the role of interactivity in visualization is the interactive and artistic design of glyphs in a virtual environment, to improve the expressiveness of multivariate flow visualizations [96, 97].

Experience has shown that interactivity fundamentally simplifies the task of transfer function specification in direct volume rendering of scalar fields [107, 108], so it is frustrating that volume rendering directly from the tensor field currently exceeds the computational ability of graphics hardware. As mentioned in Section 6.5, however, this can be expected to change, in which case the vocabulary of interactive tensor visualizations methods would expand significantly. Interactive methods would be especially helpful in the exploration and design of analysis methods (Chapter 5), since the variables defined by projections of the tensor gradient could be used in transfer functions, allowing volumetric exploration of the structures delineated by gradients of the tensor field.

Beyond the *perceptual* validation of the methods described above, a major avenue for longer-term future work is the *clinical* validation of the methods presented in this dissertation. The initial component of clinical validation would be a careful determination of the noise sensitivity of the methods, based on previous work describing the noise characteristics that can be expected in clinical scans [17, 18, 19, 141]. This is a problem for some methods more than others. The virtue of the superquadric glyphs, for example, is that they can display all tensor attributes at the originally scanned point samples, noise included. Volume rendering is similar in that noise will be evidenced in visibly rougher anisotropy surfaces, especially when using reconstruction kernels which interpolate rather than smooth. Because of its reliance on differentiation, the analysis method of Chapters 5 and 7 is especially sensitive to noise, although the intended development of feature-preserving filtering may address this.

A more significant component of clinical validation will be demonstrating that the methods provably assist in one or more of the many clinical applications of DT-MRI (outlined in Section 1.1). It is common to find a gap between the intent of an advanced visualization method, and its actual acceptance by the scientists or doctors for whom the method is intended. The reason for this gap is likely due to the basic difference between the work involved in developing the algorithmic and mathematical underpinnings for a technique, versus performing statistical tests in a specific scientific or clinical context to evaluate the application of the technique. Visualization research in computer science has

generally focused more on the former than the latter, and this dissertation is no exception. In this light, our reliance on a single dataset is a short-coming, but not a major weakness.

Each of the methods presented in this dissertation could be tested by collaborations with medical professionals to define a particular diagnostic task, testing the new methods against established ones through user-studies, and calculating the statistics to quantify the difference between methods. This represents a major undertaking because the task will likely be different for the different methods: for glyphs the task may be detecting certain artifacts in the tensor dataset, for tensorlines it may be characterizing a tumor's influence on surrounding white matter, and for volume rendering it may be detecting larger-scale anatomical differences between a disease and a normal population. It is our hope that at least some of the methods presented here can be carried through to this final stage of validation and scientific acceptance.

APPENDIX A

INDEX NOTATION

Index notation, or *Einstein summation notation*, enables complicated expressions, often involving matrix multiplications, to be represented concisely, so that essential transformations and simplifications are easier to apply. This appendix provides a self-contained introduction to index notation; Barr [11] provides additional information.

The basic rule is that the repetition of an index within a term implies summation of the term over the index's range of values. As an example of index notation, here are two definitions of the matrix multiplication $C = AB$, in both explicit summation and index notation:

$$\begin{aligned} C_{ij} = (AB)_{ij} &= \sum_k A_{ik} B_{kj} \\ (AB)_{ij} &= A_{ik} B_{kj} \ . \end{aligned}$$

Two types of indices appear in index notation. Indices repeated within a term (such k above) are *dummy* indices, since they could be replaced by any other (un-used) letter without changing the value of sum. Non-repeated indices (such as i and j above) are *free* indices, and these have some significance outside the scope of an individual term. The $(AB)_{ij}$ on the left-hand side indicates which row and which column of the product matrix is being defined, so i and j have significance on the right-hand side, whereas k could just as well be replaced by l or m. Typically, dummy indices appear exactly twice in each term. In the context of matrix multiplication, for example, the dummy index is always the second index of the first matrix (running over the columns), and the first index of the second matrix (running over the rows): $A_{ik} B_{kj}$.

As seen in Section 2.1.2, a common use of index notation is to express a vector in some basis. Assume an orthonormal basis $\mathcal{B} = \{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}$, so that $\mathbf{b}_i \cdot \mathbf{b}_j = \delta_{ij}$. The coordinates of $\mathbf{v}$ in $\mathcal{B}$ are (v_1, v_2, v_3), $v_i = \mathbf{v} \cdot \mathbf{b}_i$, so we can write:

$$\mathbf{v} = v_i \mathbf{b}_i \ .$$

As a first example of using index notation, we can prove that the dot-product of two vectors is invariant with respect to basis. If the dot product has been defined geometrically, in terms of the cosine of the angle between the vectors, then there is nothing to prove, because the dot product is inherently coordinate-free. If the dot product is defined, however, as the usual pair-wise product of coordinates:

$$\mathbf{v} \cdot \mathbf{u} = \sum_i v_i u_i \ .$$

then showing the invariance is less trivial. Let P be the orthogonal matrix which represents a change of basis from unmarked coordinates v_i to tilde coordinates $\widetilde{v}_i$. Without index notation:

$$\begin{aligned}
\mathbf{v}\cdot\mathbf{u} &= \sum_i \widetilde{v}_i\widetilde{u}_i \\
&= \sum_i (\sum_j P_{ij}v_j)(\sum_k P_{ik}u_k) \\
&= \sum_i \sum_j \sum_k P_{ij}P_{ik}v_j u_k \\
&= \sum_j \sum_k \sum_i P^{\mathsf{t}}_{ji}P_{ik}v_j u_k \\
&= \sum_j \sum_k (P^{\mathsf{t}}P)_{jk}v_j u_k \\
&= \sum_j \sum_k \delta_{jk}v_j u_k \\
&= \sum_j v_j u_j \ .
\end{aligned}$$

With index notation:

$$\begin{aligned}
\mathbf{v}\cdot\mathbf{u} &= \widetilde{v}_i\widetilde{u}_i \\
&= P_{ij}v_jP_{ik}u_k \\
&= P_{ij}P_{ik}v_ju_k \\
&= P^{\mathsf{t}}_{ji}P_{ik}v_ju_k \\
&= (P^{\mathsf{t}}P)_{jk}v_ju_k \\
&= \delta_{jk}v_ju_k \\
&= v_ju_j \ .
\end{aligned}$$

As another example of the utility of index notation, two proofs of $\sum_{i=1}^{3}\mathbf{b}_i\otimes\mathbf{b}_i = \mathbf{I}$ are given. Both proofs work by showing that $(\sum_{i=1}^{3}\mathbf{b}_i\otimes\mathbf{b}_i)\mathbf{v} = \mathbf{v}$ for any vector $\mathbf{v}$. First, the proof without index notation:

$$\begin{aligned}
\sum_i(\mathbf{b}_i\otimes\mathbf{b}_i)\mathbf{v} &= \sum_i \mathbf{b}_i(\mathbf{b}_i\cdot\mathbf{v}) \\
&= \sum_i \mathbf{b}_i(\mathbf{b}_i\cdot(\sum_j v_j\mathbf{b}_j)) \\
&= \sum_i \mathbf{b}_i(\sum_j v_j\mathbf{b}_i\cdot\mathbf{b}_j) \\
&= \sum_i \mathbf{b}_i(\sum_j v_j\delta_{ij}) \\
&= \sum_i \mathbf{b}_i v_i \\
&= \mathbf{v} \ .
\end{aligned}$$

Then with index notation:

$$\begin{aligned}(\mathbf{b}_i \otimes \mathbf{b}_i)\mathbf{v} &= \mathbf{b}_i(\mathbf{b}_i \cdot \mathbf{v}) \\ &= \mathbf{b}_i(\mathbf{b}_i \cdot v_j\mathbf{b}_j) \\ &= \mathbf{b}_i(v_j\delta_{ij}) \\ &= \mathbf{b}_i v_i \\ &= \mathbf{v}\ .\end{aligned}$$

Note that in both of the proofs above, δ_{ij} plays the role of filtering out a single term from a summation over the j index, effectively replacing occurances of j with i elsewhere in the expression.

Showing that the trace of a matrix is invariant under change-of-basis is another example of the utility of index notation. First, note that the trace of matrix D can be notated simply as D_{ii}. The repeated index represents summation of the matrix entries for which the row and column are equal (the diagonal). The trace could also be notated D_{kk}: the letter used in the dummy index doesn't matter. Below, the matrix P represents the change of basis, which could actually be any invertible matrix (not just an orthogonal one representing a rotation):

$$\begin{aligned}\mathrm{tr}(PDP^{-1}) &= (PDP^{-1})_{ii} \\ &= P_{ij}D_{jk}{P^{-1}}_{ki} \\ &= {P^{-1}}_{ki}P_{ij}D_{jk} \\ &= (P^{-1}P)_{kj}D_{jk} \\ &= \delta_{kj}D_{jk} \\ &= D_{kk} \\ &= \mathrm{tr}(D)\ .\end{aligned}$$

The ability to rearrange factors (such as moving P^{-1} to the beginning of the term) without worrying about distinctions between row and column vectors is another advantage of index notation. The individual factors of an index notation term are in fact scalars, and unlike matrix multiplication, scalar multiplication is commutative, which enables the reordering used above. The proofs above were careful to only show one small transformation or rearrangement at a time, but familiarity with index notation eventually permits one to comfortably write these proofs with much fewer steps.

APPENDIX B

ORTHOGONALITY OF μ_1, μ_2, α_3

This section provides a self-contained proof of the fact that that mean μ_1, variance μ_2, and skewness α_3, viewed as functions from $\mathbb{R}^n$ to $\mathbb{R}$, are mutually orthogonal in the sense that:

$$\nabla\mu_2 \cdot \nabla\mu_1 = 0 \tag{B.1}$$
$$\nabla\alpha_3 \cdot \nabla\mu_1 = 0 \tag{B.2}$$
$$\nabla\alpha_3 \cdot \nabla\mu_2 = 0\,. \tag{B.3}$$

In the following, $\vec{x} = (x_1, x_2, ..., x_n) \in \mathbb{R}^n$, $\vec{1} = (1, 1, ..., 1)$, and $\langle x\rangle = \frac{1}{n}\sum x_i$.

Starting with expressions for μ_1, μ_2, and μ_3,

$$\begin{aligned}
\mu_1 &= \langle x\rangle \\
\mu_2 &= \langle (x - \langle x\rangle)^2\rangle \\
&= \langle x^2 - 2x\langle x\rangle + \langle x\rangle^2\rangle \\
&= \langle x^2\rangle - 2\langle x\rangle\langle x\rangle + \langle x\rangle^2 \\
&= \langle x^2\rangle - \langle x\rangle^2 \\
\mu_3 &= \langle (x - \langle x\rangle)^3\rangle \\
&= \langle x^3 - 3x^2\langle x\rangle + 3x\langle x\rangle^2 - \langle x\rangle^3\rangle \\
&= \langle x^3\rangle - 3\langle x^2\rangle\langle x\rangle + 3\langle x\rangle\langle x\rangle^2 - \langle x\rangle^3 \\
&= \langle x^3\rangle - 3\langle x^2\rangle\langle x\rangle + 2\langle x\rangle^3\,,
\end{aligned}$$

we form expressions for their gradients:

$$\begin{aligned}
\nabla\mu_1 &= \frac{\vec{1}}{n} \\
\nabla\mu_2 &= \frac{2}{n}(\vec{x} - \langle x\rangle\vec{1}) \\
\nabla\mu_3 &= \frac{3}{n}\vec{x^2} - 3\left(\frac{2}{n}\vec{x}\langle x\rangle + \frac{1}{n}\langle x^2\rangle\vec{1}\right) + \frac{6}{n}\langle x\rangle^2\vec{1} \\
&= \frac{3}{n}\left(\vec{x^2} - 2\langle x\rangle\vec{x} + (2\langle x\rangle^2 - \langle x^2\rangle)\vec{1}\right)\,.
\end{aligned}$$

From this we can show $\nabla\mu_1 \cdot \nabla\mu_2 = 0$ (Equation B.1):

$$\begin{aligned}
\nabla\mu_1 \cdot \nabla\mu_2 &= \frac{2}{n^2}(\vec{x}\cdot\vec{1} - \langle x\rangle\vec{1}\cdot\vec{1}) \\
&= \frac{2}{n}(\langle x\rangle - \langle x\rangle) \\
&= 0\,.
\end{aligned}$$

To show $\nabla\mu_1 \cdot \nabla\alpha_3 = 0$ (Equation B.2) and $\nabla\mu_2 \cdot \nabla\alpha_3 = 0$ (Equation B.3), we start by expressing $\nabla\alpha_3$ using the quotient rule of derivatives. Recall that skewness $\alpha_3 = \mu_3/\sigma^3$ and $\sigma^3 = \sqrt{\mu_2}^3$.

$$\begin{aligned}
\nabla\alpha_3 &= \frac{\sigma^3\nabla\mu_3 - \mu_3\nabla\sigma^3}{\sigma^6} \\
&= \frac{\nabla\mu_3}{\sigma^3} - \frac{3}{2}\frac{\mu_3\sqrt{\mu_2}\nabla\mu_2}{\sigma^6} \\
&= \frac{\nabla\mu_3}{\sigma^3} - \frac{3}{2}\frac{\mu_3\nabla\mu_2}{\sqrt{\mu_2}^5} \ .
\end{aligned}$$

With this and Equation B.1, to show $\nabla\alpha_3 \cdot \nabla\mu_1 = 0$ (Equation B.2), it suffices to show:

$$\begin{aligned}
\nabla\mu_3 \cdot \nabla\mu_1 &= \frac{3}{n^2}(\vec{x^2} \cdot \vec{1} - 2\langle x\rangle\vec{x} \cdot \vec{1} + (2\langle x\rangle^2 - \langle x^2\rangle)\vec{1} \cdot \vec{1}) \\
&= \frac{3}{n}(\langle x^2\rangle - 2\langle x\rangle\langle x\rangle + 2\langle x\rangle^2 - \langle x^2\rangle) \\
&= 0 \ .
\end{aligned}$$

Because $\nabla\mu_1 = \vec{1}/n$, this implies $\nabla\mu_3 \cdot \vec{1} = 0$, and from Equation B.1 we know $\nabla\mu_2 \cdot \vec{1} = 0$. It is also helpful to know:

$$\begin{aligned}
\nabla\mu_2 \cdot \vec{x} &= \frac{2}{n}(\vec{x} - \langle x\rangle\vec{1}) \cdot \vec{x} \\
&= \frac{2}{n}(\vec{x} \cdot \vec{x} - n\langle x\rangle\langle x\rangle) \\
&= 2(\langle x^2\rangle - \langle x\rangle^2) \\
&= 2\mu_2 \ ,
\end{aligned}$$

and:

$$\begin{aligned}
\nabla\mu_3 \cdot \vec{x} &= \frac{3}{n}(\vec{x^2} - 2\langle x\rangle\vec{x} + (2\langle x\rangle^2 - \langle x^2\rangle)\vec{1}) \cdot \vec{x} \\
&= 3(\langle x^3\rangle - 2\langle x\rangle\langle x^2\rangle + (2\langle x\rangle^2 - \langle x^2\rangle)\langle x\rangle) \\
&= 3(\langle x^3\rangle - 3\langle x\rangle\langle x^2\rangle + 2\langle x\rangle^3) \\
&= 3\mu_3 \ .
\end{aligned}$$

From these pieces, Equation B.3 follows easily:

$$\begin{aligned}
\nabla\alpha_3 \cdot \nabla\mu_2 &= \frac{2}{n}\left(\frac{\nabla\mu_3}{\sigma^3} - \frac{3}{2}\frac{\mu_3\nabla\mu_2}{\sqrt{\mu_2}^5}\right) \cdot (\vec{x} - \langle x\rangle\vec{1}) \\
&= \frac{2}{n}\left(\frac{\nabla\mu_3}{\sigma^3} - \frac{3}{2}\frac{\mu_3\nabla\mu_2}{\sqrt{\mu_2}^5}\right) \cdot \vec{x} \\
&= \frac{2}{n}\left(\frac{3\mu_3}{\sigma^3} - \frac{3\mu_3}{\sqrt{\mu_2}^3}\right) \\
&= 0 \ .
\end{aligned}$$

Note that $\nabla\mu_3$ and $\nabla\mu_2$ are not perpendicular:

$$\begin{aligned}\nabla\mu_3 \cdot \nabla\mu_2 &= \frac{2}{n}\nabla\mu_3 \cdot (\vec{x} - \langle x\rangle\vec{1}) \\ &= \frac{6}{n}\mu_3 \ .\end{aligned}$$

APPENDIX C

RAY-TRACING SUPERQUADRICS

This appendix describes a previously unpublished detail pertinent to ray-tracing superquadric tensor glyphs, based on the implicit surface representation.

Glyph-based visualization is most easily implemented in ray-tracing by the method of "instancing" [170, 69]. Instead of computing the intersection of the ray and the glyph, we compute the intersection of a transformed ray and the canonical base glyph geometry G_0. The superquadric glyph ray-tracing task is thereby reduced to computing ray intersections with the surface represented by the q_z function Equation 3.3 (axial symmetry around z axis) and Equation 3.5 (axial symmetry around x axis). The discussion below considers only the former.

The ray is parameterized by:

$$\mathbf{p}(t) = \mathbf{p}_0 + t\mathbf{d}; t \in [t_{\min}, t_{\max}] \ .$$

As sampled along the ray, the superquadric function can be parameterized by:

$$q(t) = q_z(\mathbf{p}(t)) \ .$$

The ray-superquadric intersection is defined as the lowest $t_0 \in [t_{\min}, t_{\max}]$ such that $q(t_0) = 0$. Because the superquadric is analytically defined, we should be able to use the Newton-Raphson iterative root-finding method to rapidly compute the intersection [155]. For this we need the gradient of q_z:

$$\nabla q_z(x, y, z) = \frac{2}{\beta} \begin{bmatrix} x^{\frac{2}{\alpha}-1}(x^{\frac{2}{\alpha}} + y^{\frac{2}{\alpha}})^{\frac{\alpha}{\beta}-1} \\ y^{\frac{2}{\alpha}-1}(x^{\frac{2}{\alpha}} + y^{\frac{2}{\alpha}})^{\frac{\alpha}{\beta}-1} \\ z^{\frac{2}{\beta}-1} \end{bmatrix} \ .$$

The derivative of $q(t)$ can be computed using the definition of the directional derivative [124]:

$$q'(t) = \mathbf{p}'(t) \cdot \nabla q_z(\mathbf{p}(t)) = \mathbf{d} \cdot \nabla q_z(\mathbf{p}(t)) \ .$$

In ray-tracing it is often helpful to perform a number of checks to rule out the possibility of an intersection, before doing the work to compute it. The first test, and a standard part of instancing, is to test if the ray intersects the superquadric bounding box $[-1, 1]^3$. Next, because $\alpha \leq 1$ and $\beta \leq 1$ (by Equation 3.6), the superquadric glyphs are always convex, which allows three further sequential tests:

1. If $q(t_{\min}) < 0$ and $q(t_{\max}) < 0$, then the ray segment endpoints are within the glyph, and there is no intersection.

2. If $q(t_{\min})q(t_{\max}) > 0$ and $q'(t_{\min})q'(t_{\max}) > 0$, then the segment starts and stops on the same side of the glyphs surface (inside or outside), and the derivative $q'(t)$ must have constant sign within the segment, so there is no intersection.

3. If $q'(t_{\min})q'(t_{\max}) \leq 0$, then $q'(t_{\min}) \leq 0$ and $q'(t_{\max}) \geq 0$ (by convexity), and the Golden Section Search [155] is used to find the $t_m \in [t_{\min}, t_{\max}]$ such that $q(t_m)$ is minimized. If $q(t_m) > 0$, then there is no intersection. Otherwise, assign $t_{\max} \leftarrow t_m$.

At this point, we know there is an intersection within $[t_{\min}, t_{\max}]$, and one iteration of bisection is used to generate an initial guess.

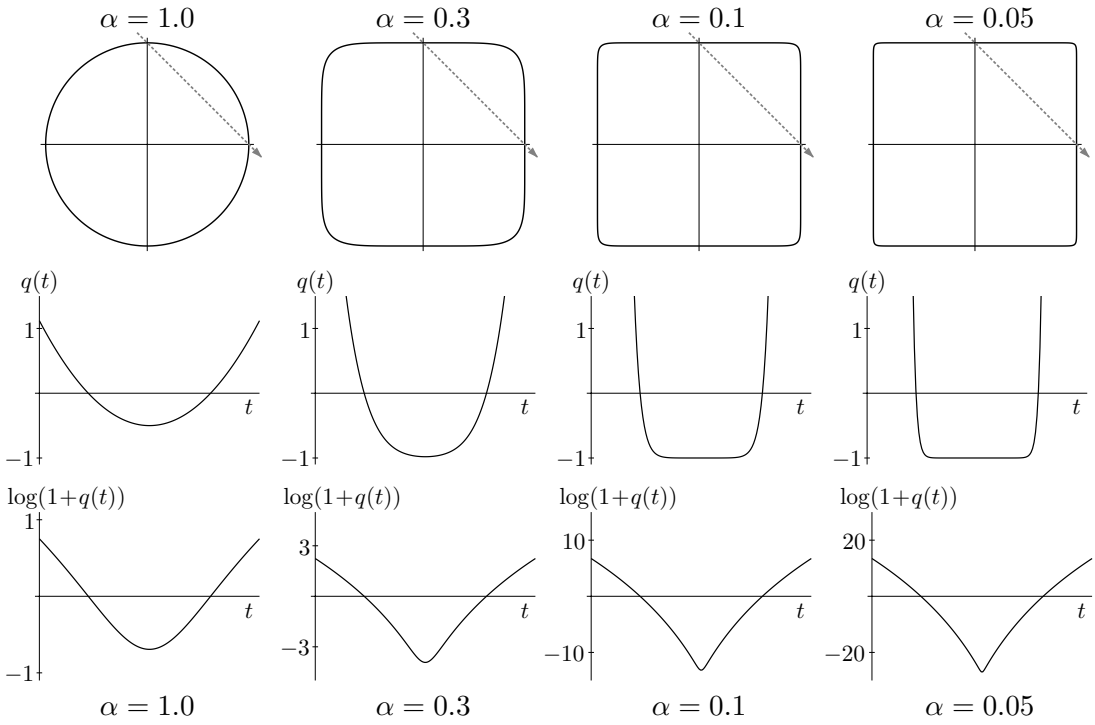


Figure C.1. As α decreases, due to higher anisotropy or a higher setting of γ (Equation 3.6), root-finding by Newton-Raphson becomes more challenging for the function $q(t)$, but not so for $\log(1 + q(t))$.

Using Newton-Raphson to find the root is hampered by the fact that $q(t)$ often exhibits precisely the kind of behavior which causes Newton-Raphson to diverge: steep slopes next to flat regions. Figure C.1 demonstrates how this problem arises in glyphs for anisotropic tensors, especially given a high γ setting. Our solution is to perform root-finding on $\log(1 + q(t))$ rather than $q(t)$. The bottom row of graphs in Figure C.1 shows that the shape of $\log(1 + q(t))$ is much more conducive to iterative root-finding. We evaluate $\log(1+q(t))$ numerically, but use the analytical gradient of $\log(1+q_z(\mathbf{x}))$ to find $d\log(1 + q(t))/dt$:

$$\nabla \log(1 + q_z(x, y, z)) = \begin{bmatrix} \frac{2x^{\frac{2}{\alpha}-1}}{b(x^{2/\alpha}+y^{2/\alpha}+z^{2/\beta}(x^{2/\alpha}+y^{2/\alpha})^{1-\frac{\alpha}{\beta}})} \\ \frac{2y^{\frac{2}{\alpha}-1}}{b(x^{2/\alpha}+y^{2/\alpha}+z^{2/\beta}(x^{2/\alpha}+y^{2/\alpha})^{1-\frac{\alpha}{\beta}}))} \\ \frac{2}{bz(1+z^{-2/\beta}(x^{2/\alpha}+y^{2/\alpha})^{\alpha/\beta})} \end{bmatrix}$$

$$\frac{d}{dt} \log(1 + q(t)) = \mathbf{d} \cdot \nabla \log(1 + q_z(x, y, z)) \; .$$

Although the expressions appear complicated, most of the sub-expressions are repeated, which permits faster evaluation. With this method, three to five iterations of Newton-Raphson are sufficient to find the intersection within acceptable tolerances. This ray-tracing technique is how Figures 3.9 and 3.10 were produced.

APPENDIX D

COMPUTING INVARIANT GRADIENTS

This appendix provides a brief explanation of how the symbolic form of the invariant gradients are derived. The procedure used in the derivations below is presented in [86].

Assuming $\mathbf{D}$ is symmetric, the gradients of the J_i are:

$$\begin{aligned}
\boldsymbol{\nabla} J_1(\mathbf{D}) &= \mathbf{I} \\
\boldsymbol{\nabla} J_2(\mathbf{D}) &= \mathrm{tr}(\mathbf{D})\mathbf{I} - \mathbf{D} \\
\boldsymbol{\nabla} J_3(\mathbf{D}) &= \det(\mathbf{D})\mathbf{D}^{-1} \\
\boldsymbol{\nabla} J_4(\mathbf{D}) &= 2\mathbf{D} \, .
\end{aligned}$$

The basic tool for deriving the relations above is the first-order Taylor expansion of invariant J around tensor $\mathbf{D}$:

$$J(\mathbf{D} + \boldsymbol{\epsilon}) = J(\mathbf{D}) + \boldsymbol{\epsilon} : \boldsymbol{\nabla} J(\mathbf{D}) + O(\boldsymbol{\epsilon}^2) \, .$$

The strategy is to symbolically expand $J(\mathbf{D} + \boldsymbol{\epsilon})$, subtract out $J(\mathbf{D})$ and any $O(\boldsymbol{\epsilon}^2)$ or $O(\boldsymbol{\epsilon}^3)$ terms, and then manipulate what remains until it is of the form $\boldsymbol{\epsilon} : \mathbf{G}$. Then, because $\boldsymbol{\epsilon}$ is arbitrary, $\boldsymbol{\nabla} J = \mathbf{G}$.

Showing $\boldsymbol{\nabla} J_1(\mathbf{D}) = \mathbf{I}$ is easy:

$$\begin{aligned}
J_1(\mathbf{D} + \boldsymbol{\epsilon}) &= \mathrm{tr}(\mathbf{D} + \boldsymbol{\epsilon}) \;=\; \mathrm{tr}(\mathbf{D}) + \mathrm{tr}(\boldsymbol{\epsilon}) \;=\; J_1(\mathbf{D}) + \mathrm{tr}(\boldsymbol{\epsilon}) \\
\Rightarrow \boldsymbol{\epsilon} : \boldsymbol{\nabla} J_1 &= \mathrm{tr}(\boldsymbol{\epsilon}) \;=\; \boldsymbol{\epsilon} : \mathbf{I} \\
\Rightarrow \boldsymbol{\nabla} J_1 &= \mathbf{I} \, .
\end{aligned}$$

To show $\boldsymbol{\nabla} J_2(\mathbf{D}) = \mathrm{tr}(\mathbf{D})\mathbf{I} - \mathbf{D}$:

$$\begin{aligned}
J_2(\mathbf{D} + \boldsymbol{\epsilon}) &= (\mathrm{tr}(\mathbf{D} + \boldsymbol{\epsilon})^2 - \mathrm{tr}((\mathbf{D} + \boldsymbol{\epsilon})^2))/2 \\
&= (\mathrm{tr}(\mathbf{D})^2 + 2\,\mathrm{tr}(\mathbf{D})\,\mathrm{tr}(\boldsymbol{\epsilon}) + \mathrm{tr}(\boldsymbol{\epsilon})^2 - \mathrm{tr}(\mathbf{D}^2) - 2\,\mathrm{tr}(\mathbf{D}\boldsymbol{\epsilon}) - \mathrm{tr}(\boldsymbol{\epsilon}^2))/2 \\
&= J_2(\mathbf{D}) + \mathrm{tr}(\mathbf{D})\,\mathrm{tr}(\boldsymbol{\epsilon}) - \mathrm{tr}(\mathbf{D}\boldsymbol{\epsilon}) + O(\boldsymbol{\epsilon}^2) \\
\Rightarrow \boldsymbol{\epsilon} : \boldsymbol{\nabla} J_2 &= \mathrm{tr}(\mathbf{D})\,\mathrm{tr}(\boldsymbol{\epsilon}) - \mathrm{tr}(\mathbf{D}\boldsymbol{\epsilon}) \\
&= \mathrm{tr}(\mathbf{D})(\boldsymbol{\epsilon} : \mathbf{I}) - \mathbf{D}^{\mathsf{t}} : \boldsymbol{\epsilon} \\
&= \boldsymbol{\epsilon} : (\mathrm{tr}(\mathbf{D})\mathbf{I}) - \boldsymbol{\epsilon} : \mathbf{D}^{\mathsf{t}} \\
\Rightarrow \boldsymbol{\nabla} J_2 &= \mathrm{tr}(\mathbf{D})\mathbf{I} - \mathbf{D}^{\mathsf{t}} \, .
\end{aligned}$$

If $\mathbf{D}$ is symmetric, then $\boldsymbol{\nabla} J_2 = \mathrm{tr}(\mathbf{D})\mathbf{I} - \mathbf{D}$.

To show $\boldsymbol{\nabla} J_4(\mathbf{D}) = 2\mathbf{D}$:

$$\begin{aligned}
J_4(\mathbf{D}+\boldsymbol{\epsilon}) &= \|\mathbf{D}+\boldsymbol{\epsilon}\|^2 \\
&= \left(\sqrt{(\mathbf{D}+\boldsymbol{\epsilon}) : (\mathbf{D}+\boldsymbol{\epsilon})}\right)^2 \\
&= \mathrm{tr}((\mathbf{D}+\boldsymbol{\epsilon})^{\mathsf{t}}(\mathbf{D}+\boldsymbol{\epsilon})) \\
&= \mathrm{tr}(\mathbf{D}^{\mathsf{t}}\mathbf{D} + \mathbf{D}^{\mathsf{t}}\boldsymbol{\epsilon} + \boldsymbol{\epsilon}^{\mathsf{t}}\mathbf{D} + \boldsymbol{\epsilon}^{\mathsf{t}}\boldsymbol{\epsilon}) \\
&= J_4(\mathbf{D}) + \mathrm{tr}(\mathbf{D}^{\mathsf{t}}\boldsymbol{\epsilon}) + \mathrm{tr}(\boldsymbol{\epsilon}^{\mathsf{t}}\mathbf{D}) + O(\boldsymbol{\epsilon}^2) \\
\Rightarrow \boldsymbol{\epsilon} : \boldsymbol{\nabla} J_4 &= \mathbf{D} : \boldsymbol{\epsilon} + \boldsymbol{\epsilon} : \mathbf{D} \\
\Rightarrow \boldsymbol{\nabla} J_4 &= 2\mathbf{D} \; .
\end{aligned}$$

Showing $\boldsymbol{\nabla} J_3(\mathbf{D}) = \det(\mathbf{D})\mathbf{D}^{-1}$ is a bit more involved:

$$\begin{aligned}
J_3(\mathbf{D}+\boldsymbol{\epsilon}) &= \det(\mathbf{D}+\boldsymbol{\epsilon}) = \det(\mathbf{D}+\mathbf{I}\boldsymbol{\epsilon}) \\
&= \det(\mathbf{D}+\mathbf{D}\mathbf{D}^{-1}\boldsymbol{\epsilon}) = \det(\mathbf{D}(\mathbf{I}+\mathbf{D}^{-1}\boldsymbol{\epsilon})) \\
&= \det(\mathbf{D})\det(\mathbf{I}+\mathbf{D}^{-1}\boldsymbol{\epsilon}) \\
&= -\det(\mathbf{D})\det(-\mathbf{I}-\mathbf{D}^{-1}\boldsymbol{\epsilon}) \; .
\end{aligned}$$

This may seem somewhat contrived, but now $\det(-\mathbf{I}-\mathbf{D}^{-1}\boldsymbol{\epsilon})$ is the characteristic polynomial of $\mathbf{D}^{-1}\boldsymbol{\epsilon}$, evaluated at $\lambda = -1$:

$$\begin{aligned}
\det(\lambda\mathbf{I}-\mathbf{D}^{-1}\boldsymbol{\epsilon}) &= \lambda^3 - J_1(\mathbf{D}^{-1}\boldsymbol{\epsilon})\lambda^2 + J_2(\mathbf{D}^{-1}\boldsymbol{\epsilon})\lambda - J_3(\mathbf{D}^{-1}\boldsymbol{\epsilon}) \\
&= \lambda^3 - \mathrm{tr}(\mathbf{D}^{-1}\boldsymbol{\epsilon})\lambda^2 + (\mathrm{tr}(\mathbf{D}^{-1}\boldsymbol{\epsilon})^2 - \mathrm{tr}(\mathbf{D}^{-2}\boldsymbol{\epsilon}^2))\lambda/2 - \det(\mathbf{D}^{-1}\boldsymbol{\epsilon}) \\
\Rightarrow \det(-\mathbf{I}-\mathbf{D}^{-1}\boldsymbol{\epsilon}) &= -1 - \mathrm{tr}(\mathbf{D}^{-1}\boldsymbol{\epsilon}) + O(\boldsymbol{\epsilon}^2) + O(\boldsymbol{\epsilon}^3) \; .
\end{aligned}$$

This used the fact that the determinant is cubic in its argument. The higher order terms in $\boldsymbol{\epsilon}$ can now be dropped:

$$\begin{aligned}
J_3(\mathbf{D}+\boldsymbol{\epsilon}) = -\det(\mathbf{D})\det(-\mathbf{I}-\mathbf{D}^{-1}\boldsymbol{\epsilon}) &= -\det(\mathbf{D})(-1-\mathrm{tr}(\mathbf{D}^{-1}\boldsymbol{\epsilon}) \\
&= \det(\mathbf{D}) + \det(\mathbf{D})\,\mathrm{tr}(\mathbf{D}^{-1}\boldsymbol{\epsilon}) \\
&= J_3(\mathbf{D}) + \det(\mathbf{D})((\mathbf{D}^{-1})^{\mathsf{t}} : \boldsymbol{\epsilon}) \\
\Rightarrow \boldsymbol{\epsilon} : \boldsymbol{\nabla} J_3 &= \boldsymbol{\epsilon} : (\det(\mathbf{D})(\mathbf{D}^{-1})^{\mathsf{t}}) \\
\Rightarrow \boldsymbol{\nabla} J_3 &= \det(\mathbf{D})(\mathbf{D}^{-1})^{\mathsf{t}} \; .
\end{aligned}$$

Then, if $\mathbf{D}$ is symmetric, so is $\mathbf{D}^{-1}$, in which case $\boldsymbol{\nabla} J_3 = \det(\mathbf{D})\mathbf{D}^{-1}$.

REFERENCES

[1] D Akers, A Sherbondy, R Mackenzie, R Dougherty, and B Wandell. Exploration of the brain's white matter pathways with dynamic queries. In *Proceedings of IEEE Visualization 2004*, page (to appear), October 2004.

[2] AL Alexander, K Hasan, G Kindlmann, DL Parker, and JS Tsurada. A geometric analysis of diffusion tensor measurements of the human brain. *Magnetic Resonance in Medicine*, 44:283–291, 2000.

[3] AL Alexander, KM Hasan, M Lazar, JS Tsuruda, and DL Parker. Analysis of partial volume effects in diffusion-tensor MRI. *Magnetic Resonance in Medicine*, 45:770–780, 2001.

[4] DC Alexander, GJ Barker, and SR Arridge. Detection and modeling of non-gaussian apparent diffusion coefficients profiles in human brain data. *Magnetic Resonance in Medicine*, 48:331–340, 2002.

[5] MM Bahn. Invariant and orthonormal scalar measures derived from magnetic resonance diffusion tensor imaging. *Journal of Magnetic Resonance*, 141:68–77, 1999.

[6] MM Bahn. A linear relationship exists among brain diffusion eigenvalues measured by diffusion tensor magnetic resonance imaging. *Journal of Magnetic Resonance*, 137:33–38, 1999.

[7] R Bajcsy and F Solina. Three dimensional object representation revisited. In *Proceedings of First International Conference on Computer Vision*, pages 231–240, London, England, 1987. IEEE.

[8] DC Banks. Illumination in diverse codimensions. In *Proceedings of ACM SIGGRAPH 94*, volume 28, pages 327–334. Addison Wesley, 1994.

[9] J Bard and I Lauder. How well does Turing's theory of morphogenesis work? *Journal of Theoretical Biology*, 45(2):501–531, June 1974.

[10] A Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 18(1):11–23, 1981.

[11] AH Barr. The Einstein Summation Notation: Introduction and extensions. In *ACM SIGGRAPH 89 Course Notes #30: Topics in Physically Based Modeling*, pages J1–J12, 1989.

[12] PJ Basser. New histological and physiological stains derived from diffusion-tensor mr images. *Annals of the New York Academy of Sciences*, 820:123–138, 1997.

[13] PJ Basser. Fiber-tractography via diffusion tensor MRI (DT-MRI). In *Proceedings of the 6th Annual Meeting of ISMRM*, page 1226, 1998.

[14] PJ Basser and DK Jones. Diffusion-tensor MRI: Theory, experimental design and data analysis - A technical review. *Nuclear Magnetic Resonance in Biomedicine*, 15:456–467, 2002.

[15] PJ Basser, J Mattiello, and D Le Bihan. Estimation of the effective self-diffusion tensor from the NMR spin-echo. *Journal of Magnetic Resonance, B*, 103(3):247–254, 1994.

[16] PJ Basser, J Mattiello, and D Le Bihan. MR diffusion tensor spectroscopy and imaging. *Biophysics Journal*, 66(1):259–267, 1994.

[17] PJ Basser and S Pajevic. Statistical artifacts in diffusion tensor MRI (DT-MRI) caused by background noise. *Magnetic Resonance Medicine*, 44:41–50, 2000.

[18] PJ Basser and S Pajevic. Dealing with uncertainty in diffusion tensor MR data. *Isreal Journal of Chemistry*, 43:128–144, 2003.

[19] PJ Basser and S Pajevic. A normal distribution for tensor-valued random variables: Applications to diffusion tensor MRI. *IEEE Transactions on Medical Imaging*, 22(7):785–794, July 2003.

[20] PJ Basser, S Pajevic, C Pierpaoli, J Duda, and A Aldroubi. In vivo fiber tractograpy using DT-MRI data. *Magnetic Resonance in Medicine*, 44:625–632, 2000.

[21] PJ Basser and C Pierpaoli. Microstructural and physiological features of tissues elucidated by quantitative-diffusion-tensor MRI. *Journal of Magnetic Resonance, Series B*, 111:209–219, 1996.

[22] PJ Basser and C Pierpaoli. A simplified method to measure the diffusion tensor from seven MR images. *Magnetic Resonance in Medicine*, 39:928–923, 1998.

[23] ME Bastin, PA Armitage, and I Marshall. A theoretical study of the effect of experimental noise on the measurement of anisotropy in diffusion imaging. *Magnetic Resonance Imaging*, 16(7):773–785, 1998.

[24] C Beaulieu. The basis of anisotropic water diffusion in the nervous system - A technical review. *Nuclear Magnetic Resonance in Biomedicine*, 15:435–455, 2002.

[25] PN Belhumeur, DJ Kriegman, and AL Yuille. The Bas-Relief ambiguity. *International Journal of Computer Vision*, 35(1):33–44, November 1999.

[26] AG Belyaev, A Pasko, and TL Kunii. Ridges and ravines on implicit surfaces. In *Computer Graphics International '98*, pages 530–535, 1998.

[27] MJ Bentum, BBA Lichtenbelt, and T Malzbender. Frequency analysis of gradient estimators in volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 2(3):242–254, September 1996.

[28] D Le Bihan. Molecular diffusion, tissue microdynamics and microstructure. *Nuclear Magnetic Resonance in Biomedicine*, 8:375–386, 1995.

[29] D Le Bihan, R Tuner, and P Douek. Is water diffusion restricted in human brain white matter? An echo-planar NMR imaging study. *Neuroreport*, 4:887–890, 1993.

[30] M Bilgen, I Elshafiey, and PA Narayana. Mohr diagram representation of anisotropic diffusion tensor in MRI. *Magnetic Resonance in Medicine*, 47:823–827, 2002.

[31] M Bilgen and PA Narayana. Mohr diagram interpretation of anisotropic diffusion indices in MRI. *Magnetic Resonance Imaging*, 21:567–572, 2003.

[32] BioPSE: Problem Solving Environment for modeling, simulation, and visualization of bioelectric fields. Scientific Computing and Imaging Institute (SCI), http://software.sci.utah.edu/biopse.html, 2002.

[33] G Birkhoff. *A Survey of Modern Algebra*, chapter 2. Macmillan, New York, New York, 3rd edition, 1965.

[34] JF Blinn. Models of light reflection for computer synthesized pictures. *Computer Graphics*, 11:192–198, July 1977.

[35] WM Boothby. *An Introduction to Differentiable Manifolds and Riemmanian Geometry*. Academic Press, 2nd edition, 2001.

[36] DE Bourne and PC Kendall. *Vector Analysis and Cartesian Tensors*, chapter 9. Thomas Nelson and Sons, Ltd, Great Britain, 2nd edition, 1977.

[37] B Cabral and C Leedom. Imaging vector fields using line integral convolution. In *Proceedings of of ACM SIGGRAPH 1993*, pages 263–270. Addison Wesley, 1993.

[38] PT Callaghan. *Principles of Nuclear Magnetic Resonance Microscopy*. Oxford University Press, Inc., New York, New York, 1991.

[39] HBG Casimir. On Onsager's principle of microscopic reversibility. *Reviews of Modern Physics*, 17(2,3), April-July 1945.

[40] M Catani, DK Jones, R Donato, and DH ffytche. Occipito-temporal connections in the human brain. *Brain*, 126:2093–2107, 2003.

[41] T Conturo, N Lori, T Cull, E Akbudak, A Snyder, J Shimony, R McKinstry, H Burton, and M Raichle. Tracking neuronal fiber pathways in the living human brain. *Proceedings of the National Academy of Sciences*, 96:10422–10427, 1999.

[42] TE Conturo, RC McKinstry, E Akbudak, and BH Robinson. Encoding of anisotropic diffusion with tetrahedral gradients: A general mathematical diffusion formalism and experimental results. *Magnetic Resonance in Medicine*, 35(3):399–412, 1996.

[43] HSM Coxeter. *Introduction to Geometry*. John Wiley and Sons, Inc., New York, New York, 1969.

[44] J Crank. *The Mathematics of Diffusion*. Oxford University Press, Oxford, England, 1975.

[45] R Crawfis and N Max. Direct volume visualization of three-dimensional vector fields. In *Proceedings of 1992 Workshop on Volume Visualization*, pages 55–60, Boston, Massachusetts, 1992.

[46] R Crawfis and N Max. Texture splats for 3D scalar and vector field visualization. In *Proceedings of IEEE Visualization '93*, pages 261–266, 1993.

[47] JC Criscione, JD Humphrey, AS Douglas, and WC Hunter. An invariant basis for natural strain which yields orthogonal stress response terms in isotropic hyperelasticity. *Journal of Mechanics and Physics of Solids*, 48:2445–2465, 2000.

[48] J Damon. Generic structure of two-dimensional images under Gaussian blurring. *SIAM Journal on Applied Mathematics*, 59(1):97–138, 1998.

[49] DA Danielson. *Vectors and Tensors in Engineering and Physics*. Westview Press, Boulder, Colorado, 2nd edition, 2003.

[50] WC de Leeuw and JJ van Wijk. Enhanced spot noise for vector field visualization. In *Proceedings of IEEE Visualization '95*, pages 233–239. IEEE Computer Society Press, 1995.

[51] T. Delmarcelle and L. Hesselink. Visualization of second order tensor fields and matrix data. In *Proceedings of IEEE Visualization '92*, pages 316–323, 1992.

[52] T Delmarcelle and L Hesselink. Visualizing second-order tensor fields with hyper streamlines. *IEEE Computer Graphics and Applications*, 13(4):25–33, 1993.

[53] T Delmarcelle and L Hesselink. A unified framework for flow visualization. In R. S. Gallagher, editor, *Computer visualization: Graphics techniques for scientific and engineering analysis*, pages 129–170. CRC Press, 1995.

[54] Ç Demiralp, DH Laidlaw, C Jackson, D Keefe, and S Zhang. Subjective usefulness of CAVE and fishtank VR display systems for a scientific visualization application. In *IEEE Visualization Poster Compendium*, 2003.

[55] J Dou, W-YI Tseng, TG Reese, and VJ Wedeen. Combined diffusion and strain MRI reveals structure and function of human myocardial laminar sheets in vivo. *Magnetic Resonance in Medicine*, 50:107–113, 2003.

[56] RA Drebin, L Carpenter, and P Hanrahan. Volume rendering. *Computer Graphics*, 22(4):65–74, 1988.

[57] DS Dummit and RM Foote. *Abstract Algebra*. Prentice-Hall, Inc., New Jersey, 1991.

[58] D Ebert and P Rheingans. Volume illustration: Non-photorealistic rendering of volume data. In *Proceedings of IEEE Visualization 2000*, pages 195–202, 2000.

[59] DS Ebert and CD Shaw. Minimally immersive flow visualization. *IEEE Transac-*

tions on Visualization and Computer Graphics, 7(4):343–350, 2001.

[60] A Einstein. *Investigations on the Theory of Brownian Motion*. Dover Publications, New York, 1956. R Fürthe, editor. AD Cowper, translator.

[61] DB Ennis, G Kindlmann, PA Helm, I Rodriguez, H Wen, and ER McVeigh. Visualization of high-resolution myocardial strain and diffusion tensors using superquadric glyphs. In *Proceedings of the 12th Annual Meeting of International Society for Magnetic Resonance in Medicine (ISMRM)*, May 2004.

[62] R Fernando, editor. *GPU Gems: Programming Techniques, Tips, and Trick for Real-Time Graphics*. Addison-Wesley, Boston, Massachusetts, 2004.

[63] J Foley, A van Dam, S Feiner, and J Hughes. *Computer Graphics Principles and Practice*, pages 592–595, 722–731. Addison-Wesley, second edition, 1990.

[64] LR Frank. Anisotropy in high angular resolution diffusion-weighted MRI. *Magnetic Resonance in Medicine*, 45:935–939, 2001.

[65] LR Frank. Characterization of anisotropy in high angular resolution diffusion-weighted MRI. *Magnetic Resonance in Medicine*, 47:1083–1099, 2002.

[66] T Frühauf. Raycasting vector fields. In *Proceedings of IEEE Visualization '96*, pages 115–120, 1996.

[67] M Gardner. The superellipse: A curve that lies between the ellipse and the rectangle. *Scientific American*, 213(3):222–234, 1965.

[68] C Garth, X Tricoche, T Salzbrunn, T Bobach, and G Scheuermann. Surface techniques for vortex visualization. In *Proceedings of Joint Eurographics IEEE TCVG Symposium on Visualization*, pages 155–164,346, Konstanz, Germany, May 2004.

[69] A Glassner. *An Introduction to Ray Tracing*. Morgan Kaufmann, San Fransisco, California, 1989.

[70] GH Golub and CF Van Loan. *Matrix Computations*, chapter 7. Johns Hopkins University Press, Baltimore and London, 1996.

[71] RC Gonzalez and RE Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, Reading, MA, 2nd edition, 2002.

[72] AM Goodbody. *Cartesian Tensors with Applications to Mechanics, Fluid Mechanics, and Elasticity*. Ellis Horwood, Ltd., Chichester, England, 1982.

[73] H Gray, LH Bannister, MM Berry, and PL Williams. *Gray's Anatomy: The Anatomical Basis of Medicine and Surgery*. Churchill Livingstone, Oxford, United Kingdom, 38th edition, 1995.

[74] JV Guadagno, C Calautti, and J-C Baron. Progress in imaging stroke: Emerging clinical applications. *British Medical Bulletin*, 65:145–157, 2003.

[75] P Hagmann, J-P Thiran, L Jonasson, P Vandergheynst, S Clarke, P Maeder, and R Meuli. DTI mapping of human brain connectivity: Statistical fibre tracking and virtual dissection. *NeuroImage*, 19:545–554, 2003.

[76] C Hansen and CR Johnson, editors. *Section IV: Tensor Field Visualization*, page (to appear). Academic Press, 2004.

[77] C Hansen and CR Johnson, editors. *Topological Methods for Tensor Visualization*, page (to appear). Academic Press, 2004.

[78] KM Hasan, PJ Basser, DL Parker, and AL Alexander. Analytical computation of the eigenvalues and eigenvectors in DT-MRI. *Journal of Magnetic Resonance*, 152:41–47, 2001.

[79] KM Hasan, DL Parker, and AL Alexander. Comparison of optimization procedures for diffusion-tensor encoding directions. *Journal of Magnetic Resonance Imaging*, 13:769–780, 2001.

[80] CG Healey, V Interrante, D Kremers, DH Laidlaw, and P Rheingans. Nonphotorealistic rendering in scientific visualization. In *Course Notes of ACM SIGGRAPH 2001, Course 32*, 2001.

[81] RG Henry, JI Berman, SS Nagarajan, P Mukherjee, and MS Berger. Subcortical pathways serving cortical language sites: Initial experience with diffusion tensor imaging fiber tracking combined with intraoperative language mapping. *NeuroImage*, 21:616–622, 2004.

[82] L Hesselink, Y Levy, and Y Lavin. The topology of symmetric, second-order 3D tensor fields. *IEEE Transactions on Visualization and Computer Graphics*, 3(1), jan–mar 1997.

[83] L Hesselink, F Post, and JJ van Wijk. Research issues in vector and tensor field visualization. *IEEE Computer Graphics and Applications*, 14:76–79, 1994.

[84] K Hoffman and R Kunze. *Linear Algebra*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1971.

[85] J Hollerbach. Personal Communication, June 2004. Third Floor Central Hallway, Merrill Engineering Building.

[86] GA Holzapfel. *Nonlinear Solid Mechanics*, chapter 1. John Wiley and Sons, Ltd, England, 2000.

[87] MA Horsfield and DK Jones. Applications of diffusion-weighted and diffusion tensor MRI to white matter diseases - a review. *NMR in Biomedecine*, 15:570–577, 2002.

[88] EW Hsu, AL Muzikant, SA Matulevicius, RC Penland, and CS Henriquez. Magnetic resonance myocardial fiber-orientation mapping with direct histological correlation. *American Journal of Physiology*, 274:1627–1634, 1998.

[89] V Interrante. Illustrating surface shape in volume data via principal direction-

driven 3D line integral convolution. In *Proceedings of ACM SIGGRAPH '97*, volume 31, pages 109–116, 1997.

[90] V Interrante and C Grosch. Strategies for effectively visualizing 3D flow with volume LIC. In *Proceedings of IEEE Visualization '97*, pages 421–424, Phoenix, Arizona, 1997. IEEE Computer Society Press.

[91] A Jaklic, A Leonardis, and F Solina. *Segmentation and Recovery of Superquadrics*, volume 20 of *Computational Imaging and Vision*. Kluwer, Dordrecth, The Netherlands, 2000.

[92] J Jeong and F Hussain. On the identification of a vortex. *Journal of Fluid Mechanics*, 285:69–94, 1995.

[93] CR Johnson and AR Sanderson. A next step: Visualizing errors and uncertainty. *IEEE Computer Graphics and Applications*, 23(5):6–10, September/October 2003.

[94] DK Jones. The effect of gradient sampling schemes on measures derived from diffusion tensor MRI: A monte carlo study. *Magnetic Resonance in Medicine*, 51:807–815, 2004.

[95] JT Kajiya. An overview and comparison of rendering methods. *A Consumer's and Developer's Guide to Image Synthesis*, pages 259–263, 1988. ACM Siggraph '88 Course 12 Notes.

[96] D Karelitz, D Keefe, and DH Laidlaw. Using CavePainting to create scientific visualizations. IEEE Visualization 2003 Poster Compendium, October 2003.

[97] D Keefe, D Karelitz, E Vote, and DH Laidlaw. Artistic collaboration in designing VR visualizations. *IEEE Computer Graphics and Applications*, page (to appear), 2004/2005.

[98] D-S Kim, M Kim, I Ronen, E Formisano, K-H Kim, K Ugurbil, S Mori, and R Goebel. In vivo mapping of functional domains and axonal connectivity in cat visual cortex using magnetic resonance imaging. *Magnetic Resonance Imaging*, 21:1131–1140, 2003.

[99] G Kindlmann. Semi-automatic generation of transfer functions for direct volume rendering. Master's thesis, Cornell University, Ithaca, NY, January 1999.

[100] G Kindlmann. Superquadric tensor glyphs. In *Proceedings of IEEE TVCG/EG Symposium on Visualization 2004*, pages 147–154, May 2004.

[101] G Kindlmann and JW Durkin. Semi-automatic generation of transfer functions for direct volume rendering. In *IEEE Symposium On Volume Visualization*, pages 79–86, 1998.

[102] G Kindlmann and D Weinstein. Hue-balls and lit-tensors for direct volume rendering of diffusion tensor fields. In *Proceedings of IEEE Visualization '99*, pages 183–189, 1999.

[103] G Kindlmann, D Weinstein, and DA Hart. Strategies for direct volume rendering of diffusion tensor fields. *IEEE Transactions on Visualization and Computer Graphics*, 6(2):124–138, April-June 2000.

[104] G Kindlmann, DM Weinstein, AD Lee, AW Toga, and PM Thompson. Visualization of anatomic covariance tensor fields. In *Proceedings of the 26th Annual Conference of the IEEE Engineering in Medicine and Biology Society (EMBS)*, September 2004.

[105] G Kindlmann, R Whitaker, T Tasdizen, and T Möller. Curvature-based transfer functions for direct volume rendering: Methods and applications. In *Proceedings of IEEE Visualization 2003*, pages 513–520, October 2003.

[106] M Kirby, H Marmanis, and DH Laidlaw. Visualizing multivalued data from 2D incompressible flows using concepts from painting. In *Proceedings of IEEE Visualization '99*, pages 333–340, 1999.

[107] J Kniss, G Kindlmann, and C Hansen. Interactive volume rendering using multidimensional transfer functions and direct manipulation widgets. In *Proceedings of IEEE Visualization 2001*, pages 255–262, October 2001.

[108] J Kniss, G Kindlmann, and C Hansen. Multidimensional transfer functions for interactive volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):270–285, July-September 2002.

[109] D Laidlaw. *Geometric Model Extraction from Magnetic Resonance Volume Data*. PhD thesis, California Institute of Technology, May 1995.

[110] DH Laidlaw, ET Ahrens, D Kremers, MJ Avalos, RE Jacobs, and C Readhead. Visualizing diffusion tensor images of the mouse spinal cord. In *Proceedings of IEEE Visualization '98*, pages 127–134, 1998.

[111] DH Laidlaw, D Kremers, F Frankel, V Interrante, and TF Banchoff. Art and visualization: Oil and water? In *Proceedings of IEEE Visualization '98*, pages 507–509, 1998.

[112] OS Lawlor. Reaction-diffusion textures. CS 497 project write-up, March 2004. ⟨http://charm.cs.uiuc.edu/users/olawlor/projects/2003/rd/⟩.

[113] M Lazar, D Weinstein, K Hasan, and AL Alexander. Axon tractography with tensorlines. In *Proceedings of the 8th Annual Meeting of ISMRM*, page 481, 2000.

[114] M Lazar, DM Weinstein, JS Tsuruda, KM Hasan, K Arfanakis, E Meyer, B Badie, HA Rowley, V Haughton, A Field, and AL Alexander. White matter tractography using diffusion tensor deflection. *Human Brain Mapping*, 18:306–321, 2003.

[115] IJ LeGrice, BH Smaill, LZ Chai, SG Edgar, JB Gavin, and PJ Hunter. Laminar structure of the heart. I: Ventricular myocyte arrangement and connective tissue architecture in the dog. *American Journal of Physiology*, 269:H571–H582, 1995.

[116] M Levoy. Display of surfaces from volume data. *IEEE Computer Graphics &*

Applications, 8(5):29–37, 1988.

[117] G-S Li, UD Bordoloi, and H-W Shen. Chameleon: An interactive texture-based rendering framework for visualizing three-dimensional vector fields. In *Proceedings of IEEE Visualization 2003*, pages 241–248, Seattle, Washington, 2003.

[118] Z-P Liang and PC Lauterbur. *Principles of Magnetic Resonance Imaging: A Signal Processing Perspective*. Wiley-IEEE Press, 1999.

[119] B Lichtenbelt, R Crane, and S Naqvi. *Introduction to Volume Rendering*. Prentice Hall PTR, Upper Saddle River, New Jersey, 1998.

[120] HJ Lipkin. *Lie Groups for Pedestrians*. Dover Publications, 2002.

[121] WE Lorensen and H . Cline. Marching cubes: A high resolution 3D surface construction algorithm. *Computer Graphics*, 21(4):163–169, July 1987.

[122] N Makris, AJ Worth, G Sorensen, GM Papadimitriou, O Wu, TK Reese, VJ Wedeen, TL Davis, JW Stakes, VS Caviness, E Kaplan, BR Rosen, DN Pandya, and DN Kennedy. Morphometry of in vivo human white matter association pathways with diffusion weighted MRI. *Annals of Neurology*, 42(6):951–962, 1997.

[123] SR Marschner and RJ Lobb. An evaluation of reconstruction filters for volume rendering. In RD Bergeron and AE Kaufman, editors, *Proceedings of IEEE Visualization '94*, pages 100–107, 1994.

[124] JE Marsden and AJ Tromba. *Vector Calculus*, chapter 2.6, 4.2. W.H. Freeman and Company, New York, New York, 1996.

[125] N Max, B Becker, and R Crawfis. Flow volumes for interactive vector field visualization. In *Proceedings of IEEE Visualization '93*, pages 19–24, 1993.

[126] N Max, R Crawfis, and C Grant. Visualizing 3D velocity fields near contour surfaces. In *Proceedings of IEEE Visualization '94*, pages 248–255. IEEE Computer Society Press, 1994.

[127] JM McCarthy. *An Introduction to Theoretical Kinematics*. Massachusetts Institute of Technology, Massachusetts, 1990.

[128] D Mitchell and A Netravali. Reconstruction filters in computer graphics. In *Proceedings of ACM SIGGRAPH '88*, pages 221–228, August 1988.

[129] T Möller, R Machiraju, and K Mueller R Yagel. Evaluation and design of filters using a taylor series expansion. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):184–199, April 1997.

[130] T Möller, K Müller, Y Kurzion, R Machiraju, and R Yagel. Design of accurate and smooth filters for function and derivative reconstruction. In *Proc. IEEE Symposium on Volume Visualization 1998*, pages 143–151, 1998.

[131] O Monga, S Benayoun, and O Faugeras. From partial derivatives of 3D density images to ridge lines. In *IEEE Conference on Vision and Pattern Recognition*

(CVPR), Urbana Champaign, June 1992.

[132] S Mori, BJ Crain, VP Chacko, and PCM van Zijl. Three dimensional tracking of axonal projections in the brain by magnetic resonance imaging. *Annals Neurology*, 45:265–269, 1999.

[133] S Mori and PC Van Zijl. Fiber tracking: Principles and strategies - A technical review. *Nuclear Magnetic Resonance in Biomedicine*, 15:468–480, 2002.

[134] ME Moseley, Y Cohen, J Mintorovitch, L Chileuitt, H Shimizu, J Kucharczyk, MF Wendland, and PR Weinstein. Early detection of regional cerebral ischemia in cats: Comparison of diffusion- and T2-weighted MRI and spectroscopy. *Magnetic Resonance in Medicine*, 14(2):330–346, May 1990.

[135] JD Murray. *Mathematical Biology*, chapter 9, 14, 15. Springer-Verlag, Berlin, 1993.

[136] RWD Nickalls. A new approach to solving the cubic: Cardan's solution revealed. *The Mathematical Gazette*, 77:354–359, November 1993.

[137] L Onsager. Reciprocal relations in irreversible processes, I. *Physical Review*, 37, 1931.

[138] L Onsager. Reciprocal relations in irreversible processes, II. *Physical Review*, 38, 1931.

[139] J Ou and E Hsu. Generalized line integral convolution rendering of diffusion tensor fields. In *Proceedings of the 9th Scientific Meeting and Exhibition of the International Society for Magnetic Resonance in Medicine (ISMRM)*, Glasgow, United Kingdom, 2001.

[140] S Pajevic, A Aldroubi, and PJ Basser. A continuous tensor field approximation of discrete DT-MRI data for extracting microstructural and architectural features of tissue. *Journal of Magnetic Resonance*, 154:85–100, 2002.

[141] S Pajevic and PJ Basser. Parametric and non-parametric statistical analysis of DT-MRI data. *Journal of Magnetic Resonance*, 161:1–14, 2003.

[142] S Pajevic and C Pierpaoli. Color schemes to represent the orientation of anisotropic tissues from diffusion tensor data: Application to white matter fiber tract mapping in the human brain. *Magnetic Resonance in Medicine*, 42(3):526–540, 1999.

[143] A.J. Parker, C. Christou, B.G. Cumming, E.B. Johnston, M.J. Hawken, and A. Zisserman. The analysis of 3D shape: Psychological principles and neural mechanisms. In G.W. Humphreys, editor, *Understanding Vision*, chapter 8. Blackwell, Oxford, 1992.

[144] S Parker, M Parker, Y Livnat, P-P Sloan, CD Hansen, and P Shirley. Interactive ray tracing for volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 5(3):238–250, July-September 1999.

[145] S Parker, P Shirley, Y Livnat, C Hansen, and P-P Sloan. Interactive ray tracing

for isosurface extraction. In *Proceedings of IEEE Visualization '98*, pages 233–238, October 1998.

[146] DA Patterson and JL Hennessy. *Computer Architecture: A Quantitative Approach*, chapter 5. Morgan Kaufmann Publishers, Inc., San Fransisco, CA, 3rd edition, 2003.

[147] JE Pearson. Complex patterns in a simple system. *Science*, 261, July 1993.

[148] AP Pentland. Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28(2):293–331, 1986.

[149] AP Pentland. Recognition by parts. In *Proceedings of First International Conference on Computer Vision*, pages 612–620, London, England, 1987. IEEE.

[150] D Pickens. Magnetic resonance imaging. In J Beutel, HL Kundel, and RL van Metter, editors, *Handbook of Medical Imaging*, pages 373–461. Society of Photo-Optical Instrumentation Engineers (SPIE), Bellingham, Washington, 2000. Volume 1: Physics and Psychophysics.

[151] C Pierpaoli. Oh no! One more method for color mapping of fiber tract direction using diffusion MR imaging data. In *Proceedings of the 5th Annual Meeting of ISMRM*, page 1741, 1997.

[152] C Pierpaoli and PJ Basser. Toward a quantitative assessment of diffusion anisotropy. *Magnetic Resonance in Medicine*, 33:893–906, 1996.

[153] C Pierpaoli, P Jezzard, PJ Basser, A Barnett, and G DiChiro. Diffusion tensor MR imaging of the human brain. *Radiology*, 201(3):637–648, 1996.

[154] FJ Post, FH Post, T van Walsum, and D Silver. Iconic techniques for feature visualization. In *Proceedings of IEEE Visualization '95*, pages 288–295, 1995.

[155] WH Press, BP Flannery, SA Teukolsky, and WT Vetterling. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, Cambridge, United Kingdom, 2nd edition, 1992.

[156] TG Reese, O Heid, RM Weisskoff, and VJ Wedeen. Reduction of eddy-current-induced distortion in diffusion MRI using a twice-refocused spin echo. *Magnetic Resonance in Medicine*, 49:177–182, 2003.

[157] C Rezk-Salama, P Hastreiter, T Christian, and T Ertl. Interactive exploration of volume line integral convolution based on 3D-texture mapping. In *Proceedings of IEEE Visualization '99*, pages 233–240, 1999.

[158] TM Rhyne, DH Laidlaw, CG Healey, V Interrante, and D Duke. Realism, expressionism, and abstraction. In *Proceedings of IEEE Visualization '01*, pages 523–526, 2001.

[159] L Rosenblum, R Earnshaw, J Encarnacao, H Hagen, A Kaufman, SV Klimenko, G Nielson, F Post, and D Thalmann. *Scientific Visualization: Advances and*

Challenges. Academic Press, 1994.

[160] P Sabella. A rendering algorithm for visualizing 3D scalar fields. *Computer Graphics*, 22(4):51–58, 1988.

[161] M Sach, G Winkler, V Glauche, J Liepert, B Heimbach, MA Koch, C Blüchel, and C Weiller. Diffusion tensor MRI of early upper motor neuron involvement in amyotrophic lateral sclerosis. *Brain*, 127:340–350, 2004.

[162] A Sanderson, M Kirby, and CR Johnson. Display of vector fields using a reaction diffusion model. In *Proceedings of IEEE Visualization 2004*, page (to appear), October 2004.

[163] HH Schild. *MRI Made Easy*. Berlex Laboratories, Wayne, New Jersey, 1994.

[164] W Schroeder, K Martin, and B Lorensen. *The Visualization Toolkit: An Object Oriented Approach to Graphics*. Kitware, Inc., Clifton Park, New York, 2003.

[165] DF Scollan, A Holmes, R Winslow, and J Forder. Histological validation of myocardial microstructure obtained from diffusion tensor magnetic resonance imaging. *American Journal of Physiology*, 275:2308–2318, 1998.

[166] Y Seo, H Shinar, Y Morita, and G Navon. Anisotropic and restricted diffusion of water in the sciatic nerve: A (2)H double-quantum-filtered NMR study. *Magnetic Resonance in Medicine*, 42:461–466, 1999.

[167] CD Shaw, DS Ebert, JM Kukla, A Zwa, I Soboroff, and DA Roberts. Data visualization using automatic, perceptually-motivated shapes. In *Proceedings of Visual Data Exploration and Analysis*. SPIE, 1998.

[168] CD Shaw, JA Hall, C Blahut, DS Ebert, and DA Roberts. Using shape to visualize multivariate data. In *Proceedings of the 1999 Workshop on New Paradigms in Information Visualization and Manipulation*, pages 17–20. ACM Press, 1999.

[169] H-W Shen, C Johnson, and K-L Ma. Visualizing vector fields using line integral convolution and dye advection. In *Proceedings of IEEE 1996 Symposium on Volume Visualization*, pages 63–70, San Francisco, California, 1996.

[170] P Shirley and R Morley. *Realistic Ray Tracing*. AK Peters, Wellesley, Massachusetts, 2003.

[171] A Sigfridsson, T Ebbers, Heiberg, and L Wigström. Tensor field visualization using adaptive filtering of noise fields combined with glyph rendering. In *Proceedings of IEEE Visualization 2002*, Boston, Massachusetts, 2002.

[172] S Skare, T-Q Li, B Nordell, and M Ingvar. Noise considerations in the determination of diffusion tensor anisotropy. *Magnetic Resonance Imaging*, 18:659–669, 2000.

[173] CH Sotak. The role of diffusion tensor imagging in the evaluation of ischemic brain injury - a review. *NMR in Biomedicine*, 15:561–569, 2002.

[174] D Stalling and H-C Hege. Fast and resolution independent line integralconvolution.

In *Proceedings of ACM SIGGRAPH 95*, pages 249–256. Addison Wesley, 1995.

[175] D Stalling, M Zöckler, and H-C Hege. Fast display of illuminated field lines. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):118–128, 1997.

[176] E Stejskal and J Tanner. Spin diffusion measurements: Spin echos in the presence of time-dependent field gradient. *Journal of Chemical Physics*, 42:288–292, 1965.

[177] G Strang. *Linear Algebra and Its Applications*. Academic Press, Inc., Orlando, Florida, 1976.

[178] D. D. Streeter. Gross morphology and fiber geometry of the heart. In R. M. Berne, editor, *Handbook of Physiology, Vol 1: The Heart, Sect. 2: The Cardiovascular System*, pages 61–112. Williams & Wilkinson, Baltimore, MD, 1979.

[179] IF Talos, L O'Donnell, C-F Westin, SK Warfield, WM Wells, SS Yoo, L Panych, A Golby, H Mamata, SE Maier, P Ratiu, CG Guttmann, PMcL Black, FA Jolesz, and R Kikinis. Diffusion tensor and functional MRI fusion with anatomical MRI for image guided neurosurgery. In *Sixth International MICCAI*, 2003.

[180] T Theußl, H Hauser, and Eduard Gröller. Mastering windows: Improving reconstruction. In *Proceedings of IEEE Symposium on Volume Visualization 2000*, pages 101–108, October 2000.

[181] DS Tuch, TG Reese, MR Wiegell, N Makris, JW Belliveau, and VJ Wedeen. High angular resolution diffusion imaging reveals intravoxel white matter fiber heterogeneity. *Magnetic Resonance in Medicine*, 48:577–582, 2002.

[182] DS Tuch, RM Weisskoff, JW Belliveau, and VJ Wedeen. High angular resolution diffusion imaging of the human brain. In *Proceedings of the 7th Annual Meeting of ISMRM*, page 321, 1999.

[183] AM Turing. The chemical basis of morphogenesis. *Philosophical Transactions of Royal Society of London*, 237(B):37–72, 1952.

[184] G Turk. Generating textures on arbitrary surfaces using reaction-diffusion. In *Proceedings of ACM SIGGRAPH 1991*, volume 25, pages 289–298. Addison Wesley, 1991.

[185] G Turk and D Banks. Image-guided streamline placement. In *Proceedings of ACM SIGGRAPH 96*, volume 30, pages 453–460. Addison Wesley, 1996.

[186] AM Uluğ, N Beauchamp, RN Bryan, and PCM van Zijl. Absolute quantitation of diffusion constants in human stroke. *Stroke*, 28(3):483–490, 1997.

[187] AM Uluğ and PCM van Zijl. Orientation-independent diffusion imaging without tensor diagonalization: Anisotropy definitions based on physical attributes of the diffusion ellipsoid. *Journal of Magnetic Resonance Imaging*, 9:804–813, 1999.

[188] JJ van Wijk. Spot noise: Texture synthesis for data visualization. In *Proceedings of ACM SIGGRAPH 1991*, volume 25, pages 309–318. Addison Wesley, 1991.

[189] VJ Wedeen, TG Reese, VJ Napadow, and RJ Gilbert. Demonstration of primary and secondary muscle fiber architecture of the bovine tongue by diffusion tensor magnetic resonance imaging. *Biophysical Journal*, 80:1024–1028, Feb 2001.

[190] D Weinstein, G Kindlmann, and E Lundberg. Tensorlines: Advection-diffusion based propagation through diffusion tensor fields. In *Proceedings of IEEE Visualization '99*, pages 249–253, 1999.

[191] D.M. Weinstein. Stream bundles - cohesive advection through flow fields. Technical Report UUCS-99-005, University of Utah, Department of Computer Science, Salt Lake City, UT, 1999.

[192] EW Weisstein. *CRC Concise Encyclopedia of Mathematics*, pages 362–365, 1652. CRC Press, Florida, 1999.

[193] A Wenger, D Keefe, S Zhang, and DH Laidlaw. Interactive volume rendering of thin thread structures within multivalued scientific datasets. *Transactions on Visualization and Computer Graphics*, page (to appear), 2004.

[194] C-F Westin, SE Maier, B Khidhir, P Everett, FA Jolesz, and R Kikinis. Image processing for diffusion tensor magnetic resonance imaging. In *Proceedings of the 2nd Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 441–452, September 1999.

[195] C-F Westin, SE Maier, H Mamata, A Nabavi, FA Jolesz, and R Kikinis. Processing and visualization for diffusion tensor MRI. *Medical Image Analysis*, 6:93–108, 2003.

[196] C-F Westin, S Peled, H Gubjartsson, R Kikinis, and FA Jolesz. Geometrical diffusion measures for MRI from tensor basis analysis. In *Proceedings of the 5th Annual Meeting of ISMRM*, 1997.

[197] L Westover. Footprint evaluation for volume rendering. In *Proceedings of ACM SIGGRAPH 90*, pages 367–376. Addison Wesley, 1990.

[198] MR Wiegell, HBW Larsson, and VJ Wedeen. Fiber crossing in human brain depicted with diffusion tensor MR imaging. *Radiology*, 217(3):897–903, Dec 2000.

[199] A Witkin and M Kass. Reaction-diffusion textures. In *Proceedings of ACM SIGGRAPH 1991*, volume 25, pages 299–308, 1991.

[200] R Xue, PCM van Zijl, BJ Crain, M Solaiyappan, and S Mori. In vivo three-dimensional reconstruction of rat brain axonal projections by diffusion tensor imaging. *Magnetic Resonance in Medicine*, 42:1123–1127, 1999.

[201] S Zhang, ME Bastin, DH Laidlaw, S Sinha, PA Armitage, and TS Deisboeck. Visualization and analysis of white matter structural asymmetry in diffusion tensor MRI data. *Magnetic Resonance in Medicine*, 51:140–147, 2004.

[202] S Zhang, Ç Demiralp, D Keefe, M DaSilva, BD Greenberg, PJ Basser, C Pierpaoli, EA Chiocca, TS Deisboeck, and DH Laidlaw. An immersive virtual environment for DT-MRI volume visualization applications: A case study. In *Proceedings of*

IEEE Visualization 2001, pages 437–440, October 2001.

[203] S Zhang, C Demiralp, and DH Laidlaw. Visualizing diffusion tensor MR images using streamtubes and streamsurfaces. *IEEE Transactions on Visualization and Computer Graphics*, 9(4):454–462, October 2003.

[204] S Zhang and DH Laidlaw. Ellucidating neural structure in diffusion tensor MRI volumes using streamtubes and streamsurfaces. In *Proceedings of the 9th Annual Meeting of ISMRM*, April 2001.

[205] X Zheng and A Pang. Hyperlic. In *Proceedings of IEEE Visualization 2003*, pages 249–256, Seattle, Washington, 2003.

[206] L Zhukov and A Barr. Oriented tensor reconstruction: Tracing neural pathways from diffusion tensor MRI. In *Proceedings of IEEE Visualization 2002*, pages 387–394, 2002.

[207] L Zhukov and AH Barr. Heart-muscle fiber reconstruction from diffusion tensor MRI. In *Proceedings of IEEE Visualization 2003*, pages 597–602, October 2003.

[208] L Zhukov, K Museth, DE Breen, RT Whitaker, and AH Barr. Level set segmentation and modeling of DT-MRI human brain data. *Journal of Electronic Imaging*, 12(1):125–133, January 2003.

[209] M Zöckler, D Stalling, and H-C Hege. Interactive visualization of 3D-vector fields using illuminated streamlines. In *Proceedings of IEEE Visualization '96*, pages 107–113, 1996.