

Tractography on HARDI Data

Diplomarbeit

zur Erlangung des Grades eines Diplom-Informatikers
im Studiengang Informatik

vorgelegt von
Viktor Seib

Erstgutachter: Prof. Dr.-Ing. Stefan Müller
(Institut für Computervisualistik, AG Computergraphik)
Zweitgutachterin: Dipl. Inform. Diana Röttger
(Institut für Computervisualistik, AG Computergraphik)

Koblenz, im Dezember 2010

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ja Nein

Mit der Einstellung der Arbeit in die Bibliothek bin ich einverstanden.

Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.

.....
(Ort, Datum)

.....
(Unterschrift)

Contents

Contents	i
List of Figures	v
List of Tables	ix
List of Algorithms	xi
Abstract	xiii
Kurzfassung	xv
I Introduction	1
1 Introduction	3
1.1 Motivation	3
1.2 Objective	4
1.3 Organization of this Thesis	4
II Theoretical Foundations	7
2 Background	9
2.1 Magnetic Resonance Imaging	9
2.1.1 Signal Acquisition	10
2.1.2 Image Reconstruction	13
2.2 Diffusion Weighted Imaging	15
2.2.1 Diffusion	16
2.2.2 Image Acquisition	17
3 Diffusion MRI	21
3.1 Diffusion Tensor Imaging	21
3.1.1 The Diffusion Tensor	22
3.1.2 Estimation of the Diffusion Tensor	23

CONTENTS

3.1.3	Limitations of the Tensor Model	24
3.2	High Angular Resolution Diffusion Imaging	26
3.2.1	Model-based approaches	26
3.2.2	Model free approaches	28
3.2.3	Analytical Q-ball Imaging	33
3.3	Summary of Methods	37
3.4	Visualization	38
3.4.1	Scalar Indices	39
3.4.2	Glyphs	42
3.4.3	Orientation	43
3.4.4	Volume Rendering	43
4	Tractography	47
4.1	Local Deterministic Tractography	49
4.1.1	Streamline Tracing	49
4.1.2	Tensorline Tracing	54
4.1.3	Deterministic HARDI Tractography	55
4.1.4	Summary	61
4.2	Local Probabilistic Tractography	61
4.2.1	Representing Uncertainty	62
4.2.2	Propagation of Uncertainty	66
4.2.3	Example Algorithms	68
4.2.4	Summary	73
4.3	Global Tractography	73
4.3.1	Front Evolution Approaches	74
4.3.2	Energy Minimization	75
4.3.3	Example Algorithms	75
4.4	Summary	80
III	Practical Applications	83
5	Implementation	85
5.1	Development Environment	85
5.1.1	MeVisLab	86
5.2	Implementation Overview	87
5.2.1	ML Modules	87
5.2.2	Open Inventor Modules	89
5.2.3	Implemented Network	90
5.3	Detailed Module Description	94
5.3.1	HARDI_dODF_Reconstruction	94
5.3.2	HARDI_AnisotropyCriteria	98
5.3.3	MFC_Macro	103
5.3.4	HARDI_deterministic_Tractography	105

CONTENTS

5.3.5	So_dODF_Visualization	122
5.3.6	So_Fiber_Visualization	127
6	Evaluation	131
6.1	HARDI_dODF_Reconstruction	131
6.2	HARDI_Anisotropy_Criteria	134
6.2.1	Computation Times	134
6.2.2	Classification on the Fiber Cup Phantom	135
6.2.3	Classification on Human Brain Data	142
6.3	MFC_Macro	144
6.4	HARDI_deterministic_Tractography	146
6.4.1	Fiber Tracing Durations	146
6.4.2	Tractography Results on Phantom Data	146
6.4.3	Tractography Results on Human Brain Data	160
6.5	Total Computation Time	162
IV	Conclusion	163
7	Summary	165
7.1	Theoretical Part	165
7.2	Implementation and Results	167
8	Outlook	171
8.1	Fiber Tracing in Challenging Regions	171
8.2	Clustering of Fiber Populations	172
8.3	Visualization	172
A	Module Documentation	173
	Bibliography	185

List of Figures

2.1	A proton and its spin axis	10
2.2	Precession and excitation of the spin axis	11
2.3	Spin echo	12
2.4	Different MR image contrasts	13
2.5	MR image encoding gradients	14
2.6	EPI sequence	15
2.7	Isotropic and anisotropic diffusion	16
2.8	Effect of diffusion weighting on stationary spins	18
2.9	Effect of diffusion weighting on moving spins	18
2.10	DWI sequence	19
2.11	Effect of changing the diffusion weighting axis	20
3.1	Diffusion tensor ellipsoids	22
3.2	Fiber configurations	25
3.3	Multi-tensor model	26
3.4	Diffusion information from a multi-tensor approach	27
3.5	Grid and spherical sampling schemes	29
3.6	Steps of the Q-ball reconstruction	31
3.7	dODFs from QBI and PAS-MRI	32
3.8	Spherical deconvolution	32
3.9	Spherical deconvolution fODFs	33
3.10	SH basis functions	35
3.11	Modified SH basis functions	36
3.12	ADCs and MD	39
3.13	Comparison of MD, FA and Westin metrics	41
3.14	Different glyph shapes for DTI visualization	42
3.15	Orientation color coded slices	43
3.16	Orientation color coded HARDI glyphs	44
3.17	Volume-rendered FA isosurfaces	44
3.18	Volume-rendering with colors	45
4.1	Streamline and interpolation	50
4.2	FACT	52

LIST OF FIGURES

4.3	Multi ROI streamlines	53
4.4	Fiber branchings with DTI	53
4.5	TEND	54
4.6	Comparison of DTI and DSI tractography	58
4.7	QBI solid angle	60
4.8	MFACT	61
4.9	Cones of uncertainty	63
4.10	Estimation of uncertainty	65
4.11	Propagation of uncertainty	67
4.12	Comparison of bootstrap and residual bootstrap	69
4.13	Principles of symmetry and independence	71
4.14	Front evolution	75
4.15	Front evolution connectivity maps	76
4.16	Comparison of front evolution results	77
4.17	Spin energy model	78
4.18	Spin method results on phantom data	78
4.19	Spin method results on brain data	79
4.20	Modification of a connection	80
5.1	MeVisLab module types	86
5.2	MeVisLab example network	87
5.3	MeVisLab ml::Module class	89
5.4	Open Inventor SoShape class	90
5.5	MeVisLab network HARDI_Tractography.mlab	91
5.6	Dataflow in the main network	92
5.7	Class diagram of the <i>HARDI_dODF_Reconstruction</i> module	94
5.8	GUI-panel of the <i>HARDI_dODF_Reconstruction</i> module	95
5.9	Glyphs with different numbers of ODF sampling direction	96
5.10	Glyphs with different weights of isotropic diffusion	96
5.11	Control flow in the <i>HARDI_dODF_Reconstruction</i> module	97
5.12	Class diagram of the <i>HARDI_AnisotropyCriteria</i> module	99
5.13	GUI-panel of the <i>HARDI_AnisotropyCriteria</i> module	100
5.14	Control flow in the <i>HARDI_AnisotropyCriteria</i> module	102
5.15	Inner network of the <i>MFC_Macro</i> module	103
5.16	Dataflow in the inner network of the <i>MFC_Macro</i> module	104
5.17	Class diagram <i>HARDI_deterministic_Tractography</i> module	107
5.18	Output image structure of <i>HARDI_deterministic_Tractography</i>	108
5.19	GUI-panel of the <i>HARDI_deterministic_Tractography</i> module	108
5.20	Control flow in the <i>HARDI_deterministic_Tractography</i> module	109
5.21	Class diagram of the <i>So_dODF_Visualization</i> module	122
5.22	GUI-panel of the <i>So_dODF_Visualization</i> module	124
5.23	ODF visualization as glyph and as lines	124
5.24	Visualization examples from <i>So_dODF_Visualization</i>	125
5.25	Control flow in the <i>So_dODF_Visualization</i> module	127

LIST OF FIGURES

5.26	Class diagram of the <i>So_Fiber_Visualization</i> module	128
5.27	GUI-panel of the <i>So_Fiber_Visualization</i> module	128
5.28	Visualization of fiber pathways	129
5.29	Control flow in the <i>So_Fiber_Visualization</i> module	129
6.1	Sampling points and SH coefficients	132
6.2	dODF reconstruction durations	133
6.3	dODF reconstruction durations below one minute	133
6.4	Anisotropy criteria precalculation durations	135
6.5	B_0 -image of the Fiber Cup phantom	136
6.6	Classification results of the <i>GFA</i> criterion	137
6.7	Classification results of the <i>stdDev</i> criterion	138
6.8	Classification results of the <i>Vesna</i> criterion	139
6.9	Classification results of the <i>Chen</i> criterion	140
6.10	Classification results of the <i>FMI</i> criterion	141
6.11	Classification results of the <i>MDL</i> criterion	141
6.12	Classification results of the <i>stdDev</i> criterion (human brain)	143
6.13	Classification results of the <i>Chen</i> criterion (human brain)	143
6.14	Classification results of the <i>FMI</i> criterion (human brain)	143
6.15	<i>MFC</i> results on phantom data	145
6.16	<i>MFC</i> results on human brain data	145
6.17	Fiber Cup seedpoints and ground truth	147
6.18	Tracing results for <i>Euler PDD</i> and <i>RK4 PDD</i>	148
6.19	<i>Euler PDD</i> evaluation metrics	150
6.20	<i>RK4 PDD</i> evaluation metrics	151
6.21	<i>Weighted Dirs Angle</i> evaluation metrics	152
6.22	Tracing results of <i>Weighted Dirs Angle</i> and <i>Weighted Dirs MFC</i>	153
6.23	<i>Weighted Dirs MFC</i> evaluation metrics	154
6.24	Tracing results of the <i>Distance Based</i> algorithm	155
6.25	<i>Distance Based</i> evaluation metrics	156
6.26	Fiber Cup results first four places	157
6.27	Fiber Cup l_2 metric	160
6.28	Tracing results on human brain data	161
6.29	Comparison of Computation Times	162
A.1	GUI-panel of the <i>HARDI_dODF_Reconstruction</i> module	173
A.2	GUI-panel of the <i>HARDI_AnisotropyCriteria</i> module	175
A.3	GUI-panel of the <i>HARDI_deterministic_Tractography</i> module	178
A.4	Output image structure of <i>HARDI_deterministic_Tractography</i>	179
A.5	GUI-panel of the <i>So_dODF_Visualization</i> module	180
A.6	GUI-panel of the <i>So_Fiber_Visualization</i> module	182

List of Tables

3.1	Comparison of diffusion MRI modalities	38
3.2	Comparison of HARDI methods	39
5.1	Tesselation orders and sampling directions	95
5.2	Colors for voxel classification	123
6.1	dODF reconstruction durations	132
6.2	MDL calculation durations	135
6.3	Fiber Tracing Durations	146
6.4	Fiber Cup <i>l2</i> metric	158
6.5	Fiber Cup <i>tan</i> metric	159
6.6	Fiber Cup <i>curv</i> metric	159
6.7	Fictitious Fiber Cup Results	160
A.1	Tesselation orders and sampling directions	174
A.2	Colors for voxel classification	181

List of Algorithms

1	FMI criterion classification	101
2	Chen's criterion classification	101
3	Trace Fiber Algorithm	110
4	Euler/RK4 PDD Fiber Tracing	112
5	Weighted Dirs Angle Fiber Tracing	114
6	Weighted Dirs MFC Fiber Tracing	116
7	Distance Based Fiber Tracing	117
8	findPossibleDirections	119
9	findOptimalDistance	120
10	findLongDistances	121
11	FMI criterion classification	174
12	Chen's criterion classification	176

Abstract

Diffusion weighted imaging is an important modality in clinical imaging and the only possibility to gain insight into the human brain noninvasively and in-vivo. The applications of this imaging technique are diversified. It is used to study the brain, its structure, development and the functionality of the different areas. Further, important fields of application are neurosurgical planning, examinations of pathologies, investigation of Alzheimer's, strokes, and multiple sclerosis.

This thesis gives a brief introduction to MRI and diffusion MRI. Based on this, the mostly used data representation in diffusion MRI in clinical imaging, the diffusion tensor, is introduced. As the diffusion tensor suffers from severe limitations new techniques subsumed under the term *HARDI* (high angular resolution diffusion imaging) are introduced and discussed in detail. Further, an extensive introduction to tractography, approaches that aim at reconstructing neuronal fibers, is given.

Based on the knowledge from the theoretical part established tractography algorithms are redesigned to handle *HARDI* data and, thus, improve the reconstruction of neuronal fibers. Among these algorithms, a novel approach is presented that successfully reconstructs fibers on phantom data as well as on human brain data. Further, a novel global classification approach is presented to cluster voxels according to their diffusion properties.

Kurzfassung

Diffusionsgewichtete Bildgebung ist eine wichtige Modalität in der klinischen Praxis. Sie stellt gegenwärtig die einzige Möglichkeit dar, nicht invasiv und in vivo Einblicke in das menschliche Gehirn zu erhalten. Die Einsatzgebiete dieser Technik sind sehr vielseitig. Sie wird zur Untersuchung des Gehirns, seiner Struktur, seiner Entwicklung und der Funktionsweisen seiner verschiedenen Areale eingesetzt. Weiterhin spielt diese Modalität eine wichtige Rolle bei der Operationsplanung am Gehirn und der Untersuchung von Schlaganfall, Alzheimer und Multipler Sklerose.

Diese Arbeit gibt eine kurze Einführung in die Bildgebung mittels MRT und geht auf die Entstehung diffusionsgewichteter Bilder ein. Darauf aufbauend wird der Diffusionstensor, die am meisten verbreitete Datenrepräsentation in der Diffusionsbildgebung, vorgestellt. Da die Repräsentation der Diffusion als Diffusionstensor erhebliche Einschränkungen darstellt, werden neue Methoden zur Datenrepräsentation vorgestellt und diskutiert. Diese neuen Methoden werden unter dem Begriff *HARDI* (Diffusionsbildgebung mit hoher Winkelauflösung, von engl. high angular resolution diffusion imaging) zusammengefasst. Weiterhin wird eine ausführliche Einführung in das Thema der Traktografie, der Rekonstruktion von Nervenbahnen im Gehirn, gegeben.

Basierend auf diesem theoretischen Wissen werden etablierte Algorithmen der Traktografie von Diffusionstensor- auf *HARDI*-Daten überführt. Dadurch wird die Rekonstruktion der Nervenbahnen entscheidend verbessert. Es wird eine vollständig neue Methode vorgestellt, die in der Lage ist, Nervenbahnen sowohl auf einem Phantomdatensatz, als auch auf einem vom Menschen stammenden Gehirndatensatz zu rekonstruieren. Weiterhin wird ein neuartiger globaler Ansatz vorgestellt, um Voxel anhand ihrer Diffusionseigenschaften zu klassifizieren.

Part I

Introduction

Chapter 1

Introduction

This chapter provides an introduction to this thesis. The motivation and objectives of this thesis are presented in the first two sections. The third section gives an overview of the organization of the thesis.

1.1 Motivation

Based on magnetic resonance imaging (MRI) a new modality, diffusion weighted imaging (DWI), was developed in the last decades. This modality measures noninvasively the diffusion of water molecules in the human body. Most commonly it is used to analyze the brain. Anisotropic diffusion can be found in the nerve fibers (the white matter) of the brain. The nerve fibers of the brain restrict diffusion and, thus, allow to infer directional information of these structures. By means of DWI this structural information can be displayed in vivo.

The reconstruction of these fibers provides insight into brain structure and allows to analyze its development. Furthermore, it allows to identify pathological changes by their altered diffusion behavior. Application areas of DWI are for example planning of surgical interventions for brain tumors and examinations of Alzheimer's, strokes, and multiple sclerosis.

The diffusion rate is often described by a second order tensor, a 3×3 matrix (Diffusion-Tensor-Imaging - DTI). By extracting the eigenvectors and eigenvalues of the tensor the direction and degree of diffusion can be estimated in each voxel. Several approaches have been developed to reconstruct white matter pathways based on this directional information. These reconstruction approaches are called fiber tracing or tractography.

1.2 Objective

One major disadvantage of DTI is its inability to delineate more than one diffusion direction per voxel. Complicated structures like fiber crossings or branchings can not be represented in this manner as no reliable directional information can be extracted from DTI in such regions. These limitations encouraged the development of high angular resolution diffusion imaging (HARDI). HARDI allows to calculate an orientation distribution function (ODF) of possible diffusion directions for each voxel. Thus, several direction per voxel can be identified.

With the development of this new technique the established fiber tracing approaches need to be adjusted to the new data. More than one direction per voxel has to be considered. Further, other stopping criteria or connectivity properties between voxels might be beneficial. The purpose of this thesis is to evaluate state-of-the-art tractography algorithms. Possible extensions regarding HARDI data will be identified and fiber tracing algorithms adapted to these data. The development environment of Fraunhofer MeVis (MeVisLab) shall be used as a framework for implementation.

The individual tasks of this thesis are:

1. Familiarization with the MeVisLab framework and the current tractography approaches
2. Evaluation of current fiber tracing approaches and identification of possible extensions regarding HARDI data
3. Implementation of selected tractography algorithms based on the Q-ball reconstruction technique
4. Outlook to fiber tracing in challenging regions, possibilities for clustering and visualization
5. Documentation and evaluation of the results

1.3 Organization of this Thesis

This thesis is organized as follows: Part II provides theoretical knowledge that is important for the understanding of this thesis. Chapter 2 gives an overview on MRI and diffusion MRI. Further, the term diffusion is introduced and explained. Chapter 3 explains the DTI and HARDI techniques. Several reconstruction methods for HARDI data are introduced. The Q-ball technique is treated in more detail as it forms the basis of the implementation for this thesis. In the last section of chapter 3 visualization methods are presented. Chapter 4 focuses on tractography. State-of-the-art tractography algorithms are introduced for DTI and HARDI data. An overview is given

1.3. ORGANIZATION OF THIS THESIS

on deterministic, probabilistic and global methods. Various examples are presented and discussed for each category of approaches.

The knowledge gained in the theoretical part is applied in part III in practice. Chapter 5 explains the modules that were implemented for the MeVisLab framework. The designed algorithms are presented in detail and their individual extensions to HARDI data are discussed. Chapter 6 evaluates the implementation. The evaluation is performed on phantom data as well as on a human brain dataset.

Part IV concludes this thesis. A summary is given in chapter 7 and the results are discussed. Finally, chapter 8 gives an outlook to fiber tracing in challenging regions and possibilities for fiber clustering and visualization.

Part II

Theoretical Foundations

Chapter 2

Background

This chapter provides a brief introduction to methods and processes that form the basis of this thesis. Fundamental terms such as *magnetic resonance imaging* (section 2.1), *diffusion*, and *diffusion weighted imaging* (section 2.2) will be introduced and will be referred to throughout further chapters. A basic understanding of the signal formation in the used modalities and properties of the imaged tissue will be provided. An extensive treatment of these terms would go beyond the scope of this thesis. However, this background knowledge is sufficient to understand the drawbacks and benefits of the different methods described in chapters 3 and 4. References to further reading will be given in each section.

2.1 Magnetic Resonance Imaging

Magnetic resonance imaging (MRI) is a widely used modality for mapping human tissue *in vivo*. There are several kinds of MRI-techniques, each serving a different purpose. The medical application of MRI started in the 1980s. Descoteaux provides a brief history of the development of MRI in the introduction of his PhD thesis [Des07]. Good introductions to MRI can be found in [Rob00], [Fin07], and [JBB09]. A detailed technical and mathematical description of MRI signal acquisition is given in [MA95].

In contrast to other imaging techniques (e.g. computed tomography (CT)), MRI is noninvasive and does not use ionizing radiation. It is based on strong magnetic fields, usually in the range of 1.5 to 3.0 T, and radio waves. Where CT images provide highest contrasts in rather hard tissue (e.g. bones), MR images give better insight into soft tissue, like brain or muscle. The contrast between anatomic structures in MR images can be influenced in a number of ways by varying the intensity of the magnetic fields or the radio waves.

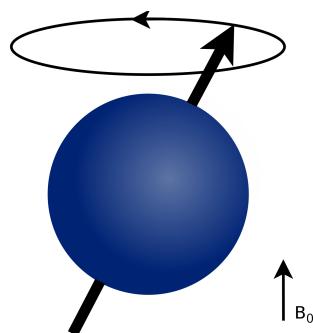


Figure 2.1: A proton and its spin axis (thick arrow). The spin axis rotates about the direction of the magnetic field, B_0 . Image adopted from [Fin07, p. 62].

2.1.1 Signal Acquisition

Atomic nuclei possess a characteristic property called *spin*. The spin can be compared to angular momentum to facilitate its understanding. To continue this comparison, a “rotational” axis can be assigned to the spins. Charged particles spinning about an axis produce a magnetic momentum. The spinning axis is thus more precisely termed *axis of magnetic momentum*, but often the simpler term *spin direction* is used instead.

Hydrogen nuclei (protons) have a very strong magnetic momentum. Thus, soft tissue (i.e. tissue with high water portions) provides highest signal strength for MR imaging. Normally, the spins of the protons are oriented randomly. In this case the magnetic momenta cancel each other out, resulting in no net magnetic momentum. However, when brought into a strong magnetic field, the spins’ axes align to the direction of this field and start precessing about its axis (i.e. the spins’ directions rotate periodically about the direction of the magnetic field, Figure 2.1). The direction of the magnetic field is commonly called the z -direction. The precessional frequency, also called *Larmor frequency*, f_l , is given by the Larmor equation

$$f_l = \gamma B_0 \quad (2.1)$$

Here, B_0 is the applied magnetic field and γ , the gyromagnetic ratio, a characteristic property of subatomic particles. Compared to other nuclei, protons have a very high γ of 42.57 MHz/T, resulting in a Larmor frequency of 127.7 MHz at typical B_0 -fields of 3 T in clinical MRI. This is also the resonance frequency that can be applied to alter the directions of the spins (see below).

After all spins are aligned and precessing about the z -axis, no further interactions take place. However, without rotating magnetic fields (i.e. with a fixed spin axis) no electric current in measuring coils is induced and hence no signal from the MR scanner can be measured.

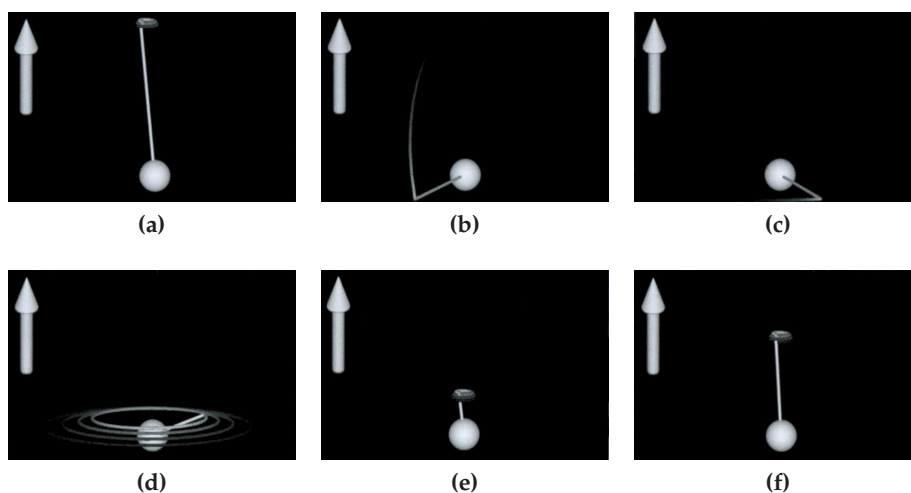


Figure 2.2: The spin's axis is precessing about the orientation of B_0 , the main magnetic field (a). The excitation pulse rotates the spinning axis by 90° into the xy -plane (b). The spin's axis still precesses about the orientation of B_0 but in the xy -plane (c). The precessional component in this plane decays exponentially with a time constant T_2 (d). The parallel component exponentially recovers with a slower time constant T_1 (e, f). Images adopted from [JBB09, p. 13].

To produce a measurable signal, a radio frequency-pulse (rf-pulse) is applied perpendicular to the z -axis. Although not as intense as the B_0 -field, the pulse is strong enough to rotate the spins (to be more precise, the net magnetic momentum), when applied at the Larmor frequency. The duration of the pulse determines the rotating angle, here it is 90° . This pulse is also called *excitation pulse*. After the excitation pulse is turned off, three things happen (Figure 2.2):

1. The net magnetic momentum is precessing in the xy -plane about the z -axis, thus inducing current in measuring coils placed in that plane. This signal is measured and used to reconstruct the MR image.
2. The component of the magnetic momentum that lies in the xy -plane decays exponentially (and so does the measured current) with a characteristic time constant T_2 . This is the T_2 -relaxation or spin-spin-relaxation and is typically on the order of 100 milliseconds.
3. The component of the magnetic momentum parallel to the main magnetic field exponentially recovers to its former direction (z -axis) with a time constant T_1 . This process is called T_1 -relaxation or spin-lattice-relaxation and induces current that is measured by coils along the z -axis. T_1 is typically on the order of 1 second.

2.1. MAGNETIC RESONANCE IMAGING

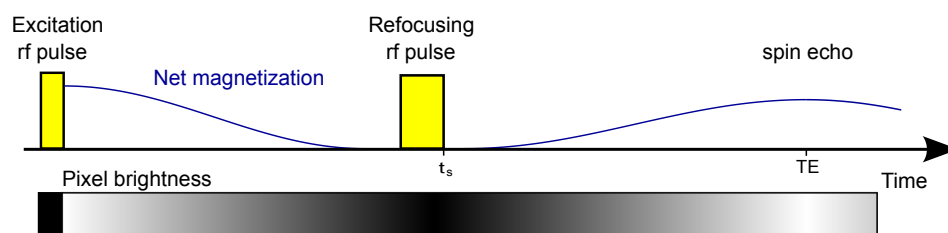


Figure 2.3: After the excitation pulse spins lose phase coherence leading to loss of net magnetization. A refocusing pulse is applied at time t_s , the spins start to regain their phase coherence. At time $2t_s$ (echo time, TE) all spins are phase coherent again. This effect is called spin echo. Image adopted from [JBB09, p. 13].

While the spins precess in the xy -plane they lose phase coherence. This is due to small inhomogeneities in the B_0 -field and different kinds of tissue, both leading to slightly different Larmor frequencies. The signal loss due to T_2 -relaxation and dephasing together is called T_2^* -relaxation. It leads to a much faster signal decay than the T_2 -relaxation alone. To reverse the signal loss from dephasing most imaging sequences use a second rf-pulse (the so called *refocusing pulse*), rotating the net magnetic momentum by 180° . After the second rf-pulse the spins regain phase coherence after a time period t_s (Figure 2.3). This time period is exactly the same, as the time that separates the two rf-pulses. The regained phase coherence leads to a signal maximum and the signal is measured for image reconstruction (note that due to the T_2 -relaxation this signal is slightly weaker than the signal directly after the excitation-pulse). This effect is called *spin-echo* and was first described by Hahn [Hah50]. The time $2t_s$ is called the echo time, TE .

It may not be straightforward that the spins regain phase coherence. A demonstrative comparison to illustrate the spin echo effect is given in [Fin07]: Imagine a sprint, where every runner starts at the same position (phase coherence after the excitation-pulse) and runs with a constant but different speed than the others (different Larmor frequencies). After a certain time t_s they all are at different positions (loss of phase coherence). At this point they reverse their directions (refocusing pulse), maintaining their respective speeds. After another time period t_s all runners arrive at the same position (spin echo).

Refocusing pulses are applied several times at intervals of TE and the characteristic time constant T_2 is measured at several spatial positions in the tissue. As the signal strength decays an excitation pulse needs to be applied after a certain time. This time is called the repetition time, TR . TE and TR are two of many parameters that can be chosen when designing imaging sequences. They are altered to achieve different contrasts in the resulting images.

Three types of images are often referred to in the literature: proton den-

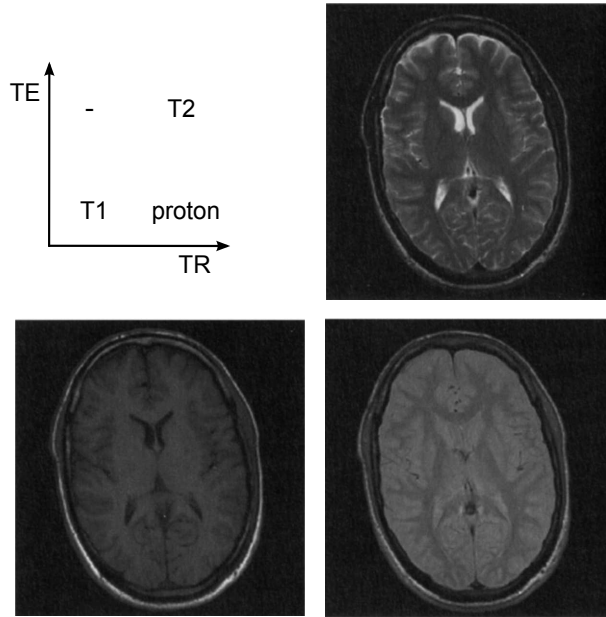


Figure 2.4: Different MR image contrasts that result from varying TE and TR : T_1 -weighted image (bottom left), T_2 -weighted image (top right), and proton density image (bottom right). The diagram in the top left corner illustrates the lengths of TE and TR needed to achieve the respective contrasts (e.g. short TE and short TR for a T_1 -weighted image). Image adapted from [Fin07, p. 66]

sity, T_1 -weighted, and T_2 -weighted images (Figure 2.4). Proton density images result from very short TE and long TR , minimizing the effects of both types of relaxation. T_1 -weighted images result from short TE and TR in the range of T_1 , while T_2 -weighted images need TE in the range of T_2 and long TR . The signal in a pixel at position x, y can be calculated as

$$S_{x,y} = M_0 \cdot (1 - \exp^{-TR/T_1}) \cdot \exp^{-TE/T_2} \quad (2.2)$$

where M_0 is the spin or proton density. More detailed descriptions of imaging characteristics are given in [Rob00] and [MA95].

2.1.2 Image Reconstruction

Once the MRI signal is measured an image needs to be reconstructed from this signal. Medical images are often acquired in *slices*. A slice is a two-dimensional image depicting a cross section through the sample tissue. Adjacent slices show adjacent regions in the tissue. All slices taken together accumulate the three-dimensional information of the sample. A whole MRI scan produces tens or hundreds of slices depending on the size of the region in question and the thickness of each slice.

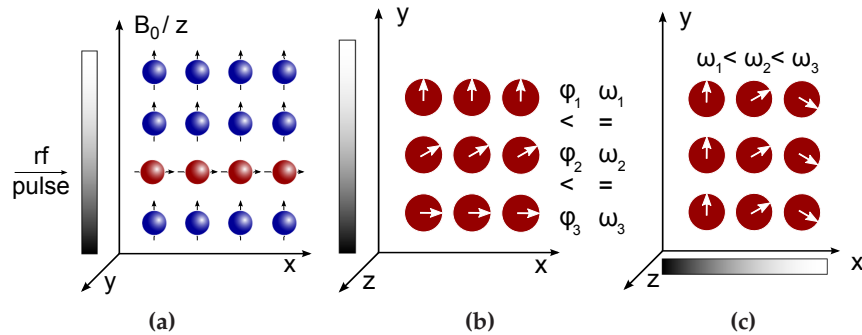


Figure 2.5: The slice selection gradient is applied along the z -axis resulting in different Larmor frequencies along this axis. The excitation pulse affects only spins with the same frequency as the pulse (red spins in (a)). Figures (b) and (c) show the excited slice. After the application of the phase encoding gradient along the y -axis all spins precess at the same frequency, ω , but with different phases ϕ (b). The effect of the frequency encoding gradient applied along the x -axis is shown in (c) (the different phases from the previous step are not shown).

To encode an image out of the MRI signal three linear magnetic field gradients are used. Typically, these are applied along the three main axes. The first gradient is used for slice selection. In this example, the direction of the *slice selecting gradient* is the same as that of the B_0 -field (z -axis). The main magnetic field, B_0 , is overlaid with a non-homogeneous field, the slice selecting gradient. Thus, the Larmor frequency of the spins along that gradient changes with the gradient, since it depends on the magnetic field. However, inside an arbitrary xy -plane along this gradient the frequency is the same. A simultaneously applied 90° excitation rf-pulse only effects one single slice, namely the slice with the same Larmor frequency as the rf-pulse. Slice selection is accomplished by varying the frequency of the rf-pulse. The thickness of a slice depends on the strength of the gradient and the frequency bandwidth of the rf-pulse. The thickness lies usually in the range of 1 to 5 mm. The application of the first gradient is illustrated in Figure 2.5a.

Within a slice the spatial position of the sampled MRI signal needs to be encoded. Therefore, a second gradient is applied along an axis perpendicular to the first one (the x -axis in this example). This is the so called *frequency encoding gradient*. It changes the precessional frequency, ω , of the spins along the x -axis within the excited plane. Thus, different frequencies along the x -axis of the resulting image are sampled (Figure 2.5c). The resolution within the slice lies in the range of 0.5 to 1 mm.

A third gradient, the *phase encoding gradient* (also line selection gradient), is applied in between the two aforementioned gradients along the re-

2.2. DIFFUSION WEIGHTED IMAGING

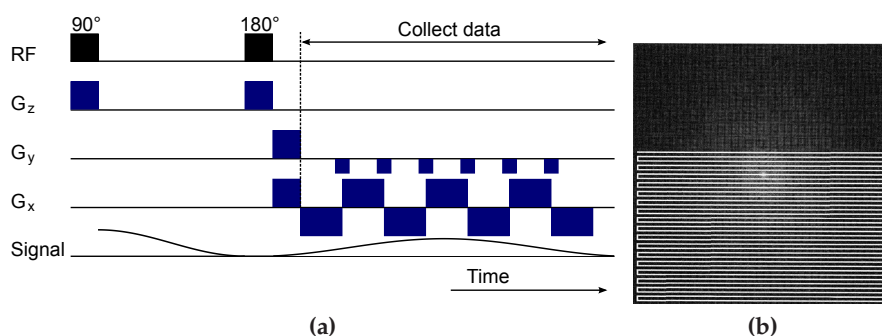


Figure 2.6: A simplified diagram of the EPI sequence (a). Data for several lines is collected after each refocusing pulse. The trajectory of the data acquisition in Fourier space is shown as a white line in (b). Image modified and adopted from [JBB09, p. 17].

maintaining axis (y in this example). This gradient is turned on for a short time only. While the gradient is switched on the spins precess at different frequencies along the y -axis. However, when it is turned off all spins precess with the same frequency but retain the respective phase, ϕ , that they acquired during the gradient application (Figure 2.5b). The resulting MRI signal is a sum of superimposed phases. The phase encoding gradient (and thus the subsequent frequency encoding gradient) needs to be applied several times, each time with a different strength, leading to a change in the phase difference. The change in the phase difference over all measurements taken together corresponds to a frequency change along the y -axis.

The resulting image contains the frequency information of the measured signals along the x - and y -axis (Figure 2.6b). Applying an inverse 2D Fourier transform decodes the desired spatial information on the tissue within the analyzed region. Numerous different imaging sequences have been designed to minimize the acquisition time and maximize the signal quality. The described imaging sequence is a simplified example of the *echo planar imaging* (EPI) sequence (Figure 2.6a). To obtain better results it is necessary to filter the image for noise and imaging artifacts (e.g. caused by movements and cardiac pulsation), before applying the inverse Fourier transform. Please refer to [MA95] and [Fin07] for details on removal of imaging artifacts. The acquisition time for all slices depends on the size of the sample and the desired resolution. It is typically in the range of 10 to 30 minutes.

2.2 Diffusion Weighted Imaging

So far imaging techniques that depict relaxation times in different kinds of tissue were treated. This “standard” MRI method results in relaxation

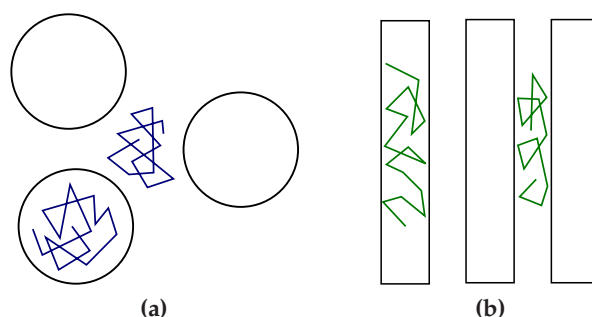


Figure 2.7: In isotropic diffusion particles move randomly in all directions (a). The movements of the particles in anisotropic diffusion are still random (b). However, barriers enforce a predominant orientation to their displacement. The *ADC* along the tubes is higher than across to them. Image modified and adopted from [Bea02, p. 2].

weighted images. In the last decade of the 20th century a new modality, diffusion weighted imaging (DWI), was developed allowing new contrasts in MR imaging. Before covering this imaging method the term *diffusion* needs to be introduced. A more extensive treatment of this term as well as of DWI can be found in [JBB09]. For a detailed mathematical derivation of the below mentioned equations please refer to [Cam04].

2.2.1 Diffusion

Diffusion, or Brownian motion, was first described by Robert Brown in 1828 when he studied pollen grains suspended in water. In 1905 Einstein explained diffusion as the random motion of molecules due to thermal energy [Ein05]. Thus, this random motion can be observed in any substance with a temperature above zero Kelvin. Diffusion is characterized by a diffusion rate, expressed as the diffusion coefficient, D , and a diffusion direction. According to Einstein the mean displacement, ΔX , along one axis of a particle follows the Gaussian distribution and is given by the equation

$$\Delta X = \sqrt{2Dt} \quad (2.3)$$

for one dimension. Here, D is the diffusion coefficient¹, and t is the diffusion time.

This equation applies to free diffusion (i.e. the particles move freely without being restricted by barriers). If free diffusion is possible in every direction, the diffusion is called isotropic (Figure 2.7a) and the equation can be extended to two or three dimensions. The opposite of isotropic diffusion is anisotropic diffusion. This term is used when diffusion is restricted in one or more directions (Figure 2.7b).

¹The diffusion coefficient D for water at body temperature (37°C) is $3 \cdot 10^{-3} \text{ mm}^2/\text{s}$.

Since in the case of anisotropic diffusion D appears smaller than in isotropic diffusion, a new term, the *apparent diffusion coefficient (ADC)*, was introduced to describe the measured rate of anisotropic diffusion.

In the human body measurements of anisotropic diffusion can be used to infer structural information of tissue. Anisotropic diffusion is present in muscles or in white matter of the brain. White matter consists of densely packed nerve fibers (*axons*) that connect different functional areas. The typical width of an axon is in the range of 0.1 to 10 μm whereas its length can reach several tens of centimeters. Thus, in measuring the *ADC* parallel and perpendicular to axons one can infer their orientation and position in the brain. The parallel *ADC* appears free, whereas the perpendicular *ADC* is restricted to a very short distance (Figure 2.7b). Another important component of the brain is gray matter. The diffusion in gray matter is isotropic, hence, this type of tissue can be distinguished from white matter.

Further, there are measurable differences in *ADC* of healthy and pathological brain tissue (e.g. brain tumors). Also examinations of the effects of Alzheimer's, strokes and multiple sclerosis in DW images can provide new insights and understandings of these diseases. Relation of diffusion to pathologies and brain structure is extensively treated in [JBB09] and [Bea02].

2.2.2 Image Acquisition

Diffusion weighted MRI is the only possibility to measure diffusion of water molecules noninvasively in vivo. Hahn was the first recognizing the effect of molecular diffusion to the spin echo signal [Hah50]. Carr and Purcell proposed a method to measure diffusion [CP54]. Stejskal and Tanner introduced many improvements and laid the foundation for modern diffusion measurements [ST65].

Stejskal and Tanner proposed the *pulsed gradient spin echo* method to add diffusion weighting to a MRI sequence. The basic idea is to apply a magnetic field gradient after the 90° excitation rf-pulse. This gradient adds a certain phase to each spin depending on the position along the direction of the gradient. Then a second gradient with same strength but reversed polarity is applied in the same direction. This time the added phase is negative due to the reversed polarity of the gradient. The net added phase on stationary spins is zero (Figure 2.8). However, if a particle and, thus, its spin moves between the two gradients the absolute values of the added phases will differ. Thus, the spins are not re-phased after the second gradient leading to signal attenuation (Figure 2.9). The greater the spatial displacement of a spin during the diffusion time, the greater the resulting signal attenuation. Nowadays, most imaging sequences use a 180° refocusing rf-pulse. Since this pulse flips the phases of the spins the diffusion weighting gradients are applied on either side of the refocusing pulse with same polarity

2.2. DIFFUSION WEIGHTED IMAGING

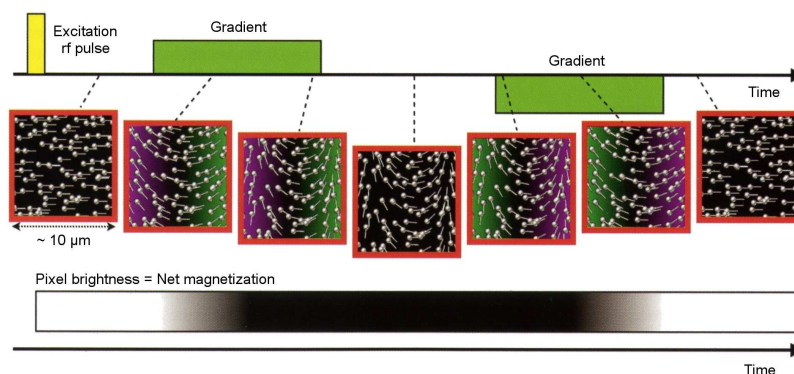


Figure 2.8: The effect of diffusion weighting gradients on stationary spins. During the application of the first diffusion weighting gradient spins precess at different frequencies (lower on the left side, shown in magenta, higher on the right side, shown in green). The second gradient has reversed polarity, thus, resulting in an equal but reversed phase change. Since the spins do not move between the two gradients, there is no net effect on the net magnetization and, thus, pixel brightness. Image adopted from [JBB09, p. 19].

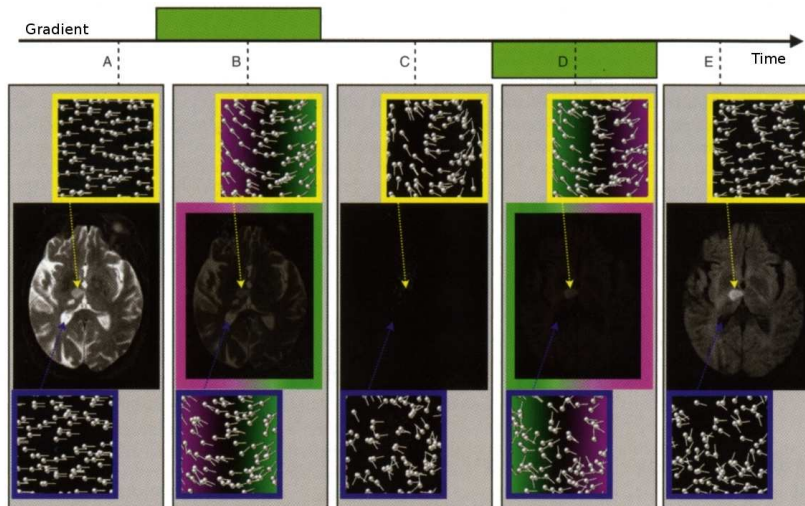


Figure 2.9: The effect of diffusion weighting gradients on moving spins. Due to the first diffusion weighting gradient spins lose their phase coherence. The second gradient recovers the magnetization incompletely because of the spin displacement between the two gradients. Regions with high diffusion have greater signal loss and appear darker (blue box) than regions with low diffusion (yellow box, area of stroke). Image adopted from [JBB09, p. 20].

2.2. DIFFUSION WEIGHTED IMAGING

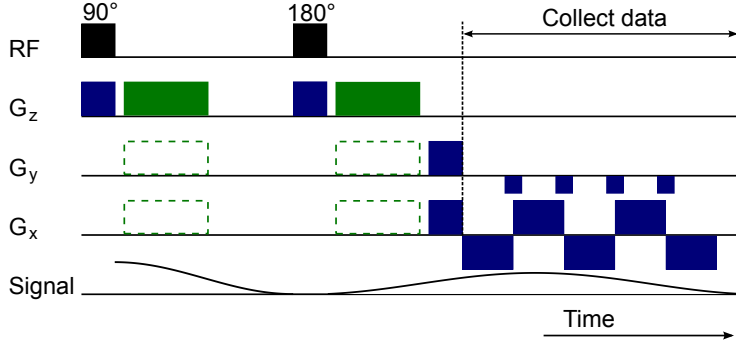


Figure 2.10: A simplified diagram of a diffusion weighted imaging sequence. Diffusion gradients are applied on either side of the refocusing pulse (shown in green) along one axis. The dotted boxes indicate that diffusion weighting can be applied along an arbitrary axis. Image modified and adopted from [JBB09, p. 24].

(Figure 2.10). The signal from each pixel in the slice can now be calculated as

$$S_{x,y} = M_0(1 - \exp^{-TR/T_1}) \exp^{-TE/T_2} \exp^{-b} ADC \quad (2.4)$$

Note that this is equation 2.2 with the added factor $\exp^{-b} ADC$, accounting for the additional signal attenuation caused by diffusion. It consists of the b -value and the (apparent) diffusion coefficient ADC along the direction of the applied gradients.

Due to the fact that the diffusion time, t , and T_2 lie on a similar time scale diffusion weighting disturbs the T_2 weighted image. Therefore, an additional T_2 image without diffusion weighting is always acquired and is referred to as the B_0 -image. The diffusion coefficient in equation 2.4 is estimated by

$$\log \left(\frac{S(b)}{S(b=0)} \right) = -bADC \quad (2.5)$$

for every applied gradient direction. Here, $S(b=0)$ is the signal without diffusion weighting and $S(b)$ the diffusion weighted signal. It is obvious that the resulting ADC depends on the chosen direction (e.g. parallel or perpendicular to nerve fibers). Therefore, usually several gradients are applied to probe the diffusion along different directions, typically the three main axes (Figure 2.11). Only if the ADC is the same in all directions, the diffusion is considered to be isotropic.

The b -value is widely used to characterize the level of induced sensitivity to diffusion. For gradients with a constant strength b is given by

$$b = (\gamma G \delta)^2 (\Delta - \delta/3) \quad (2.6)$$

Here, G denotes the gradient strength, Δ the time between the starts of the two gradients and δ the duration of each gradient. The time $\Delta - \delta/3$ is

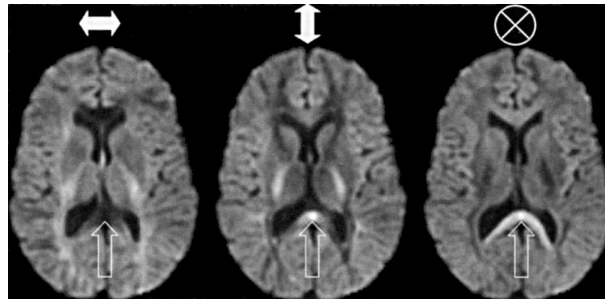


Figure 2.11: Effect of applying the diffusion weighting gradients along different axes (indicated by white arrows above the images). Dark areas have high ADCs, whereas light areas have low ADCs. The unfilled arrows point at the splenium of the corpus callosum. In the leftmost image this area appears dark, indicating the dominant left-right orientation of nerve fibers in this region. Image adopted from [JBB09, p. 41].

referred to as the diffusion time. Δ is typically in the range of some tens of milliseconds, whereas δ is chosen to be small (few milliseconds) in order not to affect spins while they diffuse over a larger length scale. Higher b-values lead to a higher sensitivity to diffusion but also to a lower signal-to-noise ratio (SNR). Analyses show that the optimal b-value for diffusion weighted imaging is in the range of 900 to 1200 s/mm² [AB05].

Chapter 3

Diffusion MRI

Based on diffusion weighted imaging that was introduced in the last chapter more sophisticated methods to measure and characterize diffusion were developed in the past years. The first of these developed methods was *diffusion tensor imaging* (DTI) treated in section 3.1. Although it has severe limitations it is still the most commonly used diffusion MRI technique in medicine. To address the limitations of DTI acquisition schemes and signal reconstruction algorithms were extended and improved. These new methods are subsumed under the term *high angular resolution diffusion imaging* (HARDI) and are discussed in section 3.2. A summary of these methods is given in section 3.3, while section 3.4 presents commonly used visualization methods in DTI and HARDI.

The purpose of this chapter is to introduce DTI and the most common HARDI reconstruction methods. Their respective advantages and disadvantages will be pointed out and discussed. Further, popular means to visualize the reconstructed data from these approaches will be introduced at the end of this chapter. References to relevant literature will be given in each section.

3.1 Diffusion Tensor Imaging

As was mentioned in the last chapter the ADC is highly dependent on the direction of the gradient and is therefore rotationally variant. Thus, the ADC alone can not adequately delineate the diffusion properties in anisotropic structures. With this in mind Basser *et al.* developed a rotationally invariant model, the diffusion tensor [BML94b], [BML94a]. This model will be treated in this section. For more information on the diffusion tensor model please refer to [JBB09], [Des07] and [Cam04].

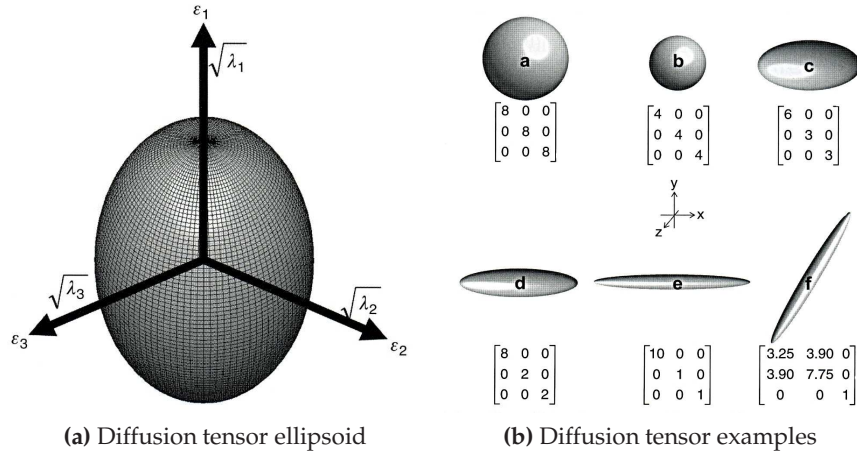


Figure 3.1: Examples of diffusion tensor visualization as an ellipsoid. Images adopted from [JBB09, p. 43].

3.1.1 The Diffusion Tensor

Instead of using a single scalar, Basser *et al.* introduced a second order tensor, a 3×3 matrix, to describe diffusion in 3D:

$$\mathbf{D} = \begin{bmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{xy} & D_{yy} & D_{yz} \\ D_{xz} & D_{yz} & D_{zz} \end{bmatrix} \quad (3.1)$$

The diagonal elements of this matrix correspond to the diffusivity along the three main axes (i.e. the *ADC* along these axes). The other elements describe the correlation between displacements along two particular axes (e.g. D_{xy} along the x - and y -axes) and should not be confused with diffusion along hypothetical new axes. Taken all elements together the diffusion tensor can be seen as a 3D covariance matrix of spin displacements. Note that the off-diagonal elements can become zero or negative, whereas the *ADC* along all axes is always positive and greater than zero. This model is based on the assumption that diffusion follows a Gaussian distribution.

If all off-diagonal elements become zero, there is no correlation of displacement among the axes. In such a case the *principal diffusion direction* (PDD) aligns with one of the main axes of reference. Further, in this special case the diagonal elements correspond to the eigenvalues of the matrix (λ_1 , λ_2 , and λ_3). The eigenvector that belongs to the biggest eigenvalue indicates the orientation of the principal diffusion direction and thus the assumed underlying fiber orientation. In the case that at least one off-diagonal element is not zero the eigenvalues have to be computed instead of taking the diagonal elements of the tensor.

The eigenvalues can be thought of as the *ADC* along the directions of the eigenvectors. The diffusion tensor can be illustrated as an ellipsoid.

The axes of the ellipsoid are given by the eigenvectors of the tensor (Figure 3.1a). Since we assume Gaussian diffusion the ellipsoid's axes have to be scaled according to equation 2.3 with the square root of the corresponding eigenvalue. The resulting ellipsoidal shape wraps the volume that a particle can diffuse to with equal probability when starting at its center. Examples of different tensor ellipsoids and the corresponding diffusion tensors are given in Figure 3.1b.

3.1.2 Estimation of the Diffusion Tensor

To estimate the diffusion tensor six values have to be found. In analogy to equation 2.5 where the scalar b-value was used to determine the scalar *ADC*, the b-value is replaced by a b-matrix, \mathbf{b} , of the form

$$\mathbf{b} = \begin{bmatrix} b_{xx} & b_{xy} & b_{xz} \\ b_{xy} & b_{yy} & b_{yz} \\ b_{xz} & b_{yz} & b_{zz} \end{bmatrix} \quad (3.2)$$

reformulating equation 2.5 to

$$\log \left(\frac{S(\mathbf{b})}{S(\mathbf{b}=0)} \right) = - \sum_{i=1}^3 \sum_{j=1}^3 b_{ij} D_{ij} \quad (3.3)$$

Here, b_{ij} is an element of the b-matrix and D_{ij} an element of the diffusion tensor. The computation of the b-matrix is beyond the scope of this chapter, please refer to [MLB97] for details.

Since equation 3.3 contains six unknown elements at least six diffusion weighted images acquired with different diffusion weighting directions are needed to solve it (plus one image without diffusion weighting). A conventional way of solving equation 3.3 is constructing a matrix, \mathbf{B} , with the number of rows, N , equal to the number of diffusion weighted images and six columns. The columns contain the elements of the b-matrices, \mathbf{b}^j ($j \in [1 \dots N]$), that were used for the acquisition of each diffusion weighted image:

$$\mathbf{B} = \begin{bmatrix} b_{xx}^1 & 2b_{xy}^1 & 2b_{xz}^1 & b_{yy}^1 & 2b_{yz}^1 & b_{zz}^1 \\ b_{xx}^2 & 2b_{xy}^2 & 2b_{xz}^2 & b_{yy}^2 & 2b_{yz}^2 & b_{zz}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{xx}^N & 2b_{xy}^N & 2b_{xz}^N & b_{yy}^N & 2b_{yz}^N & b_{zz}^N \end{bmatrix} \quad (3.4)$$

With \mathbf{S} , a vector of log-transformed measured signals from the N diffusion weighted images,

$$\mathbf{S} = \left[\log \left(\frac{S(\mathbf{b}^1)}{S(\mathbf{b}=0)} \right) \quad \log \left(\frac{S(\mathbf{b}^2)}{S(\mathbf{b}=0)} \right) \quad \dots \quad \log \left(\frac{S(\mathbf{b}^N)}{S(\mathbf{b}=0)} \right) \right]^T \quad (3.5)$$

and \mathbf{D}_v a vector containing the elements of the diffusion tensor \mathbf{D} ,

$$\mathbf{D}_v = [D_{xx} \ D_{xy} \ D_{xz} \ D_{yy} \ D_{yz} \ D_{zz}]^T \quad (3.6)$$

equation 3.3 is rewritten as

$$\mathbf{S} = \mathbf{B}\mathbf{D}_v \quad (3.7)$$

In the simple case where exactly six diffusion weighted images are acquired, \mathbf{B} is square and equation 3.7 is solved by taking the inverse of \mathbf{B} :

$$\mathbf{D}_v = \mathbf{B}^{-1}\mathbf{S} \quad (3.8)$$

However, acquiring more than six images helps reducing noise significantly. Studies have shown that optimal imaging sequences for DTI involve at least 30 gradient directions [Jon04], [BAH⁺03]. As a result, \mathbf{B} no longer is square and its pseudo-inverse needs to be taken in order to solve equation 3.7:

$$\mathbf{D}_v = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{S} \quad (3.9)$$

This *ordinary least squares* method includes the assumption that the variance of noise is equal for each diffusion weighted image. However, this is not generally true. In their original publications on the diffusion tensor model Basser *et al.* use a more complicated method, the *weighted linear least squares* approach, that fits the data more exactly [BML94b], [BML94a].

3.1.3 Limitations of the Tensor Model

The diffusion tensor is calculated based on diffusion weighted measurements along different directions. The assumption this model is based on is that diffusion follows a Gaussian distribution. As the 3D Gaussian distribution function forms an ellipsoid the tensor is visualized in different variations of this shape. Depending on the underlying diffusion it takes a cigar-shaped, pancake-shaped, spherical or any intermediate form (Figure 3.1). However, the assumption of Gaussian diffusion is only true for isotropic diffusion. In a highly organized area like white matter in the brain diffusion is hindered in many directions. Hence, the Gaussian model can only be an approximation that does not work well in some cases.

As imaging voxels not only cover nerve fibers but brain tissue in general each diffusion tensor also contains isotropic diffusion information from gray matter and diffusion between nerve fibers. This is called *partial volume effect* which is a common problem in diffusion MRI. Further, the typical voxel sizes in DTI range from 1 to 5 mm. In contrast to this diameters of axonal fibers vary from 0.1 to 10 μm . Thus, a single tensor contains diffusional information of hundreds of fibers. Since the diffusion is assumed

3.2. HIGH ANGULAR RESOLUTION DIFFUSION IMAGING

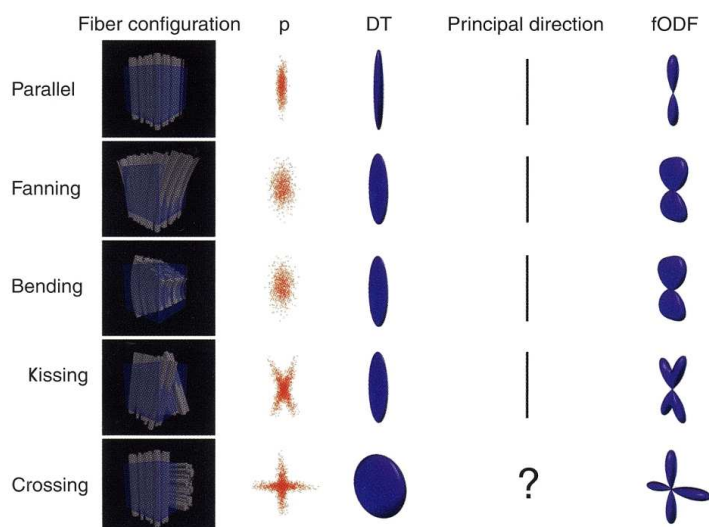


Figure 3.2: Different fiber configurations, their diffusion distributions, p , and the corresponding shapes of the diffusion ellipsoids are shown. The fourth column shows the directional information that can be extracted from the diffusion tensor. With HARDI a more detailed delineation of diffusion, the fODF, can be obtained. The fODF is shown in the rightmost column. Image adopted from [JBB09, p. 57].

to be Gaussian only one fiber orientation per voxel can be resolved. As a result the tensor's orientation is an average of all fiber directions belonging to the corresponding voxel. If all fibers in that voxel are aligned parallel to one another, as is the case in large fiber structures like the corpus callosum, the tensor depicts their orientation very well. However, many complex structures with heterogeneous fiber orientations exist in the white matter of the brain [Bea02]. The diffusion tensor is insufficient when fibers bend, fan, kiss or cross. As can be seen in Figure 3.2, bending and fanning fibers result in the same tensor ellipsoid. Although these two fiber structures are not distinguishable from the tensor, the main eigenvector provides a reasonable approximation of the underlying fiber orientations. In the case of crossing fibers, however, no information about the fiber orientation can be inferred from the tensor. The tensor's shape results in a pancake-shape or even a sphere like in the case of isotropic diffusion.

To be able to resolve multiple fiber orientation per voxel new models and acquisition schemes were developed. These are discussed in the next section.

3.2 High Angular Resolution Diffusion Imaging

In high angular resolution diffusion imaging (HARDI) usually 60 and more different diffusion weighted images are acquired. Additionally, higher b -values allow for resolving diffusion with a higher angular accuracy than in DTI. First attempts to characterize diffusion and determine the ADC profile with HARDI data have shown improvements in regions where DTI data did not provide sufficient information [ABA02], [Fra02].

Therefore, more techniques were developed to estimate the orientation of diffusion from HARDI data. This techniques can be divided into two groups: model-based and model free approaches. The most important methods of these two groups are briefly introduced in this section. One of these methods, Q-ball imaging (QBI), is described in more detail as it forms the basis for the implementation for this thesis and is used in later chapters for fiber tracing. Further information on HARDI can be found in [JBB09] and [Des07].

3.2.1 Model-based approaches

This section introduces state-of-the-art methods that resolve multiple fiber orientation per voxel by a priori assuming a certain structure of the measured signal. The corresponding model representing this structure is then fitted to the data and the diffusion profile is reconstructed.

Multi-tensor Model

A logical extension of the diffusion tensor model is modeling more than one tensor for each voxel. The multi-tensor model was introduced by Tuch *et al.* [TRW⁺02]. The Gaussian diffusion distribution is replaced by a mixture of Gaussians. This model assumes that each voxel contains n distinct

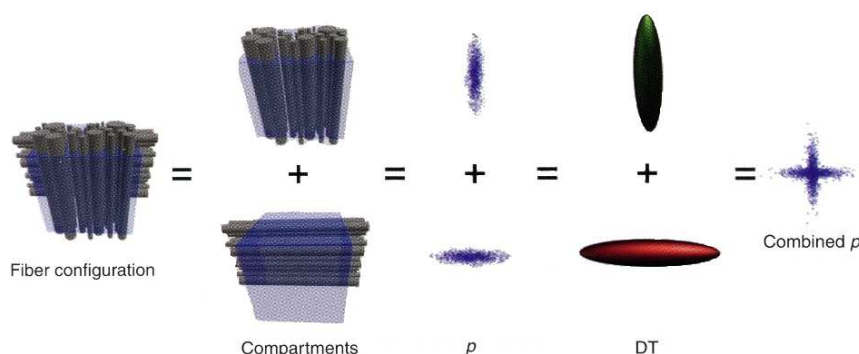


Figure 3.3: Two tensors approach modeling crossing fibers. Image adopted from [JBB09, p. 59].

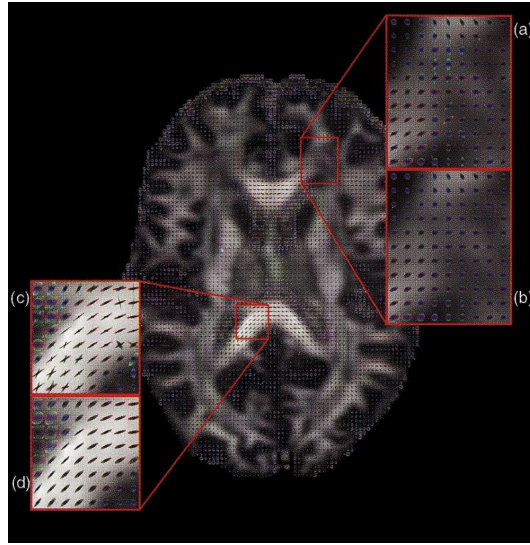


Figure 3.4: Diffusion signal modeled with two tensors per voxel (insets **(a)** and **(c)**). The same regions are shown with the modeling of only one tensor (insets **(b)** and **(d)**). Image adopted from [JBB09, p. 60].

fiber populations. A second assumption is that water molecules diffuse only within one of the fiber populations. Each of the fiber populations is described by a separate diffusion tensor. An example showing the modeling of a fiber crossing with two tensors is given in Figure 3.3. The overall diffusion probability distribution, p , is a sum of all Gaussians, weighted according to their respective volume fraction in the voxel:

$$p(\mathbf{x}) = \sum_{i=1}^n a_i G(\mathbf{x}; \mathbf{D}_i, t) \quad (3.10)$$

Here, $G(\cdot; \mathbf{D}, t)$ is a Gaussian distribution with zero mean and covariance $2\mathbf{D}t$ (similar to equation 2.3), x is the displacement and t the diffusion time. The weighting $a_i \in [0, 1]$ is the volume fraction of the i th fiber population (all a_i sum to 1).

This approach relies on the fact that the number of fiber populations, n , is known a priori. A tensor is then fitted using non-linear optimization into each of the fiber populations. In practice often a maximum n of 2 is used due to the limited number of measurements and imaging noise. With $n = 1$ this model is the same as DTI.

The multi-tensor model is able to resolve fiber crossings of two fiber populations in known anatomical structures. However, with a $n = 2$ two tensors are fitted in every voxel of the measured data. This leads to errors and instabilities in regions where only one fiber population is present (inset (c) in Figure 3.4). One cause of the instabilities is the large number of

parameters that have to be estimated for every voxel. For example, 13 parameters need to be determined for $n = 2$ (6 for each voxel plus one for the volume fraction). Apart from the method proposed by Tuch *et al.*, other methods to stabilize the multi-tensor fitting [CGZ⁺04b] and improve the a priori estimation of n [BJB⁺07a] have been proposed. Nevertheless, the correct selection of the model parameter n remains a problem that should not be underestimated in its meaning for the correctness of the multi-fiber reconstruction.

Variations

A simple variation of the multi-tensor approach is the explicit modeling of different diffusion compartments within one voxel [BWJ⁺03]. This model is also called *ball and stick* model. It assumes that the diffusing particles belong either to a restricted (stick) or to a free (ball) fiber population. The diffusion tensor to model the stick includes only one non-zero eigenvalue to clearly identify the fiber direction. By modeling various sticks multiple fiber populations can be resolved. Other variations replace the stick by a stick distribution to resolve more complex structures. The slightly more complex *composite and hindered restricted model of diffusion* (CHARMED) proposed in [AFRB04] uses a mixture of restricted fiber populations and is thus able to recover multiple fiber orientations per voxel.

3.2.2 Model free approaches

This section introduces model free state-of-the-art approaches to reconstruct the diffusion profile. The model free approaches attempt to reconstruct the fiber distribution within a voxel based only on the measured diffusion signal rather than modeling a discrete number of fiber populations. For each voxel a probability density function (PDF) is generated. This function is commonly called *orientation distribution function* (ODF) as it estimates the orientation of the fibers within a voxel. The ODF is a radial projection of the diffusion signal on a sphere, often approximated with the Funk-Radon transform explained below in this section.

Typically, two types of ODFs are distinguished in literature. Since the measurements provide the diffusion distribution, the reconstructed ODF is actually a diffusion ODF (dODF) and not the fiber ODF (fODF) of the true underlying fibers. It is important to understand the difference between these two functions. As diffusion not only takes place parallel to nerve fibers but also perpendicular to them and between them the dODF can be blurred and ambiguous. Increasing the b-value leads to sharper dODFs, thus, better approximating the fODF. However, this can only be done at the cost of lower SNR. The dODF is used in most cases as there is no way to

find the true fODF. Techniques to better approximate the fODF are a current field of research.

Diffusion Spectrum Imaging

Diffusion spectrum imaging (DSI) (also called *q-space imaging*, QSI) aims at measuring the diffusion distribution, p , directly without making any assumptions on the shape of p or the underlying fiber structure. This is accomplished by making use of the *q-space*. Q-space theory and its mathematical properties are treated in great detail in [Cal94]. For better understanding of the methods described below q-space can be thought of as the Fourier space.

The measured signal, S , is sampled on a grid in q-space. The grid is defined by the 3D wavevector $\mathbf{q} = \gamma\delta\mathbf{G}$ where γ is the gyromagnetic ratio, δ the gradient duration and \mathbf{G} the diffusion gradient vector. The vector \mathbf{q} is related to the b-value as $b = |\mathbf{q}|^2 t$ where t is the diffusion time (please compare to equation 2.6). By adjusting the gradient direction and the b-value different positions on the grid are selected.

The q-space is related to the Fourier transform of p . Thus, by taking the inverse 3D Fourier transform of S , samples of p on a grid of spatial displacements are obtained (Figure 3.5a). Further, p is projected onto a sphere to get the dODF. The resulting value for every orientation is the sum of all samples of p that lie on a line through the origin with this orientation.

The reconstructed dODF is very accurate, however, several limitations exist. DSI relies on infinitesimally short gradient pulses. In practice, these pulses are on a length scale close to the diffusion time. As a result, the reconstructed dODF is blurred. A second limitation is the necessity to acquire a big number of measurements on a wide range of b-values to cover the whole 3D grid. Typically, between 500 and 1000 measurements are acquired leading to a significant increase in acquisition time (up to 60 minutes). In the original publication measurements with 500 different values of \mathbf{q} were taken [WRT⁺00]. The b-value varied from 500 to 20000. However, with high b-values the SNR reduces as the measured parameter is signal attenuation.

Q-ball Imaging

The idea behind Q-ball imaging (QBI) [Tuc04] is to approximate the dODF that would result from DSI by means of a spherical acquisition scheme (Figure 3.5b). Rather than sampling the whole 3D grid only points on a fixed radius (with a fixed b-value) are sampled. This approximation allows for a significantly lower acquisition time. However, the resulting dODF is blurred and has a lower angular resolution that can also lead to imprecise

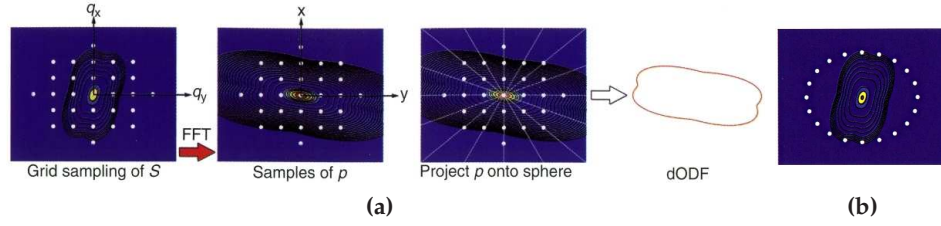


Figure 3.5: Image (a) shows the signal reconstruction of DSI. The signal, S , is sampled on a grid in Fourier space and transformed to Cartesian space. The diffusion distribution, p , is projected onto a sphere and the dODF is reconstructed. In QBI a spherical sampling scheme is used to sample the Fourier space (image (b)). Images adopted from [JBB09, p. 62].

peaks. With higher b-values the angular resolution is improved at the cost of reducing the SNR. Here, an appropriate balance needs to be found.

The subsequent steps of QBI are illustrated in Figure 3.6. After sampling the diffusion attenuation signal the discrete samples are interpolated on the sphere. To obtain the dODF, Ψ , the Funk-Radon transform (FRT) is calculated for the interpolated signal. The FRT maps one spherical function to another by taking the integral over the *great circle*, $C(\mathbf{x})$, for every desired direction, \mathbf{x} . The great circle is the plane perpendicular to \mathbf{x} through the origin. This can be expressed as

$$\Psi(\mathbf{x}) = \int_{C(\mathbf{x})} S(\mathbf{q}) d\hat{\mathbf{q}} \quad (3.11)$$

where $\hat{\mathbf{q}} = \mathbf{q}/|\mathbf{q}|$. The resulting dODF is a discrete set of points. The steps illustrated in Figure 3.6 can be combined to a single matrix multiplication. Thus, the Q-ball algorithm is computationally light.

Figure 3.7a shows the reconstructed dODFs. In regions with only one fiber population the principal direction can be seen clearly (inset (a)). In the region of fiber crossings multiple directions can be distinguished but the dODF is too smooth to form separate peaks (inset (b)). This is due to the low b-value of 1200 used in the acquisition for this example. QBI demands for b-values of 3000 and higher for better angular resolution. Tournier *et al.* used 80 gradient directions and a b-value of 4000. They were able to resolve 45 degree crossings with the Q-ball algorithm [TYC⁺08].

Later studies used *spherical harmonics* to represent the dODF, rather than radial basis functions [DAFD07]. The analytical reconstruction presented by Descoteaux *et al.* will be used in this thesis to reconstruct the dODF. It is described in more detail in section 3.2.3.

A related method, the diffusion orientation transform (DOT), was proposed by Özarlan *et al.* [OSV⁺06]. DOT calculates a single contour of the diffusion distribution, p , on a fixed radius, R_0 . This is different from the dODF which has contributions from all contours. Depending on the choice

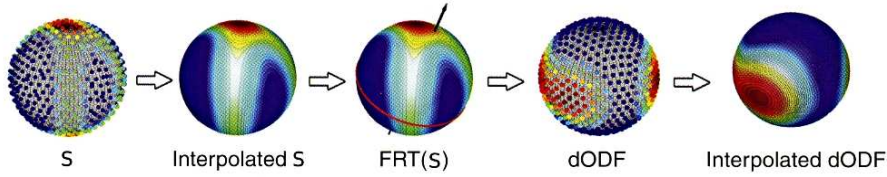


Figure 3.6: Steps of the Q-ball reconstruction. Image adopted from [JBB09, p. 63].

of R_0 and the q -vector both functions can appear similar. With the use of spherical harmonic functions DOT can be evaluated analytically and thus in a computationally efficient way.

A recent study has shown that DOT is more robust to noise than QBI [PRP⁺08]. Higher b -values and an increasing number of gradient directions significantly improve the Q-ball reconstruction, whereas the results from DOT remain unaffected. In general, DOT was able to better resolve fiber crossing at lower b -values than QBI. Further, since DOT is more robust to noise higher order SH terms than in QBI can be used to resolve fiber crossings. However, the authors of the study also pointed out than finding the optimal R_0 parameter for DOT in in-vivo data sets is challenging. Further, the calculation speed of the Q-ball approach is a significant advantage of this method compared to DOT.

Persistent Angular Structure MRI

Like Q-ball imaging persistent angular structure (PAS) MRI samples the signal in q -space at one fixed radius, r , only. The idea is to determine a function that expresses the persistent angular structure of the measured signal. In other words, the angular structure of all radii is projected onto the sphere of radius r and represents the mobility of particles in each direction. Thus, the PAS is a property of the diffusion distribution, p , rather than the underlying fODF. The resulting function has a Fourier transform that is the best approximation of the measured signal.

In the original work of Jansons and Alexander [JA03] this approximation is obtained by using a maximum entropy cost function and fitting the data with a non-linear iterative algorithm. An important property of the maximum entropy method is its capability of finding a representation (i.e. function) that contains the most information and, thus, best describes the measured signal. Since a non-linear fitting is computationally very expensive spherical harmonics can be used as a linear representation. This modification reduces the computation time significantly at the price of lower angular resolution.

PAS produces sharp ODFs (Figure 3.7b) and recovers fiber crossings at 60 degrees with 54 gradient directions, SNR of 16 and b -values in the range of 1500 to 2000 mm^{-2} [Ale05]. Further, simulations on phantom data show

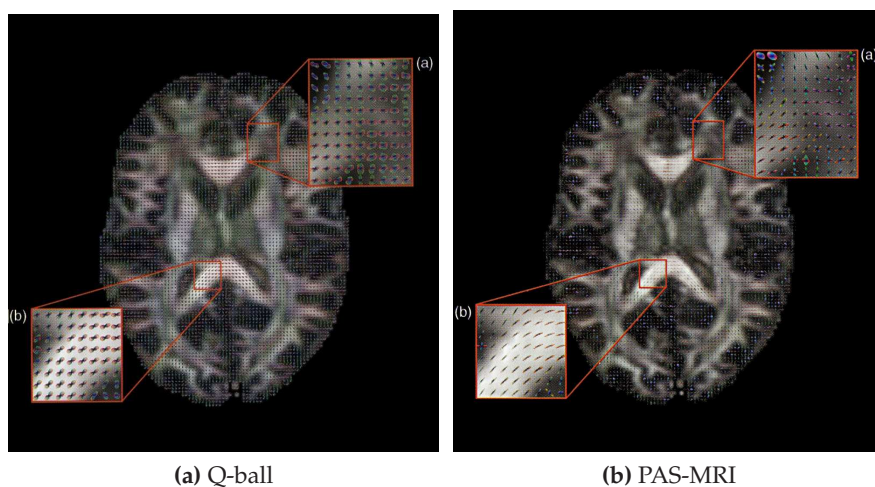


Figure 3.7: Image (a) shows the dODF resulting from Q-ball reconstruction. fODFs approximated from PAS-MRI reconstruction are shown in (b). Images adopted from [JBB09, pp. 63, 66].

that with a moderate increase in data quality the linear representation recovers fibers with a comparable accuracy to PAS MRI.

Spherical Deconvolution

Spherical deconvolution (SD) aims at recovering the fODF directly instead of using an estimate based on the dODF [TCGC04]. The basic idea is to consider the measured signal, S , as a sum of different fiber orientations within the voxel each convolved with a response function. Figure 3.8 illustrates an example of a crossing of two fibers. Each orientation of the true underlying fODF is convolved with the response function R . This results in two elements as only two non-zero orientations in the fODF exist. Their sum is the measured signal, S . For the reconstruction of the fiber orientations, S is deconvolved with the response function R . Tournier *et al.* estimate the

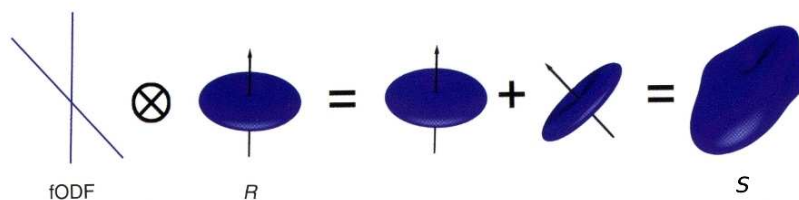


Figure 3.8: Principle of spherical deconvolution. The fODF convolved with the response function, R , is the measured signal, S . Image adopted from [JBB09, p. 64].



Figure 3.9: Output fODF of different SD algorithms. From left to right: unfiltered, low-pass filtered and super resolved cSD. Image adopted from [JBB09, p. 65].

response function by taking the average of the most anisotropic voxels in the brain data.

Figure 3.8 shows an ideal example. In practice, however, the resulting signal always contains contributions from all possible fiber directions. This is due to the movement of particles taking place in all directions even perpendicular to the main fiber orientations. Noise in the imaging process is another major source that contributes to false peaks in the reconstructed fODF. Tournier *et al.* use low-pass filtering to remove these peaks at the price of a reduced angular resolution (Figure 3.9). In a later work they introduced a regularization method (constrained SD, cSD) that allows for removing false peaks while retaining the high angular resolution [TCC07]. Further, assuming the same R for all fiber populations is another source of errors since fibers have different cell sizes, densities, and packing configurations.

With 80 gradient directions, a b-value of 1000 smm^{-2} , and SNR of 95, Tournier *et al.* were able to consistently resolve 45 degree fiber crossings on phantom data [TYC⁺08].

3.2.3 Analytical Q-ball Imaging

Of all methods introduced above only Q-ball imaging offers fast computation while not making any a-priori assumptions on the true underlying fiber structure. This is achieved by using linear basis functions for the diffusion ODF estimation, rather than spherical radial basis function as in the original publication [Tuc04]. Recently, three groups developed separately analytical reconstruction methods that use spherical harmonics (SH) as linear basis functions for QBI [And05], [HMH⁺06], and [DAFD07]. In this thesis the approach of Descoteaux *et al.* is used. As is shown in [DAFD07], this method provides some improvements over the previous approaches:

1. **Computation speed:** Descoteaux *et al.*'s method is up to 15 times faster than the original numerical QBI solution proposed by Tuch.
2. **Regularization:** In the regularization method used in [HMH⁺06] (the so called *Tikhonov regularization*) all coefficients are weighted equally. Further, the data is assumed to live on a flat manifold. Descoteaux

et al. use the *Laplace–Beltrami* regularization where the data lives on a sphere. The weighting of higher-order coefficients is reduced in order to reduce perturbations due to noise.

3. **Fiber detection:** The angular error in the detection of the diffusion ODF maxima is reduced at the cost of slightly reducing angular resolution.

This section introduces shortly the SH basis functions and Descoteaux *et al.*'s method of dODF reconstruction and regularization. For full details please refer to the original publication in [DAFD07]. A more extensive treatment of SH and their mathematical background is given in [Des07].

The Spherical Harmonics Basis

With the Fourier Transform every function can be expressed as a sum of different sines and cosines determined by their respective frequency, phase and amplitude. In analogy to the Fourier Transform one can rewrite every complex function on the sphere with a sum of spherical harmonics weighted by their respective coefficients. Spherical harmonics functions are characterized by an order l and a degree m ($l \geq 0, |m| \leq l$) and can be computed as

$$Y_l^m(\theta, \phi) = \sqrt{\frac{2l+1}{4\pi} \cdot \frac{(l-m)!}{(l+m)!}} P_l^m(\cos\theta) e^{im\phi} \quad (3.12)$$

with $\theta \in [0, \pi]$ the polar angle and $\phi \in [0, 2\pi[$ the azimuthal angle indicating a point on the sphere in polar coordinates. P_l^m is the associated Legendre polynomial of order l and degree m .

One important property of SH is the antipodal¹ symmetry for even orders l and the antipodal anti-symmetry for odd orders l . Mathematically expressed, this is

$$Y_l^m(\pi - \theta, \phi + \pi) = \begin{cases} Y_l^m(\theta, \phi), & \text{if } l \text{ even} \\ -Y_l^m(\theta, \phi), & \text{if } l \text{ odd} \end{cases} \quad (3.13)$$

The real part squared, $\text{Re}[Y_l^m(\theta, \phi)]^2$, of spherical harmonics up to order $l = 3$ is shown in Figure 3.10 to illustrate these functions.

The HARDI signal measures the signal attenuation that is caused by diffusion. This signal is real and symmetric since diffusion is not characterized by a direction but rather an orientation. To describe this signal in a physically meaningful way the spherical functions also have to feature these properties. Symmetry is achieved by only considering SH functions

¹Two points are antipodal when they are located on exactly opposite sides of the sphere.

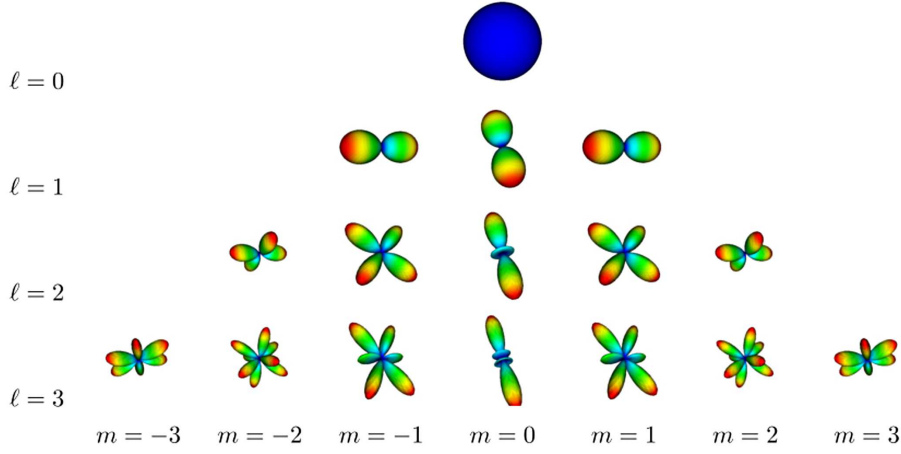


Figure 3.10: SH basis functions for orders $l = 0$ to 3 and the corresponding degrees. The colors illustrate the radius from low (blue) to high (red). Image adopted from [Des07, p. 65].

of even order l . The function becomes real-valued if the real and imaginary part of the SH is chosen appropriately according to the degree m .

With these constraints, Descoteaux *et al.* define a single index j for $l = 0, 2, 4, \dots, L$ and $m = -l, \dots, 0, \dots, l$ as $j(l, m) = (l^2 + l + 2)/2 + m$. Further, with this index they introduce a modified, real, and symmetric, SH basis as

$$Y_j = \begin{cases} \sqrt{2} \cdot \text{Re}(Y_l^{|m|}), & \text{if } m < 0 \\ Y_l^m, & \text{if } m = 0 \\ \sqrt{2} \cdot (-1)^{m+1} \text{Im}(Y_l^m), & \text{if } m > 0 \end{cases} \quad (3.14)$$

where $\text{Re}(Y_l^m)$ and $\text{Im}(Y_l^m)$ represent the real and imaginary parts of Y_l^m respectively. The normalization factor $\sqrt{2}$ imposes a third constraint, the orthonormality, on this modified basis. The modified SH basis is illustrated in Figure 3.11.

For a given order L of a SH series there are $R = 0.5 \cdot (L+1)(L+2)$ terms. With $j = 1, \dots, \infty$ every real and symmetric function on a sphere can be expressed. However, only the terms with $j = 1, \dots, R$ are considered here since higher order terms are very susceptible to noise. In image processing low-pass filtering a Fourier Transform is the analog to truncating the SH series at R terms.

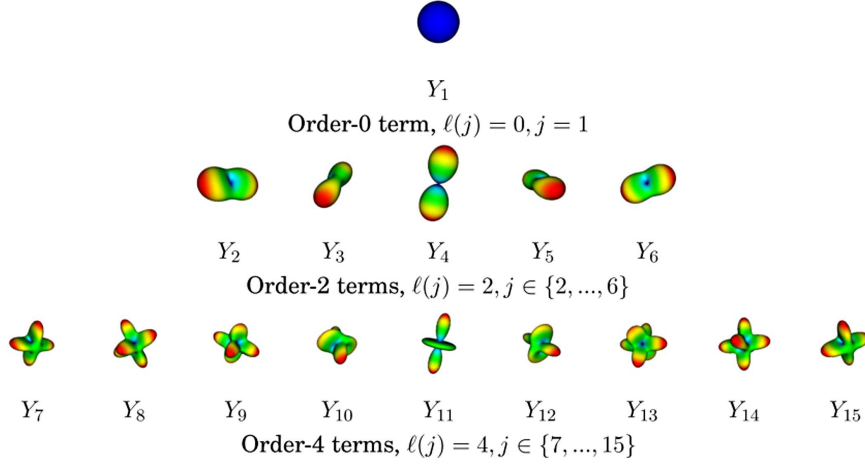


Figure 3.11: Modified SH basis functions up to order 4. Image adopted from [Des07, p. 67].

Diffusion ODF Estimation

The truncated smooth estimation of the HARDI signal for every gradient encoding direction i is formulated as

$$S(\theta_i, \phi_i) = \sum_{j=1}^R c_j Y_j(\theta_i, \phi_i) \quad (3.15)$$

where R represents the number of terms for a given SH order L , c_j are the weighting coefficients for the SH basis functions Y_j and θ_i, ϕ_i the i th gradient direction expressed in spherical coordinates.

As the signal S_i is measured it does not have to be estimated for the applied diffusion directions. Rather, the coefficients c_j that best match the measured signal have to be found. For this purpose the input signal is represented as a $N \times 1$ vector, \mathbf{S} , where N denotes the total number of gradient encoding directions. The $R \times 1$ vector, \mathbf{C} , holds the weighting coefficients. Further, a $N \times R$ matrix, \mathbf{B} , with the discrete modified SH basis is constructed:

$$\mathbf{B} = \begin{pmatrix} Y_1(\theta_1, \phi_1) & Y_2(\theta_1, \phi_1) & \dots & Y_R(\theta_1, \phi_1) \\ \vdots & \vdots & \ddots & \vdots \\ Y_1(\theta_N, \phi_N) & Y_2(\theta_N, \phi_N) & \dots & Y_R(\theta_N, \phi_N) \end{pmatrix} \quad (3.16)$$

With these definitions equation 3.15 is rewritten as a linear system

$$\mathbf{S} = \mathbf{B}\mathbf{C} \quad (3.17)$$

Since $N \gg R$ in matrix \mathbf{B} , the pseudo-inverse of \mathbf{B} needs to be taken in order to solve this linear system:

$$\mathbf{C} = (\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T \mathbf{S} \quad (3.18)$$

3.2. HIGH ANGULAR RESOLUTION DIFFUSION IMAGING

The resulting coefficients in \mathbf{C} provide the best-fitting truncated series to the measured signal from all diffusion encoding directions N . Once these coefficients are obtained the discrete equation 3.15 is rewritten as

$$S(\theta, \phi) = \sum_{j=1}^R c_j Y_j(\theta, \phi) \quad (3.19)$$

With the known coefficients this function provides an estimate of the signal S outside of the discrete measurements for any (θ, ϕ) .

Regularization and Transformation of the Diffusion Signal

To reduce ODF estimation errors when using higher SH orders Descoteaux *et al.* make use of the Laplace-Beltrami regularization. They define a diagonal regularization matrix, \mathbf{L} , as

$$\mathbf{L} = \begin{pmatrix} \ddots & & & \\ & l_j^2(l_j + 1)^2 & & \\ & & \ddots & \\ & & & \ddots \end{pmatrix} \quad (3.20)$$

where l_j is the order associated with the j th coefficient². Further they derive a diagonal matrix, \mathbf{P} , as a simplification of the Funk-Radon integral:

$$\mathbf{P} = \begin{pmatrix} \ddots & & & \\ & 2\pi P_{l_j}(0) & & \\ & & \ddots & \\ & & & \ddots \end{pmatrix} \quad (3.21)$$

Here, P_l , is the Legendre polynomial of degree l evaluated at 0:

$$P_l(0) = \begin{cases} 0, & \text{if } l \text{ odd} \\ (-1)^{l/2} \frac{1 \cdot 3 \cdot 5 \dots (l-1)}{2 \cdot 4 \cdot 6 \dots l}, & \text{if } l \text{ even} \end{cases} \quad (3.22)$$

The coefficients vector, \mathbf{C} , is obtained by a single matrix multiplication

$$\mathbf{C} = \mathbf{P}(\mathbf{B}^T \mathbf{B} + \lambda \mathbf{L})^{-1} \mathbf{B}^T \mathbf{S} \quad (3.23)$$

with the smoothing parameter λ . Descoteaux *et al.* performed experimental simulations that showed an optimal value for the smoothing parameter to be $\lambda = 0.006$.

Equation 3.23 is used in this thesis to estimate the coefficients and equation 3.19 to estimate the diffusion ODF from the measured HARDI signal.

²For $j = \{1, 2, 3, 4, 5, 6, 7, 8, \dots\}$, $l_j = \{0, 2, 2, 2, 2, 2, 4, 4, \dots\}$

3.3 Summary of Methods

Table 3.1 summarizes the acquisition requirements for the different diffusion imaging modalities introduced in this chapter. For a complete overview DWI and ADC were added to the table. The HARDI methods (below the horizontal line in table 3.1) make higher demands on data acquisition at the benefit of providing more possibilities to extract useful information from the obtained data. QBI, DOT, PAS and SD have all been optimized to collect the data in a clinically feasible time. Only DSI with the required high b -values and long acquisition time lacks of practical use for in-vivo data acquisition.

Diffusion MRI modality	Gradient strength (s/mm^2)	Number of measurements N	Acquisition time (min)
DWI	$b \leq 1000$	$N = 1$	1 – 3
ADC	$b \leq 1000$	$2 \leq N \leq 4$	2 – 4
DTI	$b \leq 1000$	$N \geq 7$	3 – 6
QBI, DOT, PAS, SD	$b \geq 1000$ ($b \geq 3000$ desirable)	$N \geq 60$	10 – 20
DSI	$b > 8000$	$N \geq 200$	15 – 60

Table 3.1: Comparison of acquisition requirements and acquisition time for different diffusion MRI modalities. The number of measurements is in most cases the number of applied diffusion encoding gradients. In DSI N is the number of different q -values (i.e. number of gradient directions and different b -values). The indicated times assume an acquisition of 30 axial slices with thickness of approximately 3 mm each. Table adopted and modified from [Des07, p. 51].

With these new methods there is no need to hold on to the established but error-prone fiber reconstruction techniques based on DTI. Rather, new HARDI algorithms need to be developed to fully use the potential of these acquisition methods.

The performance of the different HARDI methods is compared in table 3.2. The horizontal line separates model-based (above the line) and model free approaches (below the line). All model-based approaches have low requirements on acquisition but also tend to be very biased. The reason for this is obvious: If the model is designed to find two fibers in a voxel, it will - no matter how many fibers really exist. For this reason model-based approaches are of low practical relevance for fiber reconstruction.

For this thesis a method with low acquisition requirements and low computation time was desired. DSI was rejected due to the first criterion and PAS due to the second one. While SD, and especially cSD, provide good results, a major disadvantage of these approaches is the required a-priori assumption on the diffusion kernel for deconvolution. Of the re-

3.4. VISUALIZATION

maining two methods (QBI and DOT) QBI was chosen due to its simplicity and rich amount of studies which promise good results if the b-value and number of gradient directions are sufficiently high. However, due to its robustness to noise and lower acquisition requirements [OSV⁺06] DOT could gain more importance in future.

Method	Acquisition requirement	Computation time	Accuracy	Bias
Two tensors	low	medium	medium	high
Ball and stick	low	medium	medium	high
Compartment	low	medium	medium	high
DSI	very high	medium	high	medium
QBI	medium	low/medium	medium	medium
DOT	medium	medium	medium	medium
PAS	medium	high	high	low
SD	medium	low/medium	medium	medium
cSD	medium	medium	medium	low

Table 3.2: Properties of different HARDI reconstruction techniques. Table adopted and modified from [JBB09, p. 66].

3.4 Visualization

This section provides an overview of typical visualization techniques for DT images according to [VZKL06] and [JBB09]. Modifications of these techniques are used for HARDI visualization and will also be mentioned.

The main differences between the distinct visualization means are the dimensionality to which the voxel information (either the tensor or the ODF) is reduced. A second characteristic is whether local or global information is displayed. Here, global means taking the relationships to neighboring voxels into account. In contrast to that local visualization considers only the data belonging to one single voxel.

3.4.1 Scalar Indices

Scalar indices reduce the data to one dimension, a scalar. Before the DTI model was introduced the *ADC* was used to visualize diffusion weighted images. The main drawback of the *ADC* is its directional dependence. As can be seen in Figure 3.12 the *ADC* images alters greatly as the axis of diffusion weighting is changed. In contrast to this scalar indices that are derived from the diffusion tensor are rotationally invariant. The rightmost panel in Figure 3.12 shows the mean of the three *ADC* images, the mean

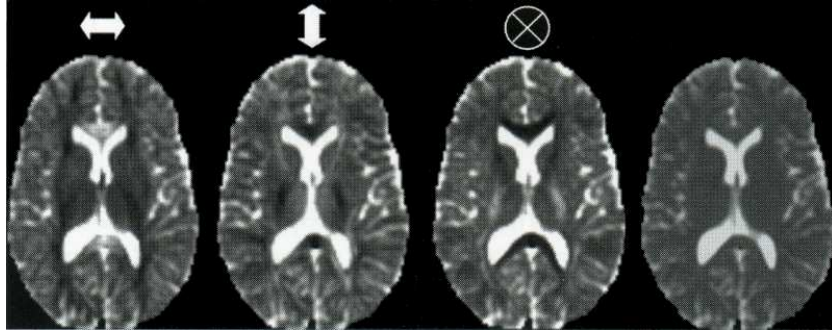


Figure 3.12: The effect of varying the diffusion encoding gradient is shown on the three left images. The arrows indicate the gradient orientation. The rightmost image illustrates the mean diffusivity. Image adopted from [JBB09, p. 45].

diffusivity (MD). MD can be derived from the diffusion tensor as

$$MD = \frac{D_{xx} + D_{yy} + D_{zz}}{3} \quad \text{or} \quad MD = \frac{\lambda_1 + \lambda_2 + \lambda_3}{3} \quad (3.24)$$

The sum of the three diagonal elements of the diffusion tensor is also called the *trace*. MD images serve to distinguish between cerebrospinal fluid (high values) and brain tissue (lower values). However, different kinds of brain tissue or structures can not be distinguished that way.

Therefore, Basser and Pierpaoli introduced two indices for that purpose: the fractional anisotropy (FA) and the relative anisotropy (RA) [BP96]. These two are calculated as

$$FA = \sqrt{\frac{\frac{3}{2} \sqrt{(\lambda_1 - MD)^2 + (\lambda_2 - MD)^2 + (\lambda_3 - MD)^2}}{\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}}} \quad (3.25)$$

and

$$RA = \sqrt{\frac{\frac{1}{3} \sqrt{(\lambda_1 - MD)^2 + (\lambda_2 - MD)^2 + (\lambda_3 - MD)^2}}{MD}} \quad (3.26)$$

The numerator of both the indices is the standard deviation of diffusion from the mean diffusivity. The denominator of the FA index is the magnitude of the tensor (the square root of the sum of squares of its eigenvalues). FA normalizes the standard deviation of diffusion by the magnitude of the tensor. It measures the fraction of the tensor that can be assigned to anisotropic diffusion. This index is normalized to take values from zero (isotropic diffusion) to one (diffusion along one axis only). FA is the mostly used scalar index in diffusion tensor imaging. An example FA image is presented in Figure 3.13b. In the RA index the standard deviation is divided by the mean diffusivity. This index is normalized to take values from zero to one. For more scalar indices please refer to [VZKL06].

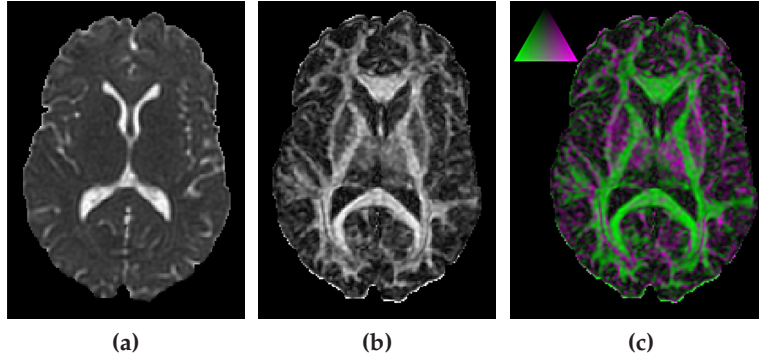


Figure 3.13: Comparison of MD (a), FA (b) and Westin metrics (c) on an axial slice of the brain. In (c) green indicates c_l and violet indicates c_p . Images adopted from [VZKL06].

So far the presented scalar indices indicated the mean diffusivity (MD) or the deviation from isotropy (FA, RA). These indices reduce the tensor to one scalar and, thus, provide no information on the shape of the tensor ellipsoid. The shape can take three different forms that depend on the eigenvalues. If the eigenvalues are sorted according to $\lambda_1 \geq \lambda_2 \geq \lambda_3$ the three forms are

- sphere-shaped ($\lambda_1 \approx \lambda_2 \approx \lambda_3$, isotropic diffusion)
- pancake-shaped ($\lambda_1 \approx \lambda_2 > \lambda_3$, planar anisotropy)
- cigar-shaped ($\lambda_1 > \lambda_2 > \lambda_3$, linear anisotropy)

To characterize these shapes Westin *et al.* proposed three separate indices: spherical anisotropy (C_s), planar anisotropy (C_p), and linear anisotropy (C_l) [WPG⁺97]:

$$C_s = \frac{\lambda_3}{MD} \quad C_p = \frac{2(\lambda_2 - \lambda_3)}{3MD} \quad C_l = \frac{(\lambda_1 - \lambda_2)}{3MD} \quad (3.27)$$

These three indices sum to one and therefore parameterize a barycentric space. A comparison of MD, FA and the Westin metrics is given in Figure 3.13.

An alternative approach is considering the third moment (skewness) of the three eigenvalues [Bas97]:

$$S = \frac{(\lambda_1 - MD)^3 + (\lambda_2 - MD)^3 + (\lambda_3 - MD)^3}{3} \quad (3.28)$$

For cigar-shaped ellipsoids S is positive, while it is negative for pancake-shaped ellipsoids. However, using higher order moments of eigenvalues augments the sensitivity to noise.

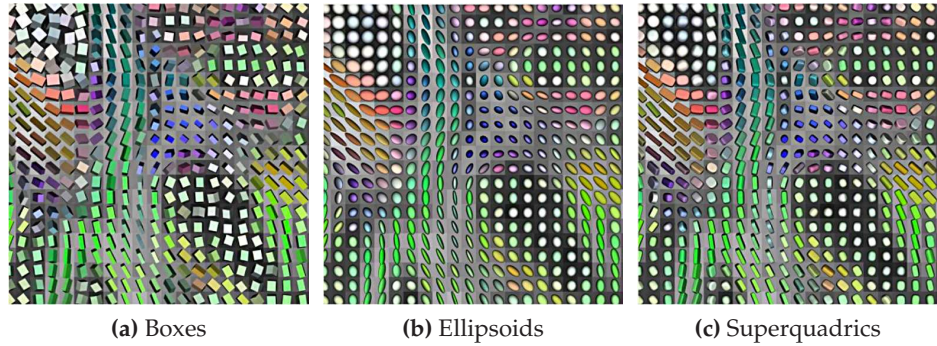


Figure 3.14: Different glyph shapes for DTI visualization. Images adopted from [VZKL06].

Since the calculation of all of these indices is based on the diffusion tensor a separate index to characterize anisotropy in HARDI data, the so called *general fractional anisotropy* (GFA), was proposed in [Tuc04]:

$$GFA = \sqrt{\frac{n \sum_{i=1}^n (\Psi(\mathbf{u}_i) - \langle \Psi \rangle)^2}{(n-1) \sum_{i=1}^n \Psi(\mathbf{u}_i)^2}} \quad (3.29)$$

Here, $\Psi(\mathbf{u}_i)$ is the dODF value at the direction vector \mathbf{u}_i and $\langle \Psi \rangle$ is the mean of all dODF values in the voxel. Like the FA the GFA normalizes the standard deviation of diffusion by the magnitude of the dODF.

When designing algorithms for HARDI data it is of great importance to classify the number of fiber populations in a voxel. Scalar indices for this purpose have been proposed. As their primary application is classification rather than visualization they will be discussed in section 4.1.3 when dealing with tractography on HARDI data.

3.4.2 Glyphs

To visualize the whole information contained in a voxel, the so called *glyphs* are employed. A glyph is a graphical object that describes diffusion in a voxel with its parameters (e.g. size, color, shape). The diffusion ellipsoid treated earlier in section 3.1 is most commonly used to visualize the diffusion tensor. The ellipsoid's shape varies depending on the size of the three eigenvalues. The size of the glyph can be used to display the mean diffusivity or be normalized to view the anatomy and pathology of regions with uniform glyphs.

Apart from ellipsoids other glyph shapes like boxes or superquadrics were proposed for diffusion tensor visualization [Kin04]. A comparison between these three shapes is given in Figure 3.14. The glyph's sizes are normalized and the colors indicate their principal orientation. While boxes

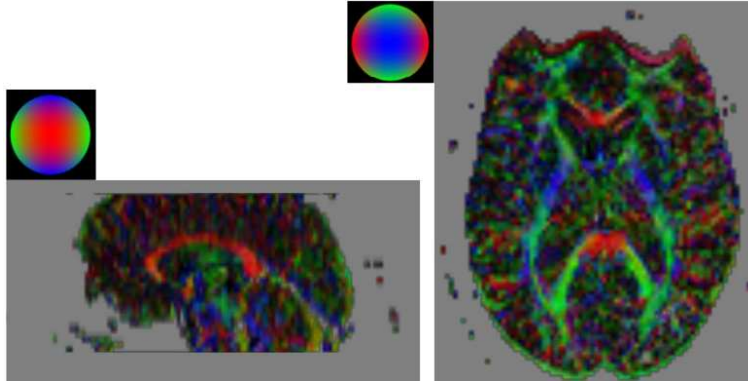


Figure 3.15: Color coded orientation of the first eigenvector of diffusion tensors on a sagittal slice (left) and an axial slice (right). Images adopted from [VZKL06].

contain only a few polygons and are therefore fast to render their orientation is poorly defined in regions of isotropic diffusion. The different shapes of the ellipsoids are well defined but can be difficult to tell apart. The proposed superquadrics try to combine the positive properties of these two shapes.

The main difference between tensor ellipsoids and glyphs for HARDI visualization is that HARDI glyphs are not restricted to a geometric shape like an ellipsoid. The individual shape of the HARDI glyphs stems from the ODF values for each sampled direction that is rendered as a radius in the corresponding direction. HARDI glyphs have already been depicted on several images in section 3.2 (e.g. rightmost column in Figure 3.2).

3.4.3 Orientation

The diffusion tensor can be reduced to the direction of the main eigenvector (i.e. the eigenvector corresponding to the biggest eigenvalue). This method is based on the assumption that the main eigenvector, \mathbf{e}_1 , defines the orientation of the underlying brain structure. As was proposed in [PP99] the orientation of this vector can be mapped to a primary color. Further, each of the colors is assigned to one of the principal axes. The components of the RGB channel can, thus, be calculated as

$$R = |\mathbf{e}_1 \cdot \mathbf{x}| \quad G = |\mathbf{e}_1 \cdot \mathbf{y}| \quad B = |\mathbf{e}_1 \cdot \mathbf{z}| \quad (3.30)$$

where \mathbf{x} , \mathbf{y} , and \mathbf{z} are the normalized direction vectors of the main axes. Figure 3.15 illustrates an example on a sagittal and axial slice of the brain. The color coded images result in smooth color transitions along axonal fibers in the brain providing information on coherence between neighboring voxels.

To visualize the orientation of HARDI glyphs the main eigenvector is replaced by each sampling direction. The glyphs' surface in this direction

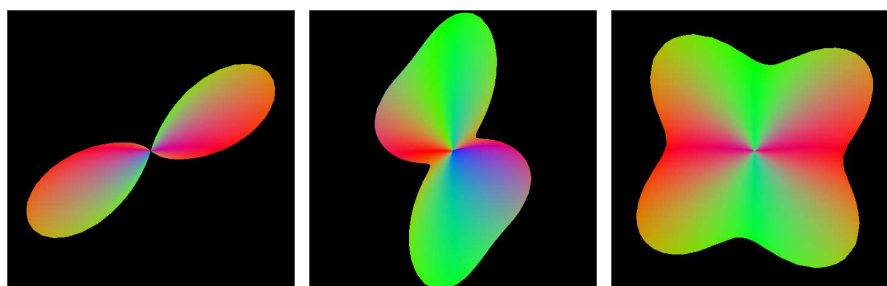


Figure 3.16: HARDI glyphs with orientation color coding (axial view).

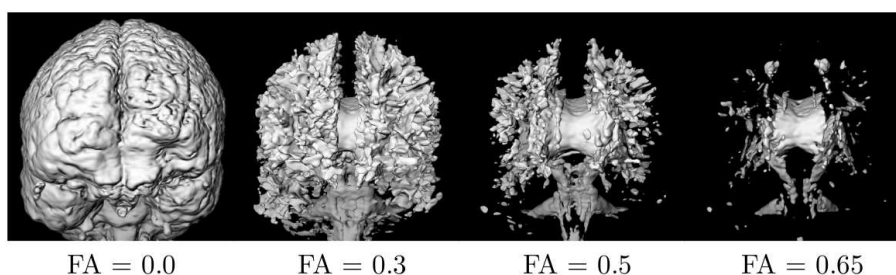


Figure 3.17: Volume-rendered isosurfaces of FA values showing 3D white matter structure. Image adopted from [VZKL06].

is rendered in the color arising from the above equation. Figure 3.16 shows some HARDI glyphs with orientation color coding.

Fiber tracing or tractography is a technique to determine the main orientation of fibers in a voxel and to reconstruct whole trajectories of fiber bundles. The reconstructed fibers are often rendered as three dimensional tubes for visualization. Since fiber tracing is the main topic of this thesis, it will be treated separately in chapter 4.

3.4.4 Volume Rendering

Volume rendering is widely used in medical visualization to illustrate data in 3D. A so-called *transfer function* assigns color and opacity to voxels depending on the properties of the particular data to display. In DTI and HARDI scalar indices, especially the FA or the GFA, are important properties of the acquired data that are visualized.

Figure 3.17 is rendered with isosurfaces of FA. The opacity of a voxel is set to 0 if the FA is below the given threshold and 1 otherwise. Color can be assigned to volume-rendered voxels to depict the shape or orientation of underlying brain structures (left image of Figure 3.18). Another application is to indicate the type of diffusion. The right image in Figure 3.18 is colored according to Figure 3.13c showing computed scalar indices c_p and c_l . In this manner insight in structure dependent diffusion properties can be gained.

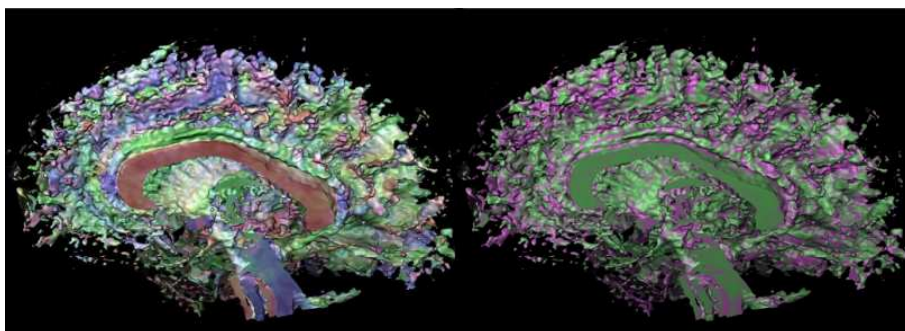


Figure 3.18: Color can be used in volume rendering to visualize the orientation of the principal diffusion direction (left). Also scalar indices can be used to depict the type of diffusion. The right image shows Westin metrics c_p (magenta) and c_l (green). Image adopted from [VZKL06].

Volume rendering provides lots of possibilities for visualization. It can be overlaid with tensor glyphs or other data to emphasize important structures. In tractography (chapter 4) axonal fiber bundles are rendered in different colors to visualize their trajectory in the brain and to separate them from adjacent structures or other fiber bundles.

Chapter 4

Tractography

Tractography (also called *fiber tracing* or *fiber tracking*) is a general term for methods to reconstruct fiber pathways in the white matter based on diffusion imaging. It offers a unique possibility to gain insight into the structure of the human brain noninvasively and *in vivo*. The information won in this manner is not only of high value for visualization of the brain structure and segmentation of the brain into different functional areas, it also provides essential information for neurosurgical planning and investigations of multiple sclerosis and Alzheimer's disease.

Promising as it is tractography is based on diffusion imaging data that itself is difficult to interpret and error prone. The reconstruction of fibers in fiber tracing is based on the assumption that the diffusion of water molecules is greater along white matter pathways than across them. Moreover, even with the most accurate MRI scanners individual fibers can not be reconstructed due to their low diameter that lies on a microscopic scale. The structures of interest are therefore fiber bundles often consisting of thousands of individual fibers. However, when using fiber tracing methods one should always keep in mind that the reconstructed entity are not fiber bundles, but rather pathways of least hindrance to diffusion. Recent studies show great correlations between the measured diffusion and actual fiber bundles [Bea02], [JBB09], thus, further encouraging the development of tractography methods.

Validation of tractography is an important issue. Anatomical knowledge can be obtained by post-mortem dissection. Reconstructed fiber pathways can then be compared with the found structures. However, fiber trajectories can vary in different subjects. *In vivo* validation is performed by injecting contrast media into a subject and register its dispersion. Further validation methods are the so called *phantoms*. Phantoms are artificially constructed fiber configuration (e.g. from rat spinal cord or acrylic fibers) with known ground truth. A phantom with challenging fiber configuration was used at the Fiber Cup to evaluate state-of-the-art tractography

algorithms. The Fiber Cup is a tractography contest held at the MICCAI 2009¹ conference in London, UK. This phantom data is available online at the Fiber Cup website² and will be used for evaluation of the implemented algorithms in chapter 6. Some of the algorithms evaluated at the Fiber Cup will be presented in this chapter.

There are several ways to characterize tractography methods. Firstly, fibers might be reconstructed in a deterministic or a probabilistic way. Deterministic methods do not take into account the uncertainty of the fiber direction. The resulting pathway might or might not be a true pathway in the brain. In contrast to this, probabilistic methods often generate several pathways and assign probabilities to them.

Secondly, fiber tracing can be local or global. Local methods only consider the information of one voxel (and sometimes its neighbors' information) at each step. Global methods, however, require a property that represents global information along the whole fiber tract. The front evolution approach presented later in this chapter is an example for global tractography.

Lastly, the reconstruction methods can be model based or model free. This characterization is the most vague in the literature. As mentioned in chapter 3 several ways exist to represent the diffusion signal in each voxel. These were also classified as model based (e.g. DTI, multi tensor) and model free (e.g. Q-ball). However, in literature in some cases even the ODF resulting from the Q-ball reconstruction is considered as a model. Therefore, this characterization will not be applied to the methods presented later in this chapter. Instead, it will be mentioned each time whether the algorithm works on DTI data or on an ODF from HARDI data.

This chapter presents global and local, deterministic and probabilistic state-of-the-art fiber tracing algorithms. The first section introduces the concept of the *streamline* and presents early fiber tracing algorithms based on DTI data. Subsequently, examples for recently proposed algorithms on HARDI data will be given. Section 4.2 introduces the theory of uncertainty representation and provides examples for probabilistic algorithms, both on DTI and HARDI data. Approaches to globally reconstruct fiber pathways will be presented in section 4.3. The last section provides a summary of the presented methods and the theoretical part of this thesis. References to the original research papers will be given at each example. A general overview of these methods can be found in [JBB09].

¹<http://www.miccai2009.org/>

²<http://www.lnao.fr/spip.php?rubrique79>

4.1 Local Deterministic Tractography

The first fiber tracing approaches were based on DTI. Although they reconstructed fiber pathways in different ways they all were based on streamline tracing. The first section on deterministic tractography introduces the concept of the streamline and some early algorithms. The tensorline tracing developed later will be introduced in the second section. A review of the different DTI based fiber tracing approaches can be found in [MZ02]. The last section on deterministic tractography presents some approaches to reconstruct fibers from HARDI data. In contrast to DTI methods no well-established algorithms exist for fiber tracing on HARDI data. Further, there are no comprehensive reviews on current methods applied at that field. Therefore, no reference to literature providing an overview on different HARDI algorithms can be given.

4.1.1 Streamline Tracing

A streamline is a line whose tangent is always parallel to a vector field. In DTI, a vector field is defined by the first eigenvector of each voxel. Voxelwise integrating the fiber orientations into a path, i.e. starting at a voxel and following the first eigenvectors, is an intuitive way to reconstruct fiber tracts.

Basser et al. represent a streamline mathematically as a 3D space-curve [BPP⁺00]:

$$\frac{d\mathbf{r}(s)}{ds} = \mathbf{e}_1(\mathbf{r}(s)) \quad (4.1)$$

The location of the streamline, \mathbf{r} , is a function of the distance, s , along the streamline from the starting voxel (*seed voxel* or *seed point*). The first eigenvector, \mathbf{e}_1 , of the diffusion tensor at \mathbf{r} is the local estimate of the fiber orientation. This vector is the tangent, $\mathbf{t}(s)$, of the streamline in s (Figure 4.1a). For other vector field representations than DTI the first eigenvector, \mathbf{e}_1 , can be substituted by the corresponding fiber orientation estimate, \mathbf{e} .

Since equation 4.1 is a differential equation it is not possible to calculate the location of the streamline at any distance s directly. This equation only allows to determine how the location of the streamline should change depending on s . Consequently, errors made in the calculation of each individual step will sum up and propagate to further steps, potentially leading to wrong results.

Apart from errors made during integration further errors stem from noise in the imaging process, potentially leading to a wrong estimation of the principal diffusion direction (PDD). Also, the choice of the model to represent the data can bear severe limitations (e.g. DTI can only represent one fiber direction per voxel). Since these errors do not arise from fiber tracing they will not be discussed here. For more details on error minimization

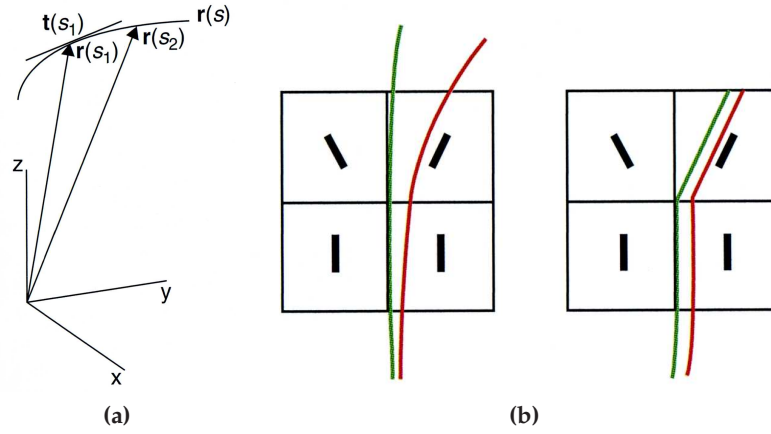


Figure 4.1: Image (a) shows a mathematical representation of a streamline. Streamlines can be propagated through a voxel with (image (b) left) or without interpolation (image (b) right). Image adopted from [JBB09, p. 335].

in the imaging process please refer to the references given in chapter 2. The selection of a data representation and the resulting errors were discussed extensively in chapter 3.

To minimize errors from integration the vector field needs to be interpolated. The right-hand side of equation 4.1 is defined in continuous space. To progress the streamline correctly the exact fiber orientation is needed at each position s . The diffusion data, however, is only available on a discrete voxel grid providing only one fiber orientation per imaging voxel. Assuming that fiber orientations contain contributions from all neighboring voxels the data can be interpolated to gain continuous fiber orientations within a voxel (Figure 4.1b).

The left-hand side of equation 4.1 demands for infinitesimally small steps along the streamline. Obviously, in practice the step size has to be bigger. The question raised here is how to discretize the continuous vector field. It can be answered by the proper choice of an integration method. The most commonly used integration methods are the *Euler* method and the fourth order *Runge-Kutta* method (RK4). In the Euler method the fiber orientation is assumed to be constant along the chosen step size h . The succeeding position is calculated as

$$\mathbf{r}(s + h) = \mathbf{r}(s) + h \cdot e(\mathbf{r}(s)) \quad (4.2)$$

The Runge-Kutta method also considers variations of the diffusion orien-

tation along the step size:

$$\mathbf{r}(s+h) = \mathbf{r}(s) + \frac{1}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4) \cdot h$$

where

$$\begin{aligned} \mathbf{k}_1 &= e(\mathbf{r}(s)) \\ \mathbf{k}_2 &= e(\mathbf{r}(s) + 0.5h \cdot \mathbf{k}_1) \\ \mathbf{k}_3 &= e(\mathbf{r}(s) + 0.5h \cdot \mathbf{k}_2) \\ \mathbf{k}_4 &= e(\mathbf{r}(s) + h \cdot \mathbf{k}_3) \end{aligned} \quad (4.3)$$

In both cases, smaller step sizes lead to less errors in the reconstructed fiber bundle trajectory.

A streamline is stopped as soon as it arrives at a point where its trajectory can no longer be trusted. As no probabilities for fiber tracts are calculated in deterministic tractography other methods are applied. Usually, streamlines terminate when entering a voxel with a low FA value. In regions with low FA values the uncertainty of the fiber orientation is high. Terminating the streamline at such a point prevents the trajectory from acquiring high directional errors. Further, low FA values indicate that the trajectory has left white matter and entered gray matter where no fibers exist. A second heuristic to stop a streamline is a high angle between two successive steps. White matter fibers usually have high curvature radii and do not bend through high angles on the scale of a voxel. However, if such a high angle is encountered, it is most probably due to errors made in previous steps than to a real fiber bundle bending that way.

Streamline Algorithms

This section presents some algorithms based on the here described techniques. The focus lies on classic approaches to reconstruct fiber bundles from DTI data.

FACT One of the first fiber tracing approaches was presented by Mori *et al.* on a rat brain [MCCZ99]. They state that the most intuitive way to trace fibers is by connecting each voxel to the adjacent one toward which the PDD is pointing (Figure 4.2a). Since this is highly erroneous Mori *et al.* propose a method which they call *fiber assignment by continuous tracking* (FACT). According to FACT, a fiber trajectory is traced continuously to the next voxel border entering the new voxel at the same position where the old one was left (Figure 4.2b). FACT is basically continuous tracing on a discrete vector field, since no interpolation is applied. The streamline continues as long as the sum of inner products of the adjacent vectors is above a threshold, i.e. high confidence in the fiber orientation exists (Figure 4.2c). However, when the orientation of adjacent direction vectors becomes random the streamline is terminated (Figure 4.2d). The step size along the streamline varies depending on the distance to the next voxel border.

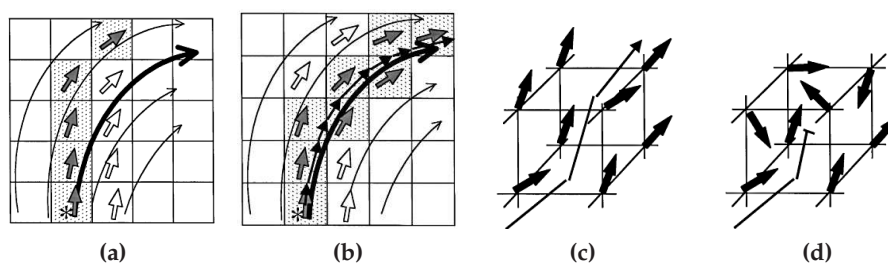


Figure 4.2: Principle of FACT. Long arrows in (a) and (b) indicate the underlying fiber trajectory, short arrows the PDD in each voxel. Tracing starts in the voxel marked with * and continues along gray marked voxels. As long as the PDDs of neighboring voxels show in a similar direction tracing continues (c). Otherwise, the streamline is stopped (d). Image adopted from [MCCZ99].

DTI Interpolated Conturo *et al.* performed fiber tracing on human brain data [CLC⁺99]. They interpolated the acquired MRI data with uniform 2.5 mm voxels to obtain smaller voxels of 1.25 mm size. Tracing was performed on the interpolated data with a constant step size of 0.5 mm. A FA threshold was used as a stopping criterion.

Streamlines With B-splines The mathematical concept of the streamline as presented here was introduced by Basser *et al.* [BPP⁺00]. They generated a continuous tensor field by tensor interpolation. The streamline was integrated with the RK4 method using a dynamic step size that depended on the local curvature of the streamline. Further, to diminish the influence of imaging noise the trajectory was locally approximated with B-spline functions.

Streamlines Least Squares Filter Zhukov *et al.* presented another approach to deal with noise [ZB02]. They used a local regularization based on a least squares filter. By finding a low degree polynomial that best matched the data around the tracing position they approximated the data to filter out noise induced perturbations. Tracing was performed on a component-wise interpolated tensor field with Euler integration.

Multi ROI FACT A different method to find fiber bundles was presented by Stieltjes *et al.* [SKZ⁺01]. Fiber tracing was performed using FACT without interpolation. Streamlines were started from every white matter voxel in the dataset. Additionally, anatomical knowledge was used to place two regions of interest (ROI) to define known pathways. Only streamlines passing through both ROIs were retained for the final result (Figure 4.3).

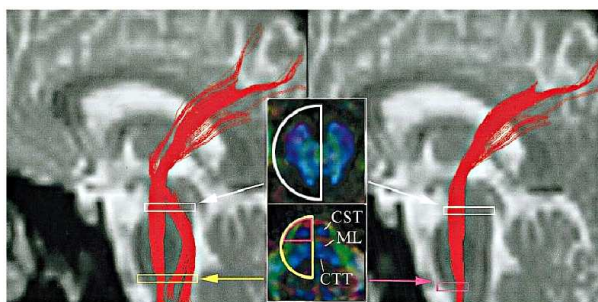


Figure 4.3: Multi ROI fiber tracing. Only streamlines passing through both ROIs are retained. Image adopted from [SKZ⁺01].

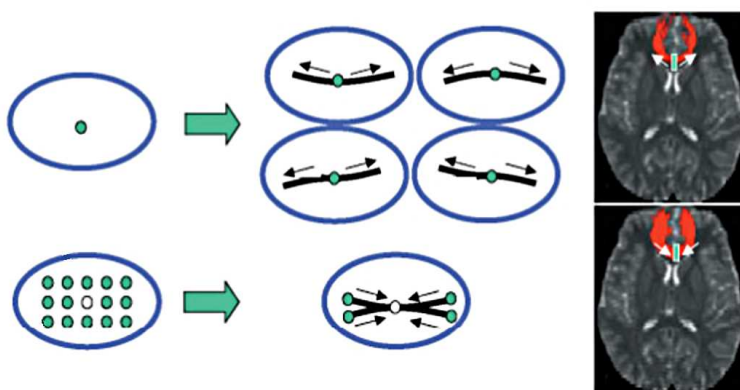


Figure 4.4: Fiber tracing through regions with fiber branchings can lead to different reconstructed trajectories without detecting the fiber branching (upper part of the image). Initiating tracing in all voxels, however, detects branchings by merging fibers (lower part of the image). Image adopted from [SKZ⁺01].

Summary

All of these methods were able to reconstruct major pathways that were consistent with prior anatomical knowledge. The reconstruction worked best in large fibers with only one dominant fiber direction. In areas with fiber crossings or kissings no reliable directional information could be determined and tracing had to be terminated. Fiber branchings, however, were detected in some cases. A method starting fiber tracing at all voxels in the brain, as it was done in [SKZ⁺01], is able to reveal fiber branchings. This is due to the fact that merging two fiber can be handled much easier than branching fibers (Figure 4.4). Of course this approach demands for a significantly higher computation time.

4.1.2 Tensorline Tracing

All methods introduced so far had one common property: They used only the first eigenvector of the diffusion tensor. As long as the tensor was cigar-shaped, i.e. the first eigenvector was well defined, tracing continued along the indicated path. However, in pancake-shaped or sphere-shaped tensors tracing had to be terminated in order not to make major integration errors due to a poorly defined principle diffusion direction.

TEND Other methods were proposed that use the entire diffusion tensor. These are called *tensorline tracing*, *tensor deflection* or short *TEND* [WKL99], [LWT+03]. The fiber trajectory was named *tensorline*. In this approach the fiber path does not automatically follow the first eigenvector of a voxel. Rather, it is deflected into the direction of the first eigenvector to a degree depending on the anisotropy of the tensor, \mathbf{D} , (Figure 4.5a). In voxels with high anisotropy the deflection is strong, whereas sphere-shaped tensors are passed without changing the direction of the fiber pathway. The outgoing direction of a voxel, \mathbf{v}_{out} , is calculated as

$$\mathbf{v}_{out} = f\mathbf{e}_1 + (1 - f)((1 - g)\mathbf{v}_{in} + g\mathbf{D}\mathbf{v}_{in}) \quad (4.4)$$

where \mathbf{v}_{in} is the incoming direction and f and g are user defined parameters between 0 and 1 to weight the influence of the first eigenvector, \mathbf{e}_1 , and the prior tracing direction respectively.

Figure 4.5b illustrates on simulated data that tensorlines are able to propagate through isotropic voxels, thus, stabilizing the fiber pathway. On

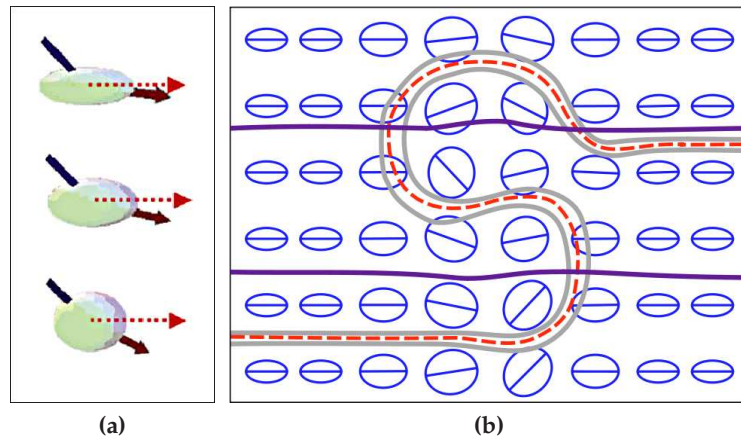


Figure 4.5: The outgoing vector (dark red) is deflected in the direction of the PDD (dotted red arrow) depending on the shape of the tensor (a). Tensorlines (violet) can propagate through isotropic voxels while a classic streamlines is diverted (dotted red line) (b). Image (a) adopted from [LWT+03], image (b) from [WKL99].

the other hand, by simply continuing pathways through isotropic voxels tensorlines can reconstruct fiber trajectories that do not really exist. Further, tensorline propagation performs well on straight pathways but tends to underestimate the curvature of bending fibers [LWT⁺03].

Lazar and Alexander compared the performance of streamlines and tensorlines [LA03]. In their analysis each of the two methods was tested with Euler and RK4 integration on interpolated tensor fields. Additionally, fiber reconstruction using FACT without interpolation was tested. Streamlines and tensorlines performed equally on a straight tensor field. However, both methods led to better results with the Euler and RK4 integration method than with FACT that does not use interpolation. On a circular tensor field streamline tracing provided in general better results than tensorline tracing. For both methods best results were achieved with RK4 and FACT while Euler underestimated the curvature. In summary, best tracing results can be expected from streamlines integrated with the RK4 method on an interpolated tensor field. However, tensorlines better propagate through crossing fibers where anisotropy is low.

4.1.3 Deterministic HARDI Tractography

There are two major differences to DTI data which need to be considered when designing fiber tracing algorithms on HARDI data. First, unlike the first eigenvector in DTI, there is no evident principal diffusion direction in HARDI. Instead, the diffusion information is sampled in several hundreds of directions uniformly distributed on the sphere. This raises the question of which direction to choose. Often, the direction indicating the highest ADC is chosen resembling the first eigenvector of the DT. Of course, many other possibilities exist which should not be ignored as doing so would waste the advantages over DTI tracing.

The second difference is the ability of HARDI data to resolve multiple fiber populations per voxel. Currently, most approaches resolve up to two fiber populations. With improved imaging techniques (i.e. higher b-values and better SNR) and therefore higher possible SH orders in the Q-ball reconstruction the ability to find three or even more fiber populations per voxel in anatomically complex regions of the brain could become common. The former question in DTI of whether a voxel features isotropic or anisotropic diffusion is extended to the question of how many fiber populations are present in a voxel. In DTI zero (isotropic voxel) or two fiber populations often result in the same diffusion tensor. In HARDI, however, these two cases have to be clearly separated.

The next section introduces methods to classify the number of fiber populations in a voxel. The subsequent section presents example algorithms to reconstruct fibers from HARDI data.

Voxel Classification

In DTI the FA value as presented in section 3.4.1 was used to separate white and gray matter. Its analogon, the GFA value, is widely used to separate isotropic from anisotropic voxels in HARDI. However, the GFA can be misleading in areas with multiple fiber populations especially in datasets with low b-values. Such voxels result in similar GFA values as gray matter and can not be told apart easily. To address the problem of fiber classification more methods were developed in the recent years. The fractional multi-fiber index (FMI) as proposed in [Fra02] determines the significance of a given order of SH functions by the fraction of the squared coefficients of different orders:

$$FMI = \frac{\sum_{\{j:l \geq 2\}} |c_j|^2}{\sum_{\{j:l=0\}} |c_j|^2} \quad (4.5)$$

where c_j is the j -th coefficient in the modified SH basis as introduced in section 3.2.3 and l is the order of items in the SH series. It can indicate whether a voxel is isotropic or not. Determining a different fraction (e.g. order 4 coefficients and higher in the numerator and order 2 coefficients in the denominator) yields the significance of the 2nd order SH coefficients and, thus, whether the voxel contains one or more fiber populations. However, this measure can lead to incorrect results and the selection of proper thresholds is an open issue. Apart from being an anisotropy measure the FMI can also be used to estimate the proper order of SH functions to use in the applied reconstruction method.

Another classifier to determine the fiber population was introduced by Chen et al. [CGZ⁺04a] which incorporates the variance of the measurements.

$$R_0 = \frac{|c_0|}{\sum_j |c_j|} \quad R_2 = \frac{\sum_{\{j:l=2\}} |c_j|}{\sum_j |c_j|} \quad R_{multi} = \frac{\sum_{\{j:l \geq 4\}} |c_j|}{\sum_j |c_j|} \quad (4.6)$$

An algorithm to infer the correct number of fiber populations based on this classifier was presented in [DAFD06]: If R_0 is large or the variance is small the diffusion is considered to be isotropic. If R_2 is large a one-fiber population is present in the voxel. A large R_{multi} -value indicates two or more fibers' diffusion. Though, the great amount of words "large" and "small" in the formulation of the algorithm demand for more thresholds than FMI. However, finding appropriate thresholds is a nontrivial challenge.

Recently, Zhang *et al.* proposed a new algorithm to classify fiber populations in a voxel [HZZJ10]. This algorithm is based on the minimum description length (MDL) criterion and the Q-ball reconstruction as presented in [DAFD07]. Zhang *et al.* compute the minimum information entropy, $minE_j$, of each subset of coefficients, \hat{c}_j , according to

$$minE_j = \min\left\{(N - j) \ln \frac{(c^T c - \hat{c}_j^T \hat{c}_j)}{2(N - j)} + j \ln \frac{\hat{c}_j^T \hat{c}_j}{j} - \ln \frac{j}{N - j}\right\} \quad (4.7)$$

with c the $R \times 1$ coefficients vector, N the number of gradient directions and $j \in \{1, \dots, R - 1\}$. Each subset \hat{c}_j of coefficients contains the j first coefficients of the coefficients vector c . From the minimum E_j the j -th coefficient is obtained. The corresponding SH order, l , of this coefficient is used to derive the number of fiber populations in a voxel as $l/2$. Although, the results on simulated data are promising this criterion needs to be evaluated on real data.

Another recent approach to classify the fiber populations per voxel was proposed by Prčkovska *et al.* [PVP⁺09]. The dODF in each voxel is normalized and the number of dODF directions above a certain threshold (here: 0.6) is counted. Additionally to this proposed method the number of the directions above the threshold can be normalized within the whole ROI to lie between 0 and 1. Based on the resulting numbers for each voxel two thresholds can be applied to separate voxels with 0, 1 or 2 fiber populations.

All these scalar values are especially useful in cases of non optimal datasets acquired with low b-values and resulting in blurred dODFs. However, even in such cases they are surprisingly rarely used. Instead, current approaches try to sharpen the dODFs [DDKA09] or use cSD [JLTS09] reconstruction methods to obtain sharp dODFs in the first place.

Example Algorithms

As stated before no well-established algorithms to infer fibers from HARDI data exist. Further, the various reconstruction techniques introduced in section 3.2 provide an even greater variety of HARDI algorithms. Some examples for state-of-the-art algorithms based on different reconstruction methods will be given here. Since Q-ball reconstruction forms the basis of the implementation for this thesis several Q-ball based algorithms will be presented.

DSI Local Maxima Weeden *et al.* presented an algorithm based on DSI [WWS⁺08]. Data were acquired post-mortem from a macaque brain with $b_{max} = 40000$ and in-vivo on humans with $b_{max} = 8000$ and $b_{max} = 12000$. In each voxel a dODF, Ψ , was reconstructed with 362 sampling directions uniformly distributed on a sphere. From these directions only those were retained that pointed to local maxima. A local maximum direction is defined as a vector, \mathbf{U}_j , such that $\Psi(\mathbf{u}_i) < \Psi(\mathbf{U}_j)$ for every vector \mathbf{u}_i adjacent to \mathbf{U}_j . After reconstruction, fiber tracing was started from every voxel inside white matter bi-directionally following each local maximum vector \mathbf{U}_j . When reaching a new voxel the local maximum vector forming the least curvature with the previous one was selected as the new direction. The Euler method with a step size smaller than 0.5 voxel was used for integration. The tracing was aborted when white matter was left or when the

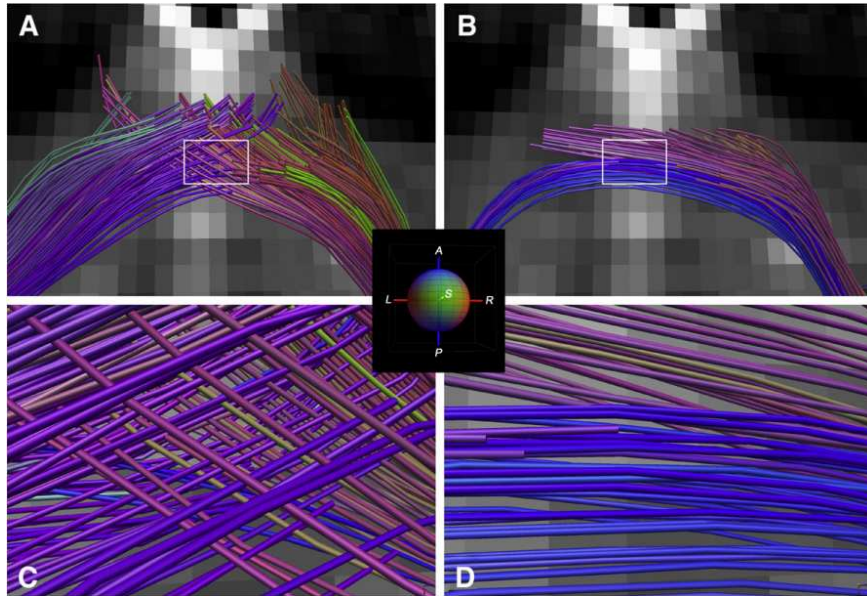


Figure 4.6: Comparison of DSI (left) and DTI (right) tracing results. DSI successfully reconstructed fiber crossings (panel (c)) where DTI only identified one fiber population (panel (d)). Image adopted from [WWS⁺08].

curvature inside a voxel was above a threshold. The results of this tracing algorithm were compared to the DTI algorithm presented in [CLC⁺99]. DSI was able to reconstruct many complex regions of the brain where DTI failed (Figure 4.6). The reconstructed pathways by DSI also matched anatomical knowledge confirming the validity of the applied approach.

PAS Local Maxima A similar local maxima approach based on the PAS reconstruction method was evaluated at the Fiber Cup and won the 5th place [Sak09]. Each voxel was sampled at 181 directions. Local maxima were determined by comparing each direction vector with its 8 nearest neighbors. In contrast to the approach of Weeden *et al.* only the two highest maxima from every voxel were retained as local fiber orientations. Streamlines were traced from every voxel inside the white matter. The longest fiber trajectories intersecting seed voxels were selected as representative fiber bundles.

Two Tensors Kalman Filter A two-tensor approach using a Kalman filter was employed for fiber tracing in [MSR09]. Every tensor was assumed to have a dominant principal diffusion direction, \mathbf{m} , with a corresponding eigenvalue λ_1 . The remaining eigenvalues were assumed to be equal ($\lambda_2 = \lambda_3$). Further, the two tensors in each voxel were equally weighted representing two compartments that equally contribute to diffu-

sion. A Kalman filter was used to estimate the system state (i.e. the model parameters \mathbf{m} , λ_1 , λ_2 of each tensor) at every step using the signal at a particular voxel. With a state transition function and an observation function the parameters for the next step were predicted. The Kalman filter fused the measured signal and the prediction to update the model at the current position. After moving a step in the resulting direction the procedure was repeated at the new position. In this approach, tracing was started from each voxel. Representative fibers were selected for each seed voxel after finishing the tracing procedure. This approach won the 3rd place at the Fiber Cup 2009. However, as the provided phantom data had only 3 slices the diffusion in this approach was constrained to two dimensions. Therefore, the proposed approach lacks a general usability and should be evaluated on human brain data without the $2D$ -constraint.

cSD Local Maxima Fiber tracing based on cSD placed 2nd at the Fiber Cup 2009 [JLTS09]. Streamlines followed a local maximum forming the least curvature with the incoming vector into the voxel. The signal was trilinearly interpolated at each position. Although this method is not very different from the local maxima approached presented above the sharp ODFs led to great success in reconstructing fiber pathways. However, a voxel for the deconvolution kernel representing the diffusion of a single fiber population had to be selected by hand. Again, the diffusion was constrained to only two dimensions and this approach can not be considered to work on real human brain data without further adjustments.

QBI Split Descoteaux *et al.* proposed an efficient Q-ball reconstruction method in [DAFD07]. Further, an additional sharpening of the reconstructed dODF was proposed in [DDKA09] to estimate a much sharper fODF. This sharpening transformation is based on spherical deconvolution. In contrast to other SD methods the deconvolution is not applied to the raw measured signal but rather to the reconstructed dODF. SD is, thus, applied merely as a sharpening method and not as a HARDI data reconstruction method. Based on this data a fiber tracing algorithm was proposed.

The fODF was sampled at 2562 directions uniformly distributed on the sphere. Further, the fODF was interpolated at each step of the Euler integration and normalized to the range $[0, 1]$. Diffusion tensors were also calculated at subvoxel precision to obtain a FA value. Tracing was stopped when the FA indicated that white matter has been left or when the curvature in a voxel exceeded 75 degrees. From the normalized fODF local maxima above 0.5 were selected. This thresholding was applied to avoid selecting small maxima that might be corrupted by noise. If more than one maximum existed tracing was split and continued along all present maxima.

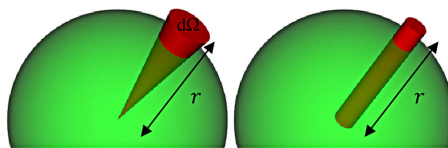


Figure 4.7: QBI with (left) and without (right) consideration of the quadric growth of the volume element of each ODF sampling direction. Image adopted from [ALS09].

QBI Solid Angle A variation of the last algorithm won the 4th place at the Fiber Cup 2009 [Goh09]. Instead of sharpening the reconstructed ODFs with SD, the reconstruction was performed with consideration of the solid angle as proposed in [ALS09] to obtain sharper dODFs. The solid angle takes into account the quadric growth of the volume element of each ODF sampling direction with its distance to the origin (Figure 4.7). The maxima selection in the tracing algorithm was modified as follows. After finding local maxima those were eliminated that were within 45 degrees of a greater maximum. Tracing followed maxima directions within 20 degrees of the incoming vector or within 60 degrees of the incoming vector if they had at least a magnitude of 85% of the maximum ODF value. If this filtering eliminated all possible outgoing directions tracing continued along the incoming direction vector if the corresponding ODF value was at least 70% of the maximum ODF value. Since several fiber trajectories per voxel are generated this way representative fiber pathways were selected along voxels that were intersected by the most fibers.

MFACT The FACT algorithm presented above has been adopted to HARDI data [CCC⁺08]. The extended algorithm is called *multiple FACT* (MFACT) as it is capable of following multiple directions per voxel. Like in FACT a streamline is continued to the edge of an adjacent voxel. Unlike FACT, MFACT follows every direction indicated by a local maximum direction that is above a user defined threshold and below a curvature threshold (Figure 4.8). The propagation of fiber tracts with MFACT can be compared to a region-growing method. The colored dots (red, orange, yellow, and green) in Figure 4.8 indicate hierarchically reached points that serve as starting points for further fiber tracing. The magnitude of the local maxima directions was used as a separating criterion for white and gray matter in this approach. Low maximum values indicated loss of directionality in diffusion and, thus, the end of white matter. As MFACT can generate lots of implausible directions multiple ROIs should be used to retain only fibers passing through all ROIs.

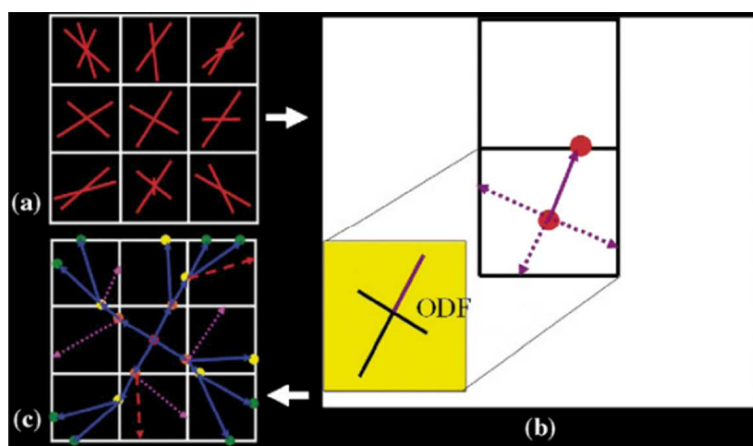


Figure 4.8: Panel (a) shows the local maxima of each voxel. Tracing is started in the central voxel along every local maxima direction (b). Panel (c) shows the streamlines after three iterations. Image adopted from [CCC⁺08].

4.1.4 Summary

In the recent years HARDI data has become the number one choice when designing new deterministic algorithm. There are only a few new deterministic algorithms that rely on the classic DTI model. The DTI algorithms presented here were proposed when HARDI techniques were still new and acquisition of large HARDI datasets was not possible in an acceptable time. However, most of the HARDI algorithms appear similar as they all use some or all local maxima for fiber propagation. The main differences are the choice of reconstruction methods and of appropriate thresholds. Although it might seem as there is no further need to develop new deterministic algorithms there is still no well established and well accepted deterministic HARDI algorithm. Further, as the Fiber Cup 2009 has shown, complex brain regions are still very challenging to reconstruct. The deterministic HARDI algorithms that placed 2nd and 3rd both used the a priori assumption that the phantom data was two-dimensional. This raises the question of whether or not they could reconstruct fiber pathways in real human brain datasets.

4.2 Local Probabilistic Tractography

Probabilistic tractography methods have been developed to overcome the limitations of deterministic algorithms. Resulting fiber trajectories from deterministic tractography have to be accepted as they are. There is no possibility to derive reliability information on these tracts. Fibers reconstructed from data with high SNR are presented with the same absolute certainty as fibers from noisy data. Although, the latter fibers are more likely to result in

implausible or even nonexistent trajectories. The goal of probabilistic tractography is, therefore, to provide uncertainty information for reconstructed fiber trajectories. Further, tracing shall not be stopped in uncertain regions but continue in the most plausible directions albeit with low certainty. As a result, probabilities can be assigned to every found fiber trajectories and it can be determined which trajectories are more probable than others. The price to pay for this additional information is a higher computation time that results from uncertainty calculation and propagation and the reconstruction of multiple streamlines.

The first step in probabilistic tractography is to represent uncertainty. This will be the topic of the first section. The second section deals with the propagation of uncertainty along the fiber trajectories. The last section provides examples for state-of-the-art algorithms. A more detailed overview of probabilistic tractography methods is given in [JBB09].

4.2.1 Representing Uncertainty

When designing probabilistic tractography algorithms the first step is to find a function to represent the uncertainty of the fiber orientations based on the available data. In [JBB09] this uncertainty function is called *uncertainty ODF* (uODF). In contrast to the dODF and the fODF, the uODF is not a physical property. It is merely a function representing the belief about the structure of the underlying tissue. For example, this function can state that the true fiber orientation lies within a certain area on the surface of the sphere with a given probability (Figure 4.9). In most cases not the whole uODF is needed but only samples of it. This section presents several methods to calculate and represent uncertainty that are used in state-of-the-art algorithms.

Bootstrapping

Theoretically, multiple repetitions of the original data could be acquired and the fiber orientation reconstructed in all datasets. These fiber orientations can then be considered as samples of the uncertainty. However, to build a representative uODF many samples are needed (about 50-1000). On the contrary, acquiring multiple repetitions of the same data is not feasible in practice as these acquisitions schemes would be very time consuming. Instead, the uODF can be calculated with bootstrapping. Bootstrapping is a method to take several samples from only a few available datasets. Since every dataset provides information on diffusion from different gradient directions these measurements can be randomly interchanged between the datasets to create samples [PB03], [Jon03]. For example, the first gradient encoding image is taken from dataset one, the second from dataset two, etc. (the selection is carried out randomly). This procedure is illustrated in Fig-

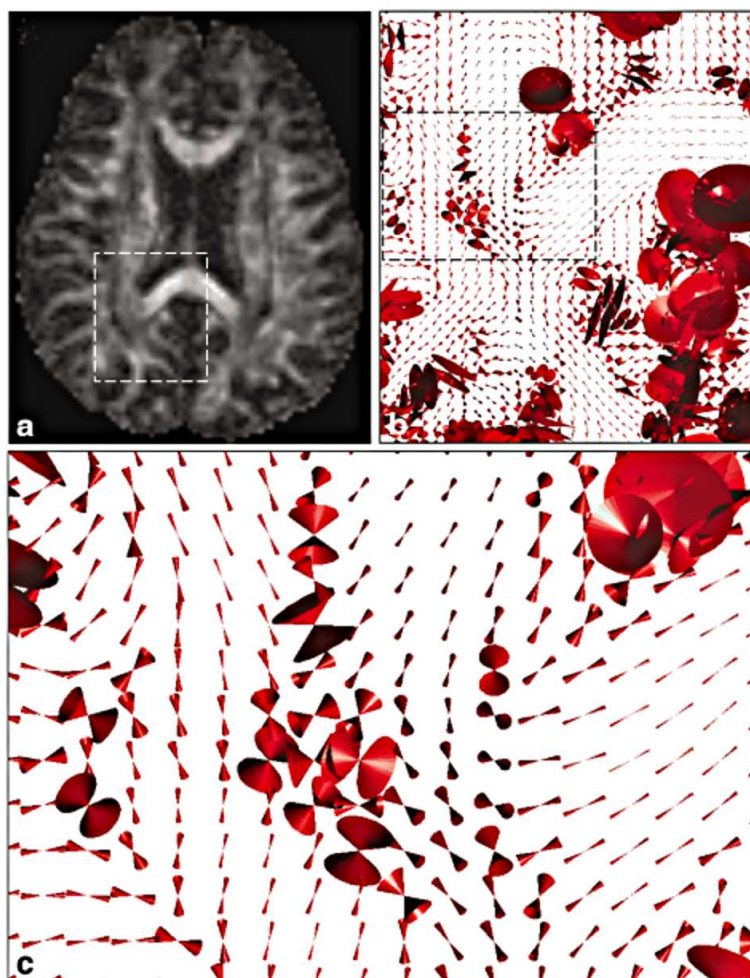


Figure 4.9: Illustration of 95% confidence intervals of the PDD. The radius of the cones represents the uncertainty. The radius is small in regions with one fiber populations and big in regions with complex fiber configurations. Image adopted from [Jon03].

ure 4.10a for three datasets. Figure 4.9 shows the 95% confidence intervals for the PDD generated with bootstrapping from two DTI datasets. Bootstrapping has two essential drawbacks. Although not many repetitions of the same data are needed, the procedure still demands for more than one data acquisition. The second drawback is that all uncertainty samples are derived from the same few datasets. As these datasets are not independent samples the uncertainty of the fiber orientation in the derived uODF is underestimated.

To overcome these limitations the use of the *residual bootstrap* (also called *wild bootstrap*) was proposed [Jon08], [BCM⁺08]. For this method only one dataset is required. From the measured signal data representing the diffu-

sion is reconstructed (e.g. with DTI or QBI). The reconstructed data (e.g. a diffusion ODF) does not fit the measured signal exactly due to noise and modeling errors. The residual is the (arithmetical) difference between the measured data and the reconstructed data. Samples are created by interchanging the residual between measurements. In [Jon08] the residuals are additionally multiplied randomly by 1 or -1 with equal probabilities.

Bayesian Methods

Bayesian methods are widely used techniques to calculate and represent probabilities in a broad variety of models. Probabilities are represented in form of a *posterior probability density function* (posterior pdf). This function can be made equivalent to the uODF by defining its parameters in spherical coordinates. Since mostly only samples of the uODF and not the whole function are needed most approaches use the Markov Chain Monte Carlo (MCMC) estimation methods. These methods allow to draw samples from a posterior distribution without the necessity to know all previous states of the system (e.g. the current state is sufficient to derive the posterior distribution).

The posterior pdf to encounter specific parameters Θ given some data D and a model M can be calculated as

$$P(\Theta|D, M) = \frac{P(D|\Theta, M)P(\Theta|M)}{P(D|M)} \quad (4.8)$$

Here, $P(\Theta|M)$ and $P(D|M)$ are the *a priori* probabilities (i.e. the prior beliefs) to encounter the parameters Θ and the data D given the model M respectively. $P(D|\Theta, M)$ is the *likelihood* of seeing the data D given the parameter values Θ .

To calculate the uODF in this manner all assumptions made in the modeling process need to be represented explicitly in the likelihood and the prior probabilities. The prior distributions represent known information about the parameters before any data is examined. A simple example for a prior probability is a fair dice roll. Before performing any experiments (i.e. rolling the dice) the probability of each possible outcome is known to be $1/6$. The likelihood has to include an assumption about the relationship between the estimated quantity (e.g. the fiber orientation) and the measured signal. Also, if noise is modeled in any way the likelihood has to include the corresponding assumptions as well. Again an example is taken from the dice roll. Given the data "even number" the probability for the outcome 2 is $1/3$ (in contrast to the prior probability of $1/6$). This approach is illustrated in Figure 4.10b.

The Bayesian approach results in a similar uODF for problems that can also be solved with a bootstrapping method. In general, however, the

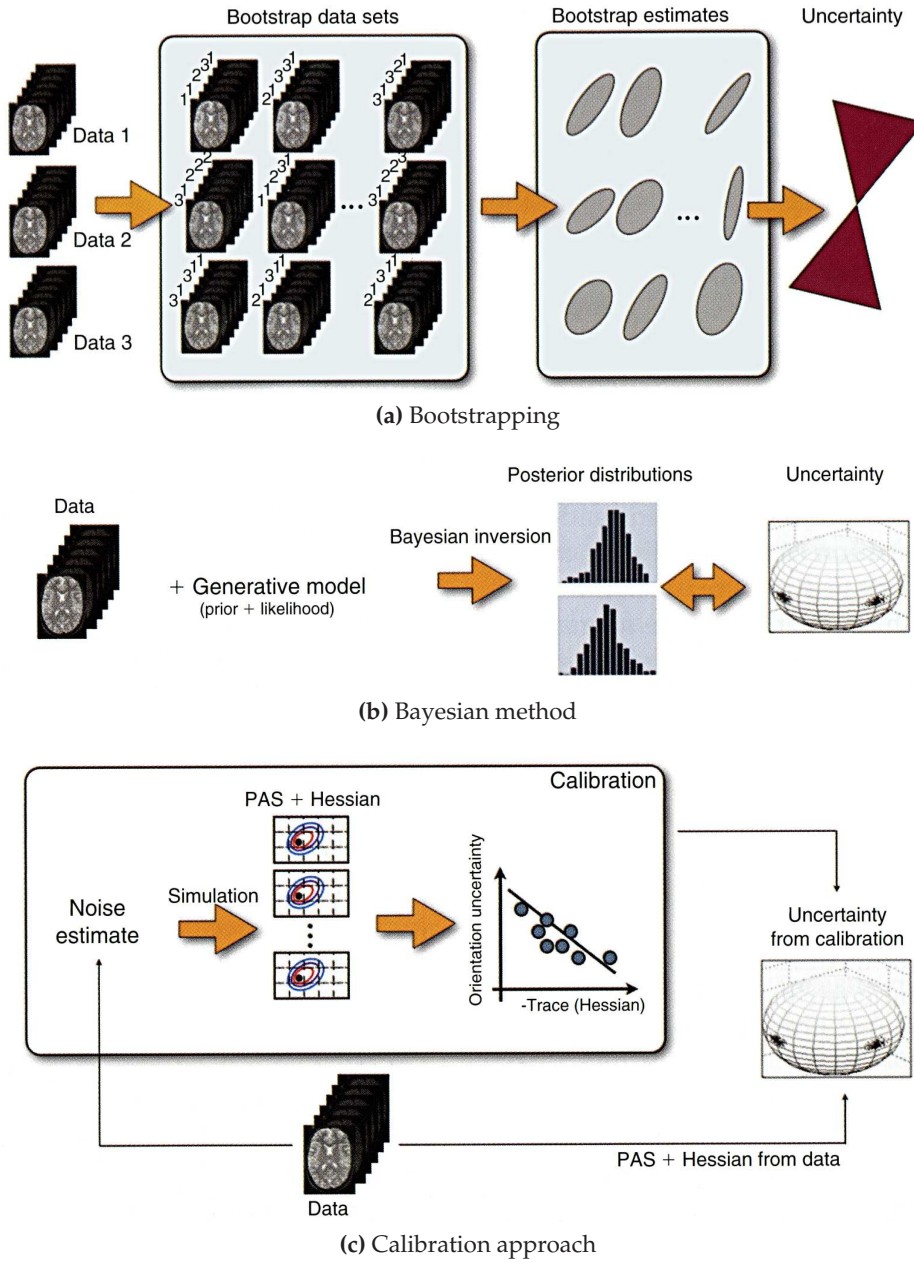


Figure 4.10: Illustration of methods for uncertainty estimation. Image adopted from [JBB09, p. 340].

Bayesian approach offers a wide variety of techniques making it practical for a wider class of problems than bootstrapping.

Functional Approximation and Calibration Approach

The last two approaches on inferring the uODF are applicable in cases where diffusion is represented by some kind of data structure (e.g. DT or dODF). In PAS MRI the dominant diffusion directions are reconstructed without a representation by a specific data structure. Here, an approximate relationship between the shape of the uODF and the sharpness of diffusion directions at different noise levels can be used. Like in DTI, the peaks of the PAS are blurred and not well defined in regions of isotropic diffusion. Therefore, with known curvature of the PAS peak (computed as the trace of the Hessian matrix of the PAS function) and the corresponding image noise the uncertainty in the PAS peaks and, thus, the uODF can be predicted [PA05]. First, the SNR of the available data needs to be determined. After that this data is simulated by a model with known parameters (e.g. a mixture of tensors) with added noise at the measured SNR. For each simulation the PAS is reconstructed and the orientation and curvature of the peaks is calculated. The simulation is then repeated many times with a different set of parameters (e.g. a different angle between fiber orientations). From these simulations an empirical relationship between the curvature of the PAS peaks and the standard deviation of its orientation can be recovered for different noise levels (Figure 4.10c).

4.2.2 Propagation of Uncertainty

In deterministic tractography a streamline would simply follow a maximum diffusion direction to integrate the fiber trajectory. Unfortunately, this is not possible in probabilistic fiber tracing. Here, not only one (or a few) possible directions exist but an infinite amount each with a different probability of being the true fiber orientation. To calculate the probability that two regions of the brain are connected all paths and their probabilities need to be considered. Mathematically, this is an integration problem that can not be solved analytically. However, instead of solving this integral it is possible to draw samples from the computed uODF to generate possible fiber pathways (see examples in the next section). Starting at a seedpoint a sample orientation is drawn from the uODF. The fiber trajectory is then advanced a certain distance along this orientation (Figure 4.11). This procedure is repeated until a stopping criterion is met. The output is one sample streamline. By creating many such streamline from a seed point, P , the entire probability density function from P can be generated.

This sampling procedure yields the same result as the computation of the above mentioned integral provided that enough samples are generated.

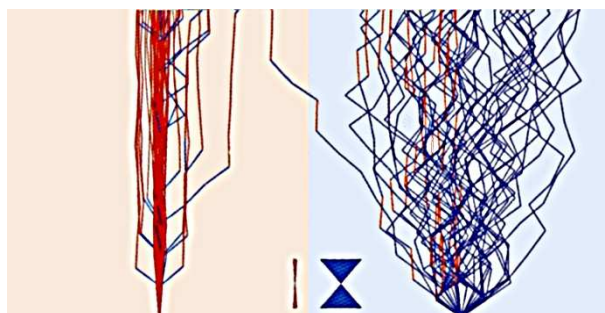


Figure 4.11: Propagation of streamlines with low (red voxel) and high uncertainty (blue voxel). Sample streamlines are started from the bottom of each voxel. Samples are drawn from the uODF and the streamline is advanced one step. The illustrated step size is one tenth of a voxel. Image adopted from [JBB09, p. 342].

In the integration procedure a pathway that is twice as probable as another one would contribute twice as much to the resulting probability. In the sampling approach every pathway is sampled with the likelihood of its probability. Hence, this pathway is twice as likely to be sampled. A different method of uncertainty propagation will be presented in section 4.3.1 when discussing global tractography approaches.

The generated streamlines must not be thought of as multiple connections starting at a single voxel. They are rather possible pathways with different possibilities of a single connection. Therefore, after completing the sampling procedure the number of streamlines passing through each voxel is counted and divided by the total number of streamlines. This way, the continuous probability distribution is turned into discrete probabilities for pathways through different voxels. When using this approach it must be understood that the division into voxels is arbitrary. Making the voxels twice as big also doubles the probability for a pathway to pass through this voxel. Therefore, it is useful to define anatomically meaningful regions and compute the probabilities of pathways passing through them instead. In practice, these two approaches can be combined. By considering probabilities for voxels the pathways can be validated to be anatomically reasonable. By defining probabilities between brain regions quantitative anatomical comparisons can be made.

In probabilistic fiber tracing the stopping criteria can be redefined. As a streamline shall not be stopped when arriving at a region with high uncertainty most of the time lower FA thresholding can be used than is common in deterministic fiber tracing. Curvature constraints are usually applied only to prevent fibers from falling back on themselves. Usually, implausible pathways are filtered out due to their low probabilities and scattered spatial distributions. Therefore, probabilistic streamlines can simply be

stopped when reaching the end of the brain.

The ability to assign probabilities to pathways and filter out unlikely connections makes probabilistic fiber tracing more meaningful and more robust to noise. On the other hand, these tracing approaches demand for a significantly higher computation time as not only one but several hundreds or even thousands of streamlines need to be generated.

4.2.3 Example Algorithms

This section presents example algorithms for the different uncertainty estimation approaches discussed above.

DTI Bootstrap Lazar and Alexander investigated the error propagation along fiber pathways on bootstrapped DTI data [LA05]. Bootstrapping datasets were generated based on 8 independent data volumes each acquired with 12 diffusion directions. For each voxel in the bootstrapped dataset N samples of the corresponding voxel were drawn with replacement from the 8 datasets. These samples were averaged to form one voxel in the resulting data volume. N was constant for one bootstrapping experiment but varied among experiments between 1 and 8. In total, 1000 datasets with different values for N were generated. The deterministic tensorline (tensor deflection) algorithm was used for fiber tracing with second order Runge-Kutta integration. Fiber tracts were stopped based on FA and curvature thresholds. Fiber tracing was performed on all 1000 volumes resulting in 1000 individual streamlines per seedpoint. The probability for the connectivity of two voxels was computed based on the number of fiber trajectories connecting these voxels. Fibers were reconstructed in the corpus callosum matching prior anatomical knowledge. Further, these experiments have shown that fiber tract dispersion depended on the distance to the seedvoxel. The dispersion decreased with higher values for N resulting in more likely pathways. This experiments confirmed that bootstrapping from several independent datasets provides valuable data for fiber tract reconstruction.

DTI Residual Bootstrap The performance of bootstrapping and residual bootstrapping on DTI data was compared and evaluated in [Jon08]. 1000 bootstrapped datasets were generated from 9 original datasets with 30 unique gradient directions. For each diffusion weighted image of the resulting datasets one corresponding image was drawn from the original dataset. Further, 1000 residual bootstrapped datasets were generated from one of the original datasets. Deterministic streamline tracing was performed with RK4 integration on each of the generated datasets. Resulting streamlines for both approaches are presented in Figure 4.12 for a seedpoint in the

4.2. LOCAL PROBABILISTIC TRACTOGRAPHY

internal capsule of the corticospinal tract (indicated by the arrow). In both images a small amount of streamlines (shown in green) erroneously reaches the other brain side. These anatomically implausible ways are mostly represented by low visitation counts (blue streamlines). When displaying only voxels reached by at least 50% of all streamlines (red), both methods yield similar anatomically plausible results. These results suggest that one data acquisition is sufficient to reliably reconstruct fiber pathways when using the residual bootstrapping approach. This simplification allows to acquire the needed data in a clinically appropriate time.

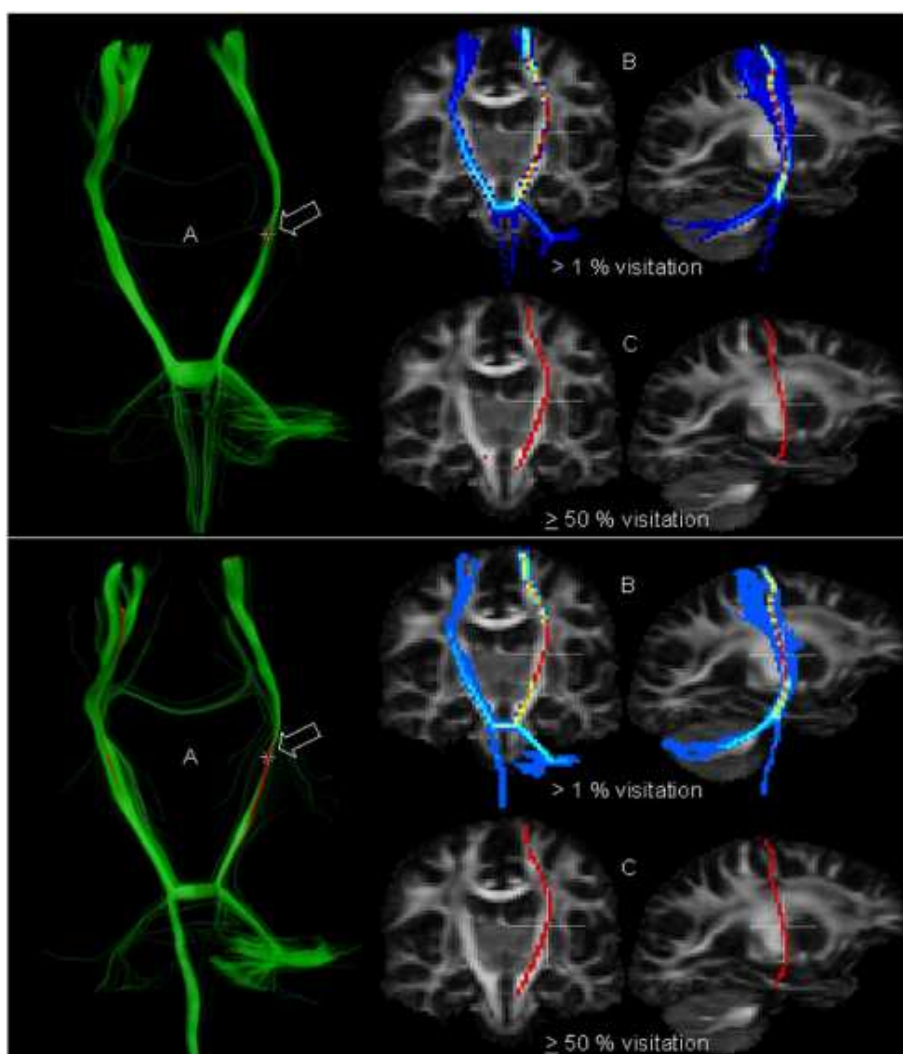


Figure 4.12: Comparison of tracing results of the bootstrap (upper panel) and residual bootstrap (lower panel) methods. Image adopted from [Jon08].

HARDI Residual Bootstrap The residual bootstrap was recently applied to HARDI data with 55 diffusion directions [BCM⁺08]. Q-ball reconstruction was performed as proposed by Hess *et al.* [HMH⁺06]. Parameters used for reconstruction were 642 sampling directions at a SH order of 4. Fiber tracing was started from 64 densely seeded points in each voxel of the user defined ROI. On reaching a new voxel a dODF was reconstructed with the residual bootstrapping approach for each individual streamline. Therefore, different streamlines were propagated by different ODFs when passing through the same voxel. In each voxel the global maximum was identified. Subsequently, all local maxima were found that were not within 45 degrees of the global maximum and had at least the magnitude of 33% of the global maximum. The fiber tract was continued along the direction most similar to the incoming trajectory. FA (0.05) and curvature (75 degrees) thresholds for used for stopping. The number of streamlines passing through each voxel served as a confidence metric in the fiber path. These results were compared with the results from the DTI FACT algorithm in anatomically well known regions of the brain. The Q-ball approach consistently provided better results as it was able to identify multiply fiber populations per voxel and, thus, follow even nondominant fibers.

DTI Independent Scoring Common to all probabilistic algorithms so far is the fact that the process of finding plausible pathways is not separated from the assessment of their validity. In other words: The validity of a pathway is determined by the number of sample streamlines passing through the corresponding voxels. A recent approach, termed *ConTrack*, aims at separating these two steps [SDBS⁺08]. DTI data with 8 repetitions and 12 diffusion encoding directions was used in this approach. Sample pathways were generated by bootstrapping and the application of two ROIs. Subsequently, a score, $Q(s)$, was calculated for each potential pathway, s , as $Q(s) = p(D|s)p(s)$. The first part, $p(D|s)$, is the data dependent score of finding the pathway, s , given the diffusion tensor D . The data dependent score incorporates two dispersion parameters. One accounts for the uncertainty of the data and is derived from bootstrapping. The other parameter handles the uncertainty introduced by the shape of the diffusion ellipsoid.

The second part of the scoring equation, $p(s)$, is the data independent term. It incorporated prior knowledge about the general structure of white matter pathways such as curvature and typical length. This scoring procedure realizes two principles: symmetry and independence (Figure 4.13). With the principle of symmetry the probability for a pathway between two regions, $R1$ and $R2$, is the same in both directions along this pathway. The principle of independence scores a pathway from a certain region independently of other potentially existing pathways starting at the same region. This approach was able to find known pathways that were missed by other

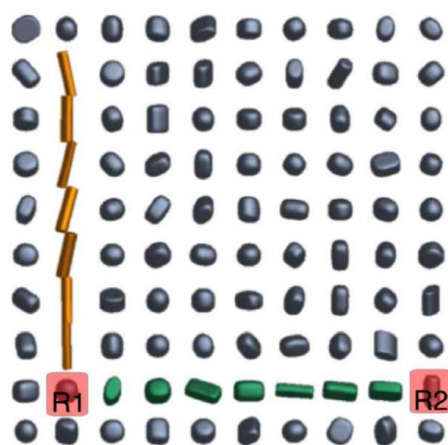


Figure 4.13: Principles of symmetry and independence for path scoring. Image adopted from [SDBS⁺08].

DTI based methods.

HARDI Bayesian Approach A method on reconstructing fiber pathways with the Bayesian approach and the compartment model is presented in [BJB⁺07b]. HARDI data with 60 diffusion encoding gradients and 3 repetitions were acquired. The so called *automatic relevance determination* (ARD) method was used to infer on the number of fiber populations in each voxel. In simple terms, ARD is a method that assigns zero probability in the posterior distribution to parameters that are not supported by the given data. In each voxel an isotropic compartment and a number of anisotropic compartments were modeled (corresponding to the number of fiber populations in that voxel). 5000 sample streamlines were generated by drawing samples from the posterior distribution and advancing the tracing position along the drawn direction. In voxels with multiple fiber populations samples were drawn only from the posterior distribution of the fiber population that corresponded to the least curvature of the incoming fiber. By doing so, nondominant pathways were able to propagate through regions of fiber crossings. Fiber bundles were identified that were previously difficult to trace.

DTI Particle Filter Particle filtering on DTI data has been applied for fiber tracing in [ZHGG09]³. An important difference to simple particle seeding approaches is to takes advantage of the weighting and resampling mechanism of particle filtering. This weighting and resampling is an extension to the Bayesian approach introduced in this chapter. Initially, particles

³Although the authors of this approach call it *global*, fiber trajectories are propagated along local diffusion tensor information.

are placed in the seed voxel. A Bayesian posterior distribution functions is calculated, samples are drawn, and each particle is advanced a constant step size along the corresponding sample direction. This first step is called *prediction*. In the second step, *weighting*, a weight is assigned to each particles representing its importance. These weights incorporate the uncertainty from the posterior distribution. It can include the plausibility of the pathway (e.g. curvature, tensor shape), data noise, and other prior assumptions. In the third step, *selection*, the particles with highest weights are retained whereas particles with low weights are removed. As the number of particles would decrease in each step the posterior distributions of the high weighted particles are sampled more often to generate more particles at the most probable locations. This procedure leads to the advancement of particles only along trajectories with the highest probabilities of representing white matter fiber bundles. However, due to the diffusion tensor model high uncertainties for particle propagation arise at fiber crossings.

HARDI Monte-Carlo Particles Fiber tracing on an interpolated Q-ball field with a Monte-Carlo estimation was proposed in [PPC⁺05]. Human brain data was acquired with 41 diffusion gradient directions. At the beginning, 20 particles are placed in every voxel of a defined ROI. An initial speed along the maximum direction of the voxel is assigned to each particle and it is advanced a certain distances along this direction. At each step the new speed's direction, \mathbf{v}_{n+1} , results from a trade-off between inertia, or last speed's direction, \mathbf{v}_n , and the force from the current dODF, \mathbf{v}_q :

$$\mathbf{v}_{n+1} = \alpha \mathbf{v}_q + (1 - \alpha) \mathbf{v}_n \quad (4.9)$$

The parameter α is the standard deviation of the dODF normalized by its maximum. In isotropic voxels α takes small values favoring the current direction. Whereas in anisotropic voxels high values for α emphasize the dODF's direction. This particular part of the algorithm is similar to the tensorline approach. The direction of the dODF force, \mathbf{v}_q , is selected randomly within a half-cone of the current direction, \mathbf{v}_n . The sampling takes place in a way that the most probable diffusion direction (i.e. the dODF maximum) inside the half-cone is most likely to be assigned to \mathbf{v}_q . This is done by assigning sampling probabilities proportional to the dODF values inside the half-cone. The angle of the half-cone can be varied to regularize the curvature of the fiber trajectories. Tracing is stopped only when the white matter mask is left. Voxels containing trajectories of only a small number of streamlines are considered as erroneous paths.

HARDI Random Walk Particles Along with a deterministic algorithm described above a probabilistic fiber tracing algorithm was presented in

[DDKA09]. The dODF was sampled from 2562 directions uniformly distributed on the sphere and sharpened with an additional SD transform. The resulting sharp dODF field was normalized and interpolated to gain sub-voxel precision. The proposed algorithm is a random walk approach that uses 100,000 particles per seedvoxel. For each transition from a voxel, x , to a different voxel, y , the transitional probability along a specific direction, \mathbf{d}_{xy} , was computed as the product of the dODF values of these directions. Mathematically expressed this probability is

$$P(x, y) = \Psi_x(\mathbf{d}_{xy}) \cdot \Psi_y(\mathbf{d}_{xy}) \quad (4.10)$$

where $\Psi_x(\mathbf{d})$ is the dODF value of voxel x in direction \mathbf{d} . In each voxel transitional probabilities for 120 directions were calculated. Tracing was stopped when a trajectory left white matter (low FA value). As a final step voxels that were reached by less than 100 particles were considered as implausible fiber trajectories.

4.2.4 Summary

As can be seen on the examples presented here probabilistic tracing algorithms offer a wider variety of possibilities to design tracing algorithms than deterministic approaches do. This is due to the many different ways to represent uncertainty and also due to many possibilities the theory of probabilities offers to combine, select, and score the generated sample pathways. However, all these methods come with a significantly higher computation time as often 1000 or more sample streamlines are generated.

Also noticeable is the fact that even several years after HARDI acquisition schemes have become possible in a clinically feasible time new probabilistic algorithms based on the classic DTI model are proposed. This can be attributed to the possibility of probabilistic approaches to reconstruct complicated brain structures by simply sampling a large amount of streamlines. Clearly, HARDI techniques offer more reliability and flexibility in fiber trajectory reconstruction especially in challenging brain regions. Nevertheless, more time is needed until HARDI methods become as common in probabilistic fiber tracing as they already are in deterministic fiber tracing.

4.3 Global Tractography

So far the main assumption of fiber tracing algorithms was that fiber trajectories are represented by paths of highest diffusion. The local algorithms introduced in the previous sections use this assumption by locally following the PDD or a path with high diffusion probability. As typically only the paths of highest diffusivity are chosen for tract propagation nondominant

connections can not be reconstructed. This is a well-known problem in areas of multiple fiber populations. Probabilistic methods partially account for this fact by randomly choosing different probable diffusion directions. Though even with this approaches some fiber tracts are hard or impossible to find.

Global tractography approaches try to solve the fiber reconstruction problem differently. Given two points A and B in the brain the task is to find a path of maximum diffusivity that connects these points. When following the path of highest diffusion locally there might be no path connecting A and B . Or expressed in other words, the path of maximum diffusivity connecting A and B might not be aligned locally with the path of least hindrance to diffusion in each point. However, out of all possible paths leading from A to B the found path is the path of highest diffusivity. One major advantage of global approaches is that local errors do not accumulate along pathways as an entire pathway is considered simultaneously. Thus, it is unlikely that a single erroneous measurement leads to a reconstruction of implausible paths.

The feature of interest in these new approaches must be global, i.e. integrated along the whole fiber tract. Mathematically, this is a global optimization problem that can be expressed as a minimization of a path integral. Two different approaches to solve the fiber tracing problem globally will be introduced in the following sections. Subsequently, example algorithms for both approaches will be presented.

4.3.1 Front Evolution Approaches

In front evolution approaches local diffusivity can be interpreted as local speed. Paths with higher diffusivity are traversed with higher speeds than paths of low diffusivity. Thus, the globally optimal connection between two regions can be thought of as the path with the minimal arrival time. Front evolution approaches are used to build spatially encoded arrival time maps starting at a seed point and leading to any possible position in the dataset. The arrival time map contains the minimal arrival time integrated along the fastest possible path to each point.

The front is the border of the region within which the arrival time is known for each voxel. Initially, the front consists of only one voxel, the seedpoint, with an arrival time of zero. Then, an arrival time is assigned to all adjacent voxels to the front. This newly assigned arrival time depends on the diffusivity properties of each voxel and its respective distance to the closest point in the front. Subsequently, the points with smallest arrival times adjacent to the front are chosen and added to the front. With the modified front all adjacent voxels are updated again. The front evolution proceeds until no further voxels can be included. Globally optimal paths on the arrival time map can be found by following the gradient of the arrival

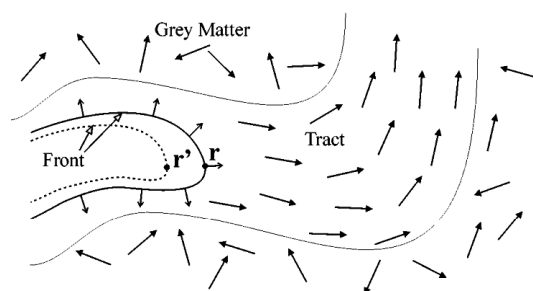


Figure 4.14: Illustration of the front evolution approach. The front evolves from the dotted to the solid line. The propagation is determined by a speed function. In this example, propagation is fastest along the normal vector connecting r' and r . Image adopted from [PWKB02].

times from any point to any other point.

4.3.2 Energy Minimization

Energy minimization approaches typically sample the white matter with a large number of particles. These particles are defined by a position and an orientation. One or more different types of energies is assigned to each of the particle. The energy types describe each particle's possibility to move, rotate, generate new particles, or connect with other particles to form larger segments. The energy of two connected particles is lower than of the same two particles if they are separated. Further, particles that are aligned to a direction of high diffusivity also have less energy than particles aligning to other directions.

Fiber tracing is performed by finding a global configuration of particles with minimal energy, i.e. a configuration where the sum of all energies calculated for all particles is minimal. Usually, the results of these tractography approaches are of high quality and anatomically plausible. However, the process of finding the minimal energy configuration is very time-consuming making energy minimization approaches clinically unfeasible.

4.3.3 Example Algorithms

This section provides several state-of-the-art examples for global tractography. Again an approach will be mentioned that participated in the evaluation of different fiber tracing algorithms at the Fiber Cup 2009.

DTI Front Evolution One of the first global tractography approaches was introduced in [PWKB02]. Two different speed propagation functions were proposed and evaluated. The first speed function is called the *voxel similarity measure*. According to this function the fastest front propagation

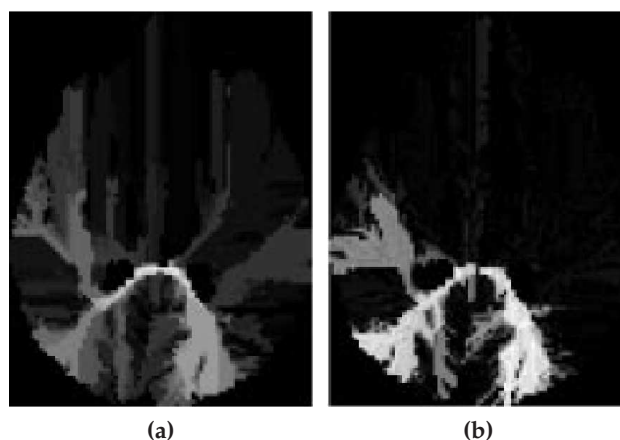


Figure 4.15: Connectivity maps resulting from the front evolution approach. Image adopted from [PWKB02].

occurs in the direction in which the first eigenvectors of the closest adjacent voxels inside and outside the front are most collinear to the normal vector of the front between these two voxels (Figure 4.14). The second speed function favors the voxel connectivity (*voxel connectivity measure*). Here, only the first eigenvector of the closest voxel inside the front and the normal vector of the front are required to be most similar. Thus, the previous diffusion direction is preferred for front propagation. In both cases the normal vector to the front is determined from the 26 neighboring voxels. The front is only propagated to voxels above a certain FA threshold.

Paths of connection were determined on a trilinearly interpolated time of arrival map by using Euler integration with a constant time step size. Two connectivity metrics were used to identify fiber pathways. First, for each of the two speed functions connectivity maps following the time of arrival gradient were generated. The second metric aimed at assessing how faithful a potential fiber path is given the underlying orientations of the first eigenvectors. For this purpose the angle between the tangent to the pathway and the first eigenvector at each point was determined. Smaller angles were considered as higher plausibilities. Experiments showed that the voxel connectivity measure combined with the pathway plausibility metric provides best results (Figure 4.15b). Pathways generated by simply following the gradients resulted in high lateral spreading of fibers and implausible anatomical connections (Figure 4.15a).

Hybrid Front Evolution A global tractography approach combining DTI and HARDI data was proposed in [CSR⁺05]. The method is based on [PWKB02]. In this new approach the whole diffusion tensor is used for front evolution. In cases where DTI does not provide reliable direc-

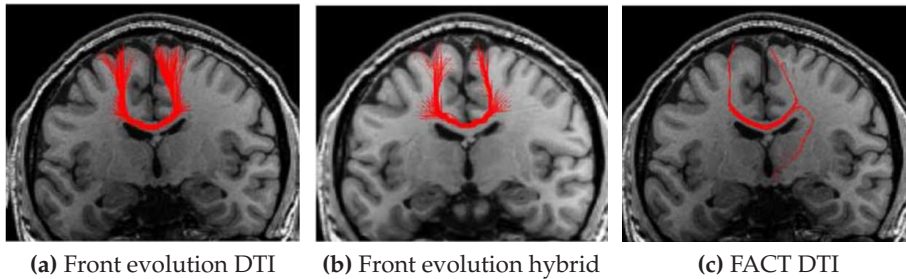


Figure 4.16: Comparison of the tracing results from DTI based front evolution, hybrid front evolution and FACT fiber tracing. Images adopted from [CSR⁺05].

tional information HARDI data can be used instead. The dODF value in the direction of the normal vector of the front is used as a speed function for propagation. In regions with high curvature the dODF values below the mean of the dODF are ignored to avoid wrong paths resulting from streamlines cutting a corner. This thresholding is not necessary in other regions as the laterally spreading trajectories get low connectivity indices. FA and curvature thresholds were used to stop the front evolution in gray matter voxels. 27 sub-voxel seedpoints were generated in every seedvoxel. Fiber trajectories were reconstructed from the time of arrival maps using FACT integration.

Experiments were performed both, on phantom and on human brain data. Front evolution was tested on DTI data, on Q-ball HARDI data, and as a hybrid approach using HARDI data only in regions where DTI directional information was insufficient. Additionally, the streamlines from a classical deterministic PDD algorithms with FACT integration were computed for comparison. The pure Q-ball approach performed less well than the DTI method. The authors of [CSR⁺05] attribute this to the high sensitivity to noise that lies in the Q-ball reconstruction. The hybrid approach performed best as the mixture of the datasets was an ideal trade-off between noise and directional information in challenging regions. The tracing results are presented in Figure 4.16.

Spin Energy Minimization Recently, an energy minimization technique was proposed that can be applied to DTI and HARDI data [FPM09]. In this approach the entire white matter is parameterized by a large number of unit length particles, called *spins*. Each of the spins has three different types of energy: a diffusion, an interaction, and a generative potential. The diffusion potential of a spin is minimized if this spin is rotated parallel to the orientation of maximum diffusivity (either indicated by a diffusion tensor or a dODF). This energy tends towards infinity in gray matter voxels. By minimizing this type of energy fiber segments align to the diffusion

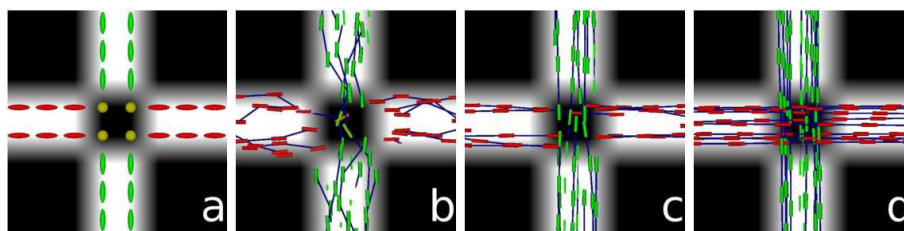


Figure 4.17: The initial tensor field is shown in (a). Spins are placed randomly and align with the PDDs of the voxels (b). Panel (c) represents the minimal energetic configuration of the spins in (b). Panel (d) represents the spin configuration after gap closing. Image adopted from [FPM09].

direction. The interaction potential allows spins to connect to longer segments. It is minimized if both ends of a spin connect to other particles with the lowest possible curvature. The third energy, the generative potential, ensures that fibers do not end inside white matter. If both ends of a spin are connected this energy is zero. However, if at least one end can not be connected to another spin this energy tends towards infinity. Further, it is maximized if a connection is possible but only with a curvature above the specified threshold. In the case of an infinite generative potential new spins are created to close the gaps in white matter. An example illustration on a phantom crossing is presented in Figure 4.17.

The optimal configuration is found when the sum over all spins of the three energies is minimal. The first two energies make conflicting demands on the spins. Hence, a trade-off between curvature and the optimal diffusion alignment needs to be found. These two energies are the first that are minimized. In the last step gaps in the fibers are closed by minimizing the third potential.

This approach was tested on a phantom representing a 45 degree fiber crossing. The results were compared to a DTI streamline algorithm and to the Q-ball approach described earlier in this chapter ([DDKA09]). The results are shown in Figure 4.18. DTI was not able to resolve the fiber crossing due to missing directional information. Some of the streamlines from Q-ball fiber tracing were able to propagate through the crossing although in some cases only one of two expected maxima was identified (green circles in Figure 4.18). The spin model achieved best results on the phantom data.

Further, full brain tractography on human brain data was performed to compare these three fiber tracing algorithms (Figure 4.19). Again, the spin model achieved the best results. However, 2 million spins and 3 days of computation time were needed to fully reconstruct the fiber trajectories.

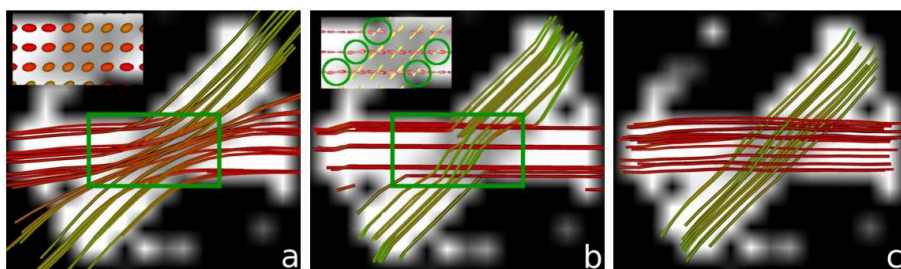


Figure 4.18: Comparison of tracing results of DTI streamlines, Q-ball and the spin method (from left to right). Image adopted from [FPM09].

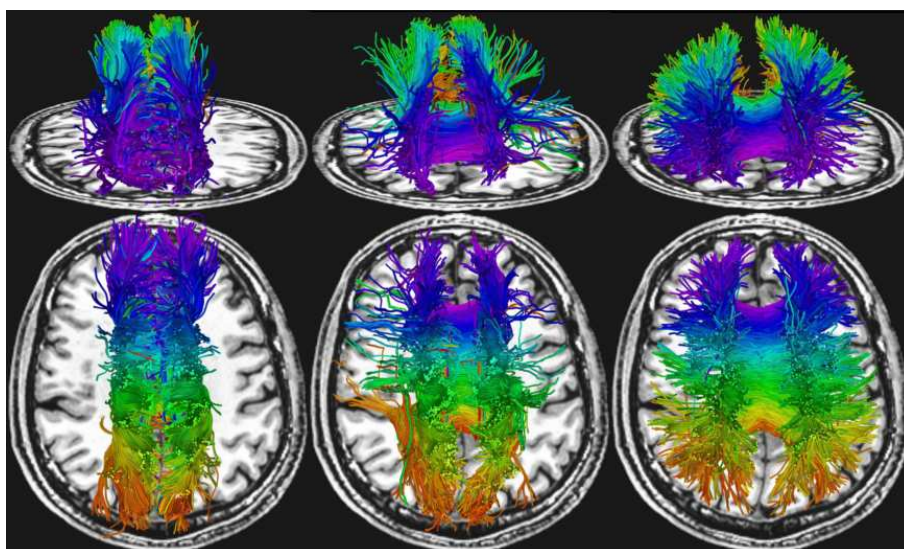


Figure 4.19: Full brain tractography results. From left to right: DTI, Q-ball, and the spin method. Image adopted from [FPM09].

Local Energy Minimization The 1st place at the Fiber Cup 2009 was won with the approach presented in [RMK09]. The key idea of this approach is to use local methods to achieve a globally optimal result. The Bayesian theorem is used to find the most likely model given the observed data. The model is represented by particles with a position, orientation, and an internal energy. The internal energy represents the interaction potential of fiber segments. It keeps connected particles together and minimizes the curvature of the fibers. Thus, only connected segments possess internal energy. Further, an external energy is calculated as the squared difference between the measured signal and the signal predicted by the model.

The so called *Metropolis-Hastings sampler* is used to sample the posterior distribution of the fiber configuration. This configuration is used as a proposed modification of the current model and is accepted with a certain probability. The proposal can be of several types: It can generate new

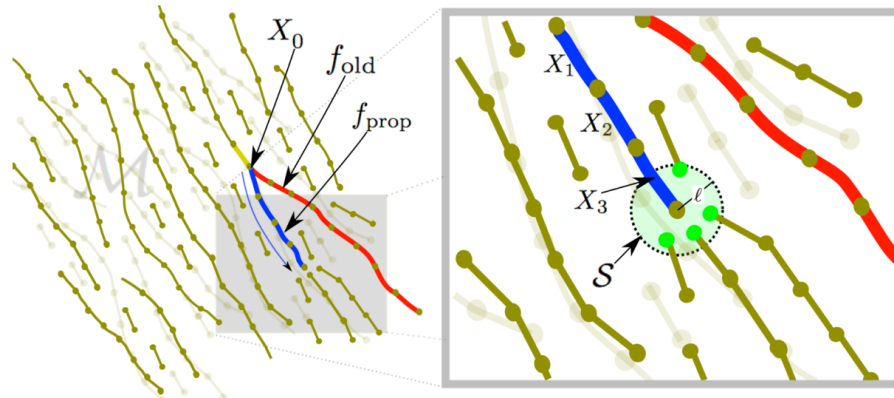


Figure 4.20: Process of segment selection to modify a connection. Image adopted from [RMK09].

particles or remove already existing ones. Further, proposals can randomly move particles, or move particles in a way to minimize their internal energy. The last type of proposal is used for fiber tracing. It connects fiber segments or modifies already existing connections. An example for the latter is given in Figure 4.20. A segment, X_0 , is selected for modification by the algorithm: The red fiber (f_{old}) is replaced by the blue one (f_{prop}). During fiber tracing the new segment searches for successors in a small radius indicated by the circle. Free endpoints in a small radius, l , (marked in green) form the set of possible successors S .

This approach was tested on human brain data and successfully reconstructed known anatomical fibers. The computation time for a whole brain tractography was below one day. As this global approach does not work with seedpoints it had to be adjusted for the Fiber Cup. Most probable fiber trajectories passing through seedpoints were selected and averaged to obtain a representative fiber tract for each seedpoint.

4.4 Summary

This chapter concludes the theoretical part of this thesis. In the preceding chapters background knowledge on MRI and diffusion MRI was provided. Further, different reconstruction methods for HARDI data were introduced. This chapter extensively treated current fiber tracing approaches. These were arranged in three groups: deterministic, probabilistic, and global methods.

Deterministic approaches have the least demands on computation time. DTI based methods of this group follow the principal diffusion direction in each voxel (streamline tracing) or use the whole diffusion tensor (tensor-line tracing) to reconstruct neuronal fibers. HARDI based algorithms of-

4.4. SUMMARY

ten select one or several local diffusion maxima to continue the streamline. Variations of the current approaches are limited to the maxima selection and different weightings of possible directions. Deterministic approaches suffer from the fact that integration errors accumulate along the fiber tract and can lead to great deviations from the true underlying fiber trajectory. Further, no reliability information on the reconstructed pathways can be inferred from this kind of approaches.

Probabilistic methods, however, indicate whether a fiber trajectory is probable or not. The idea of these approaches is to represent the uncertainty of the diffusion direction in a probability distribution. A streamline is continued by drawing samples from this distribution. Often, several hundreds of streamlines are generated and the most likely fiber pathways are extracted from the results. In contrast to deterministic algorithms new DTI based probabilistic methods are still developed. This is due to the fact that probabilistic approaches can better handle uncertainty and insufficient directional information. Nevertheless, this approaches can benefit from HARDI data.

Global approaches have the highest demands on computation time. Most of these algorithms are not suitable for clinical purposes. The idea behind global fiber tracing is to find a globally optimal path of least hindrance to diffusion instead of following locally the principal diffusion direction. This results often in a complicated global optimization problem. The results obtained with global approaches are better than with any other type of algorithm. However, the long computation times are a serious drawback making these algorithms not applicable in practice.

Based on the ideas arising from the various fiber tracing approaches presented in this chapter new algorithms will be developed and evaluated in part III of this thesis.

Part III

Practical Applications

Chapter 5

Implementation

The implementation for this thesis is described in this chapter. Section 5.1 gives a very brief introduction to MeVisLab that is used as a framework for implementation. Subsequently, section 5.2 provides an overview of the classes the implemented modules are derived from. Important properties, functions, and data types are introduced and explained. Further, the implemented image processing pipeline is explained. Finally, a detailed description of all implemented modules and algorithms is given in section 5.3.

5.1 Development Environment

The software for this thesis was developed on *Ubuntu* 10.04¹, a Linux distribution. The code was written in C++ using the *Qt Creator*² integrated development environment (IDE). The developed software is not a stand-alone application and, thus, can not be executed per se. Rather, the software consists of several modules that can be loaded and executed in *MeVisLab*³, a powerful development environment for medical image processing, provided by MeVis Medical Solutions AG, Bremen, Germany. A very short introduction to MeVisLab is given in the following section to provide a basic understanding of this framework. For further information please refer to the MeVisLab documentation web pages⁴. Further, the *Boost*⁵ library was used for computation of SH basis functions and the *Newmat*⁶ matrix library for general matrix and vector calculations.

¹<http://www.ubuntu.com>

²<http://qt.nokia.com>

³<http://www.mevislab.de>

⁴<http://www.mevislab.de/fileadmin/docs/current/MeVisLab/Resources/Documentation/Publish/SDK/GettingStarted>

⁵<http://www.boost.org>

⁶<http://www.robertnz.net>

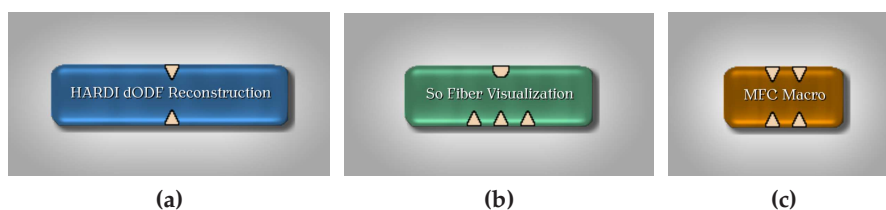


Figure 5.1: The different module types of MeVisLab. (a) shows an ML module for image processing, (b) an Open Inventor visualization module. The macro module shown in (c) subsumes a whole MeVisLab network.

5.1.1 MeVisLab

MeVisLab is a modular development platform for medical image processing and visualization. It allows for rapid prototyping and offers easy ways of combining modules to algorithm pipelines. In MeVisLab a combination of modules is called *network*. This combination of modules to new algorithms and networks can be achieved conveniently in a graphical editor simply by connecting (i.e. drawing lines) inputs and outputs of the desired modules. Currently, MeVisLab includes far more than 1000 image processing and visualization modules. The visualization modules are based on the *Open Inventor* toolkit and the graphics standard *Open GL*. By default, all visualization modules have the name prefix *So*. Further, an additional project specific prefix can be assigned to modules. For all modules implemented in this thesis the prefix is *UKO* for *University of Koblenz*. However, this additional prefix is seen only in the source code of the modules and not in MeVisLab.

There are three different types of modules used in MeVisLab:

1. ML modules for image processing (blue, Figure 5.1a)
2. Open Inventor modules for visualization (green, Figure 5.1b)
3. Macro modules containing other modules or networks (brown, Figure 5.1c)

Each module's interface is represented by its connectors. The connectors on the bottom side are used for input data while the connectors on the upper side represent the module's outputs. For example, the module in Figure 5.1b has three input connectors and one output connector. The shape of the connectors indicate the required data type. Triangles represent MeVisLab image processing Library (ML) images. ML images are six-dimensional data arrays providing enough flexibility even for large medical datasets. The half-circles stand for Open Inventor scenes. A third type, the square, stands for pointers to data structures. The last connection type was not used in this implementation.

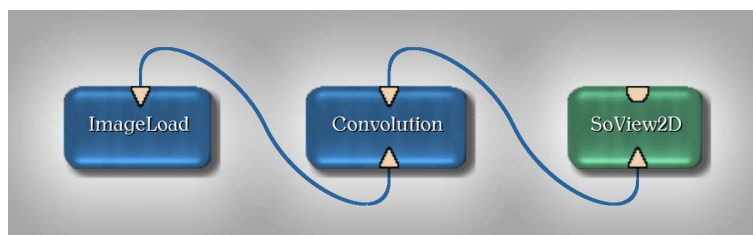


Figure 5.2: A simple example network. The dataflow is represented by the smoothly curved lines and starts in the *ImageLoad*. The loaded image is passed to the *Convolution* module where it is processed and subsequently passed to the *SoView2D* module for visualization.

A simple example network is presented in Figure 5.2. The data flow is symbolized by the smoothly curved lines connecting the modules' interfaces. In the module *ImageLoad* an input image is loaded. This image serves as input to the *Convolution* module where it is processed. The output is passed to the *SoView2D* module for visualization.

5.2 Implementation Overview

This section has the purpose to introduce *ml::Module* and *SoShape* as the base classes for the implemented modules. An overview is given on their structure and common features used in the implementation of all derived modules. At the end of this section the implemented MeVisLab network and its dataflow is explained.

5.2.1 ML Modules

The ML modules implemented for this thesis are derived from the base class *ml::Module* which is provided by the MeVisLab library. The class *ml::Module* is shown in Figure 5.3. There are three methods that are of particular interest in the design of new modules that need to be overloaded:

- **handleNotification:** This method is called whenever data of a field in the module is modified. Input images are also considered as fields. A pointer to the modified field is passed to the function so that the modified data can be handled here.
- **calculateOutputImageProperties:** In this method, the properties (e.g. the image size, data type) of the output image at the index that is passed to this function can be calculated and set. By default, the properties of the input image are copied to the output image.
- **calculateOutputSubImage:** The data for the output image is calculated and copied to the output in this function. Usually, this is the

5.2. IMPLEMENTATION OVERVIEW

place where the image processing functions are called from. This function is automatically called several times as by default only a subimage is calculated each time. If all data is calculated at once this function is still called several times to copy all subimage data to the output.

Further, important functions are *getNumInputImages* and *getInputImage* to get the number of input images and the input image at the specified index respectively. Analog functions are provided for the output images. The images are returned as pointers to *PagedImage* allowing only page specific access to the image data.

When not stated differently in section 5.3 the three above emphasized functions are used in the following way throughout the implementation for this thesis: The function *handleNotification* handles all module field changes⁷. If the changed module field is an input image its content is copied and stored in a member variable of type pointer to *VirtualVolume*. This type allows for a random access to image data in contrast to the *PagedImage* data type. After handling a module field change the function *touchOutputImageFields* is called to recalculate the output images (by first calling *calculateOutputImageProperties* and then *calculateOutputSubImage*).

In the function *calculateOutputImageProperties* a pointer to *VirtualVolume* for each output image is created. The data type of the individual image elements is set to the same type as in the input image (usually *float*). The size of the output image's first three dimensions (x, y, z) corresponds to the number of voxels in the input images. The size of the dimensions is copied to the output image. The size of dimension t is adjusted to the number of values that need to be stored for each voxel. This number depends on the individual task of each module.

Finally, the function *calculateOutputSubImage* is used to calculate the data for the output image. Unlike the function's name suggests this is done for the whole output image at once (by conveniently accessing the data in the *VirtualVolume** member variables). Therefore, the *SubImage** parameters for the input image and output image that are passed to the function are ignored. To ensure that the image calculation is not performed various times a flag is set after successfully calculating the output image. This flag is unset on module field changes. For a better overview in the control flow diagrams of each ML module these subsequent calls to this function will not be indicated. Instead, they will be marked by a gray diamond node. After calculating the image data the data is copied to the module's output image.

⁷Note that there is a difference between conventional field variables in C++ classes and variables of type *Field** used in ML modules to pass data to the module (e.g. via a GUI panel or input connectors). For a clear differentiation the former will be referred to as *member variable* whereas the latter will be referred to as *module field*.

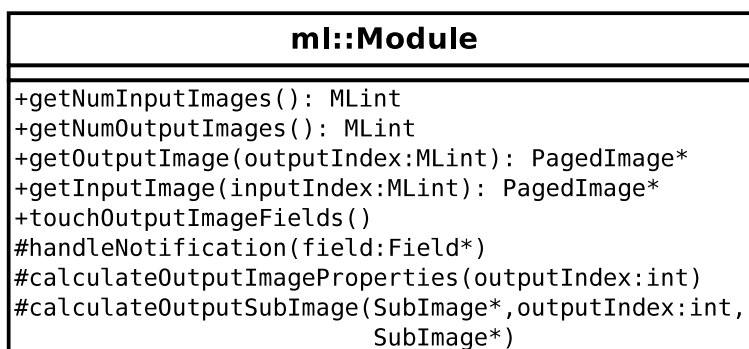


Figure 5.3: The `ml::Module` class with its most important functions. All ML modules implemented for this thesis are derived from this class.

5.2.2 Open Inventor Modules

Open Inventor provides a programming model for visualization which consists of various nodes. Each node holds different objects (e.g. shapes and transformations) that can be combined to represent a scene graph. In MeVis-Lab, the Open Inventor nodes are represented by *SoNode* modules. Scenes for visualization can be constructed by combining modules and adjusting their respective module fields.

Since the main focus of this thesis lies on image processing no sophisticated scene graphs were built for visualization purposes. Rather, the visualization modules for this thesis concentrate on displaying the reconstructed data in a user-friendly and well-arranged way. As their purpose is to render rather simple geometry from the reconstructed data the implemented Open Inventor modules are derived from the base class *SoShape* that represents geometrical objects. The class *SoShape* is a direct subclass of *SoNode*. The class *SoShape* is shown in Figure 5.4.

Two of these functions were reimplemented in the visualization modules for this thesis:

- **GLRender:** This function is called to render the scene using OpenGL. The whole geometry of the scene is rendered here or in other functions that are called here.
- **ComputeBBox:** In this function the dimensions of the bounding box for the rendered scene are set and its center position specified. The bounding box is calculated to be big enough so that the whole scene can be rotated about any axis without clipping.

Open Inventor modules do not provide a *handleNotification* function as do ML modules. Instead, callback functions for every module field need to be specified. On a module field change the corresponding callback function

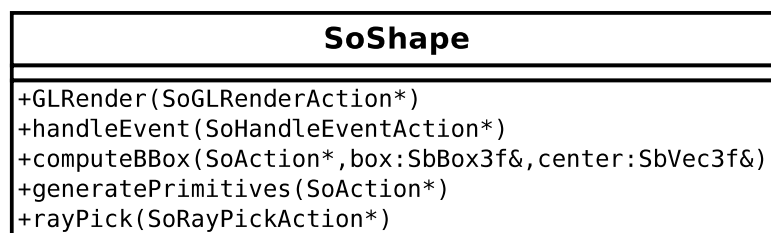


Figure 5.4: The SoShape class with its most important functions. All Open Inventor modules implemented for this thesis are derived from this class.

is executed and a function to handle the module field change can be called from within the callback function.

5.2.3 Implemented Network

Six modules have been implemented in the course of this thesis. Their individual purposes are briefly described here. A detailed description is given in the following section. Appendix A provides a user-level documentation of these modules.

Implemented ML modules:

1. **HARDI_dODF_Reconstruction:** Reconstructs the diffusion ODF from given SH coefficients.
2. **HARDI_AnisotropyCriteria:** Applies different anisotropy criteria to classify the number of fiber populations in a voxel.
3. **HARDI_deterministic_Tractography:** Provides different deterministic fiber tracing algorithms to reconstruct white matter fibers.

Implemented Open Inventor modules:

4. **So_dODF_Visualization:** Visualizes diffusion ODFs in different ways. Optionally, the fiber classification image and the B_0 -image can be displayed in the background.
5. **So_Fiber_Visualization:** Visualizes reconstructed fibers in the dataset. Optionally, the fiber classification image and the B_0 -image can be displayed in the background.

Implemented macro module:

6. **MFC_Macro:** Based on the *HARDI_AnisotropyCriteria* module this module provides a novel global fiber classification approach to determine the number of fiber populations per voxel. This novel approach is based on morphological operations and thus called *morphological fiber classification* (MFC).

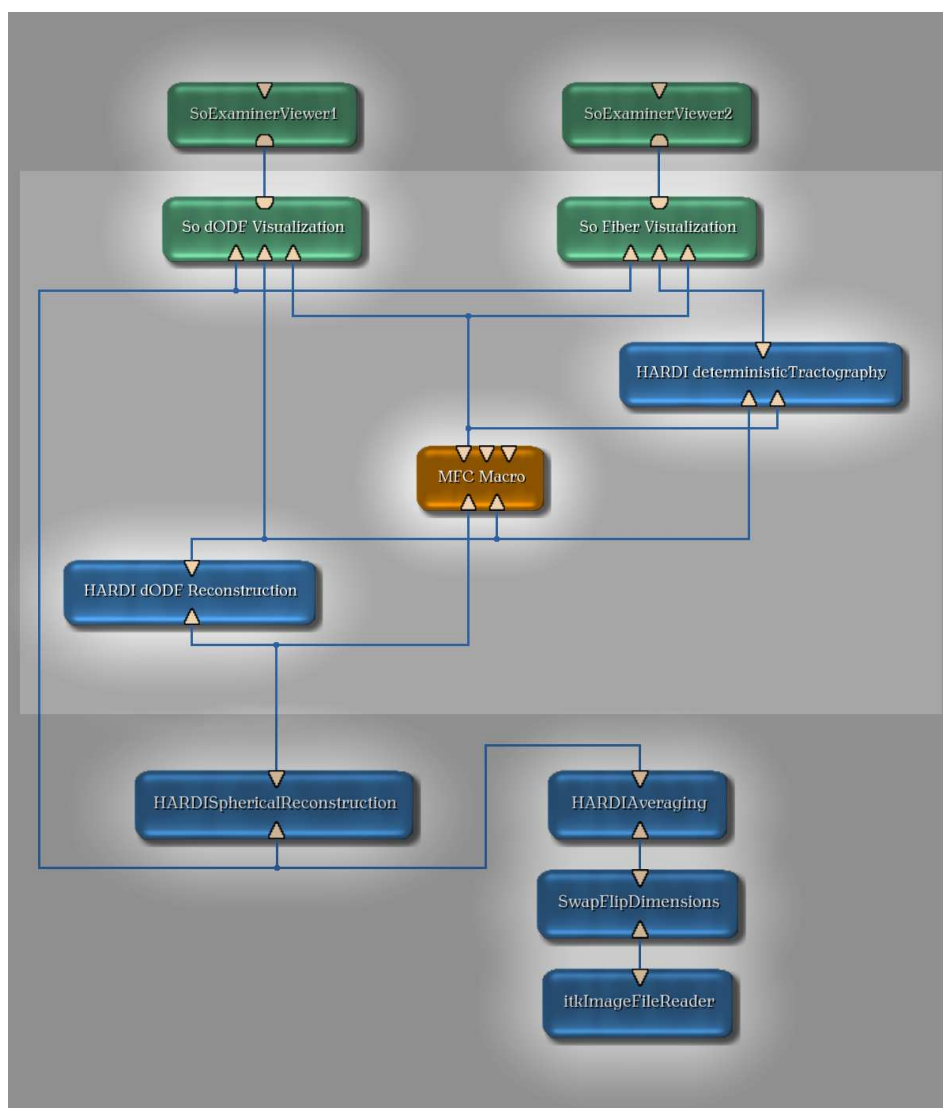


Figure 5.5: Image processing network for white matter fiber reconstruction that has been developed. The input image is loaded by the module *itkImageFileReader* and subsequently processed by all modules in the pipeline. The results are displayed by the two Open Inventor modules on top of the network.

A network, *HARDI_Tractography.mlab*, containing these modules has been created to be used as an image processing pipeline. It is shown in Figure 5.5⁸. The highlighted region contains the implemented modules. Modules outside of this region belong to the MeVisLab library or were

⁸The smoothly curved lines that normally connect the modules were replaced by orthogonal lines to improve the overview.

5.2. IMPLEMENTATION OVERVIEW

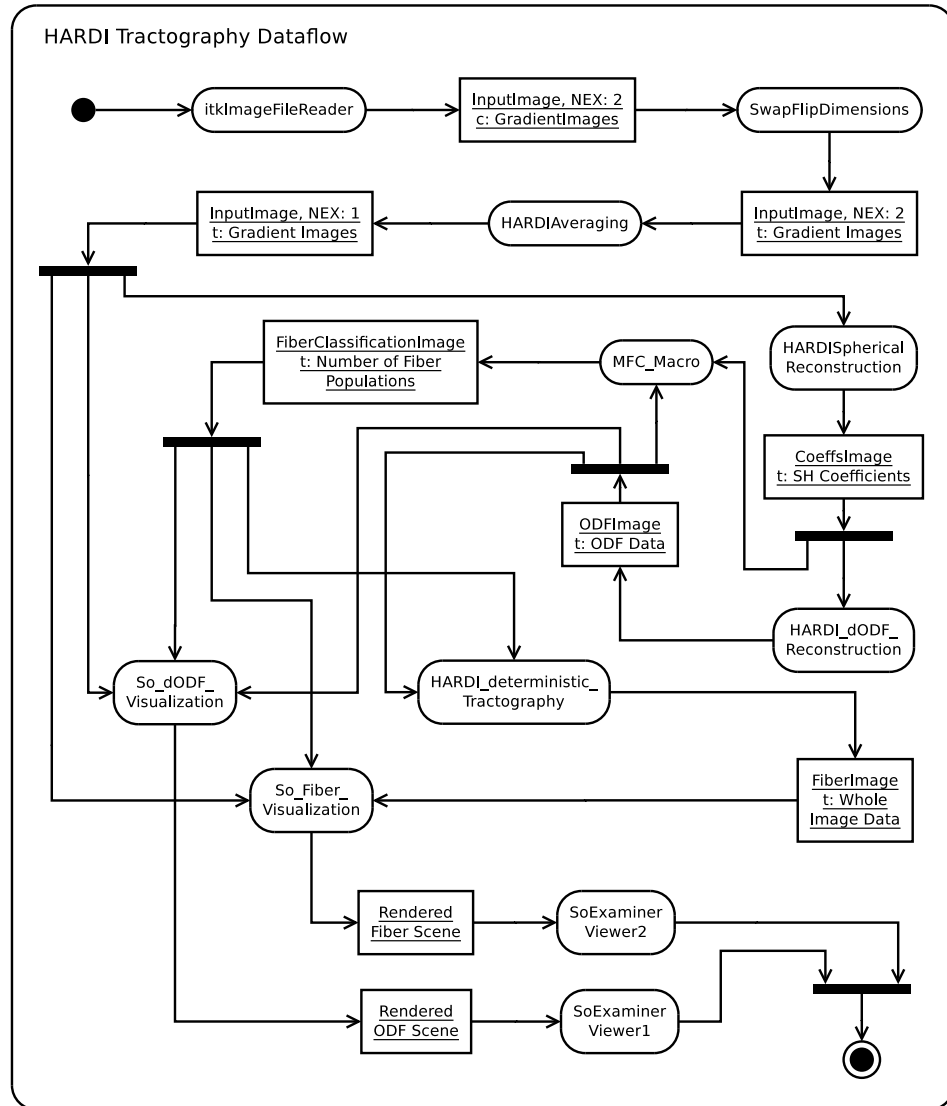


Figure 5.6: The dataflow of the main network for HARDI tractography. Each activity provides a modification of its input image as output image. The dimensions of each image are represented by the x , y , and z dimension. Specific image data is stored in the t dimension for each voxel. Exceptions are the initially loaded image (c dimension) and the fiber image (different image structure).

provided by the Computer Graphics working group of the University of Koblenz. The module *HARDI_AnisotropyCriteria* is not shown in the main network as it forms a part of the inner network of the *MFC_Macro* module. This inner network will be described in detail later in this chapter.

The network shown in Figure 5.5 can be considered as an “applica-

tion” developed during this thesis. Since the modules interact with each other by passing images an UML class diagram of the whole system would not properly depict the relationships of the modules. Instead, an activity diagram with data flow visualization will be used to describe the resulting network (Figure 5.6). Please compare Figures 5.5 and 5.6 while reading this paragraph. The control flow in Figure 5.6 starts in the *itkImageFileReader* module. An input image containing two HARDI acquisitions ($NEX = 2$) of phantom data is loaded (for more details on the dataset see chapter 6). Each acquisition contains an image without diffusion weighting and 64 diffusion weighted images resulting in total in 130 images in the whole loaded dataset. Initially, the diffusion information of the input image is stored in dimension c . Since lots of MeVisLab visualization modules consider the data of the c dimension as color information the data is moved to dimension t in the *SwapFlipDimensions* module. Further, the two acquisitions are averaged in the *HARDIAveraging* module resulting in total in 65 images stored in dimension t . This resulting image is passed to two visualization modules (see below for details) and to the *HARDISphericalReconstruction* module. Using the reconstruction method described in 3.2.3 this module reconstructs the SH coefficients. The t dimension of its output contains the SH coefficients for each voxel (e.g. 15 for the reconstruction with the SH order 4). The *HARDI_dODF_Reconstruction* module uses this image to obtain the diffusion ODF for each voxel. The data is stored again in dimension t (e.g. 162 values for each voxel with tessellation order 3). Out of the coefficients image and the dODF image the *MFC_Macro* module reconstructs the fiber classification image. The dimension t of this image contains only one value for each voxel containing the estimated number of white matter fiber populations. Currently, 0, 1, or 2 fiber populations per voxel are indicated. The unused output pins of the *MFC_Macro* module in Figure 5.5 provide an alternative fiber classification image only indicating whether isotropic or anisotropic diffusion is present in a voxel (one of the inputs provides a median filtered version of this image). The *HARDI_deterministic_Tractography* module reconstructs fibers from the ODF image and the fiber classification image. Here, the whole fiber image data is stored in the t dimension in contrast to other modules where certain values for each voxel were stored in the t dimension. After all required input images for the *So_dODF_Visualization* and the *So_Fiber_Visualization* module are calculated the ODF scene and the fiber scene is rendered and passed to the connected *SoExaminerViewer* module for visualization.

5.3 Detailed Module Description

This section provides a detailed description of the functionality of the implemented modules. A class diagram and a control flow diagram will be presented for each module. In special cases algorithm listings will summarize specific functionalities. Further, the input and output images as well as the parameters accessible via the GUI-panel will be described in full detail. Example images will be provided to illustrate important details.

5.3.1 HARDI_dODF_Reconstruction

The structure of the *HARDI_dODF_Reconstruction* module is illustrated in Figure 5.7. This module is a subclass of the class *ml::Module* and uses the class *SpherePoints*. The purpose of the *HARDI_dODF_Reconstruction* module is to reconstruct the diffusion ODF from SH coefficients for every voxel in the input image. The reconstruction is performed in the function *reconstructODF*. The alternative reconstruction function *reconstructODF2* has higher resemblance to the original reconstruction equation 3.15 and is, thus, more comprehensive. The mathematical details of this reconstruction process are described in section 3.2.3 and in [DAFD07]. SH basis functions for reconstruction can be obtained from the *getSphericalHarmonic* function or from the corresponding function of the *Boost* library.

The number of voxels in the input image has to be indicated by the extents of the *x*, *y*, and *z* dimension while dimension *t* has to carry the SH coefficients for every voxel. The extent of the dimensions *c* and *u* has to

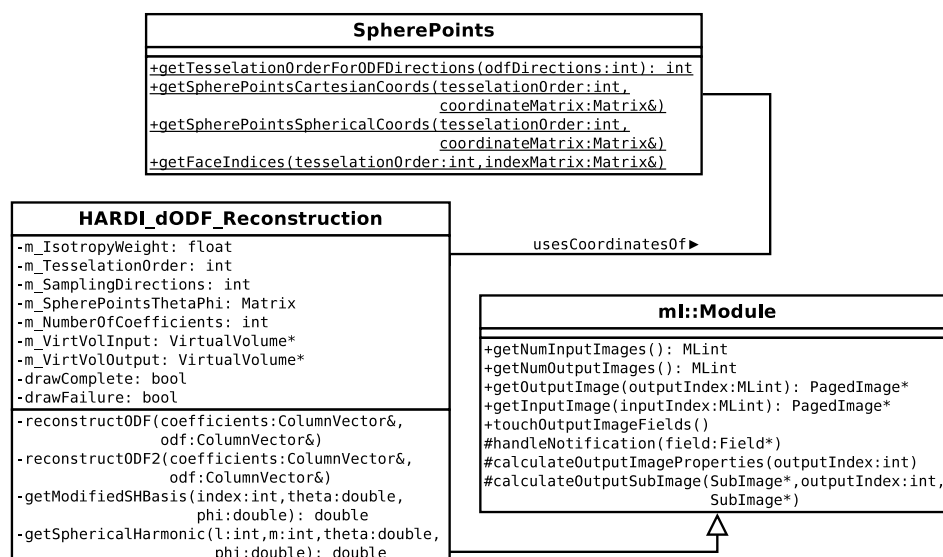


Figure 5.7: Class diagram of the *HARDI_dODF_Reconstruction* module

5.3. DETAILED MODULE DESCRIPTION



Figure 5.8: GUI-panel of the *HARDI_dODF_Reconstruction* module

Tessellation order	Sampling directions
1	12
2	42
3	162
4	642
5	2562

Table 5.1: The number of sampling directions belonging to each tessellation order of an icosahedron.

be 1. The output image contains the reconstructed dODF values in the t dimension for every voxel. The extent of the other dimensions is the same as in the input image.

Apart from SH coefficients uniformly distributed points on the surface of a sphere are needed to serve as sampling directions for the reconstruction of the dODF. These points were generated with *Blender*⁹ and stored in Cartesian coordinates in the class *SpherePoints*. They are derived from different tessellation orders of an icosahedron and form points on the surface of a sphere with radius 1. The center of this sphere lies in the origin. Thus, the coordinates of every point also form a vector from the origin to the surface of the sphere at this point and can be considered as direction vectors. For sampling the Cartesian coordinates are converted to spherical coordinates θ and ϕ , with θ as the polar angle and ϕ as the azimuthal angle.

The tessellation order that is applied in the dODF reconstruction process can be chosen via a parameter in the GUI-panel (Figure 5.8). The supported tessellation orders together with the corresponding number of sampling directions are shown in table 5.1. Choosing higher orders increases the time needed for reconstruction but also increases the directional precision of the reconstructed dODF as is shown in Figure 5.9. The principal diffusion direction (PDD) is poorly defined when using only 12 sampling directions of the tessellation order 1 (Figure 5.9a). In fiber tracing algorithms this leads to reconstruction of erroneous fiber pathways. The orders 3 through 5 show only little differences in the PDD. The proper balance between computation time and precision, therefore, can be found in the orders 3 or 4.

The second module field that can be accessed via the GUI-panel is the

⁹<http://www.blender.org/>

5.3. DETAILED MODULE DESCRIPTION

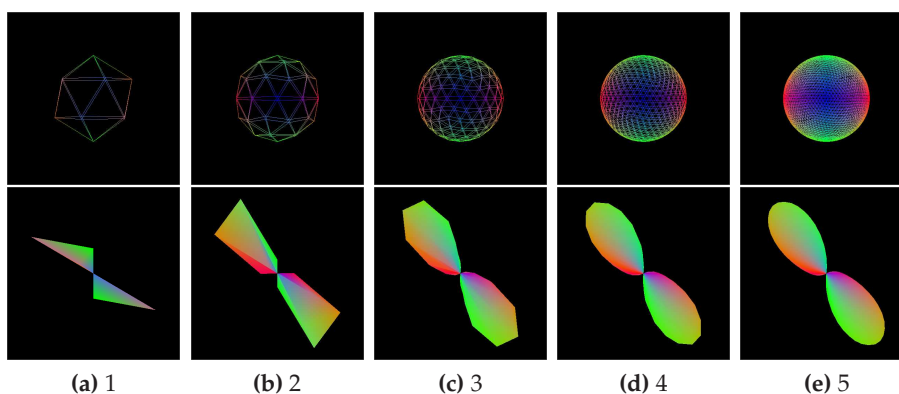


Figure 5.9: The top row shows icosahedra tessellated with different orders. The bottom row shows a corresponding glyph. Each column represents one tessellation order. The effect of choosing higher tessellation orders on the precision of the dODF and, thus, the principal diffusion direction can be seen here. Notice the difference of the principal dODF direction between the order 1 shown in Figure (a) and order 5 shown in Figure (e). Images were created with the *So_dODF_Visualization* module.

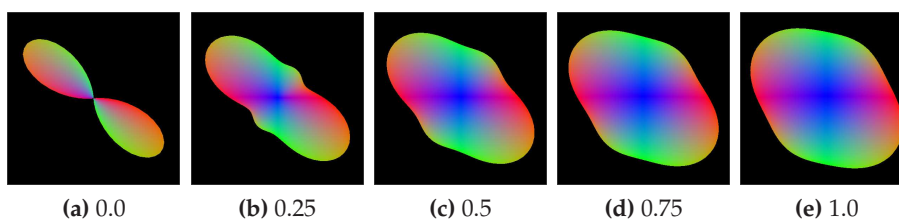


Figure 5.10: The effect of different weights on the isotropic SH coefficient is shown. While the directional information is almost completely lost when using the weighting factor 1 as is shown in Figure (e), it is clearly defined when the isotropic SH coefficient is ignored (Figure (a)). Images were created with the *So_dODF_Visualization* module.

isotropy weight. This number represents an additional factor for the SH coefficient of order 0 (isotropic diffusion). Since isotropic diffusion is predominant (especially in datasets acquired with a low b-value) a lot of the directional information can be lost in dODF reconstruction if the SH coefficient is not adjusted. Using a weight of 0 preserves the directional information of the higher order SH coefficients and allows for more precision in the reconstruction of fiber pathways (Figure 5.10).

The control flow of the dODF reconstruction is illustrated in Figure 5.11. On startup the coordinates for the sampling directions are obtained and the module waits for a module field change. Usually the first field change is the provided input image. Subsequently, the flags *drawComplete* and *drawFailure* are unset as a new output image has to be generated. The input image is copied to the corresponding *VirtualVolume** member variable and

5.3. DETAILED MODULE DESCRIPTION

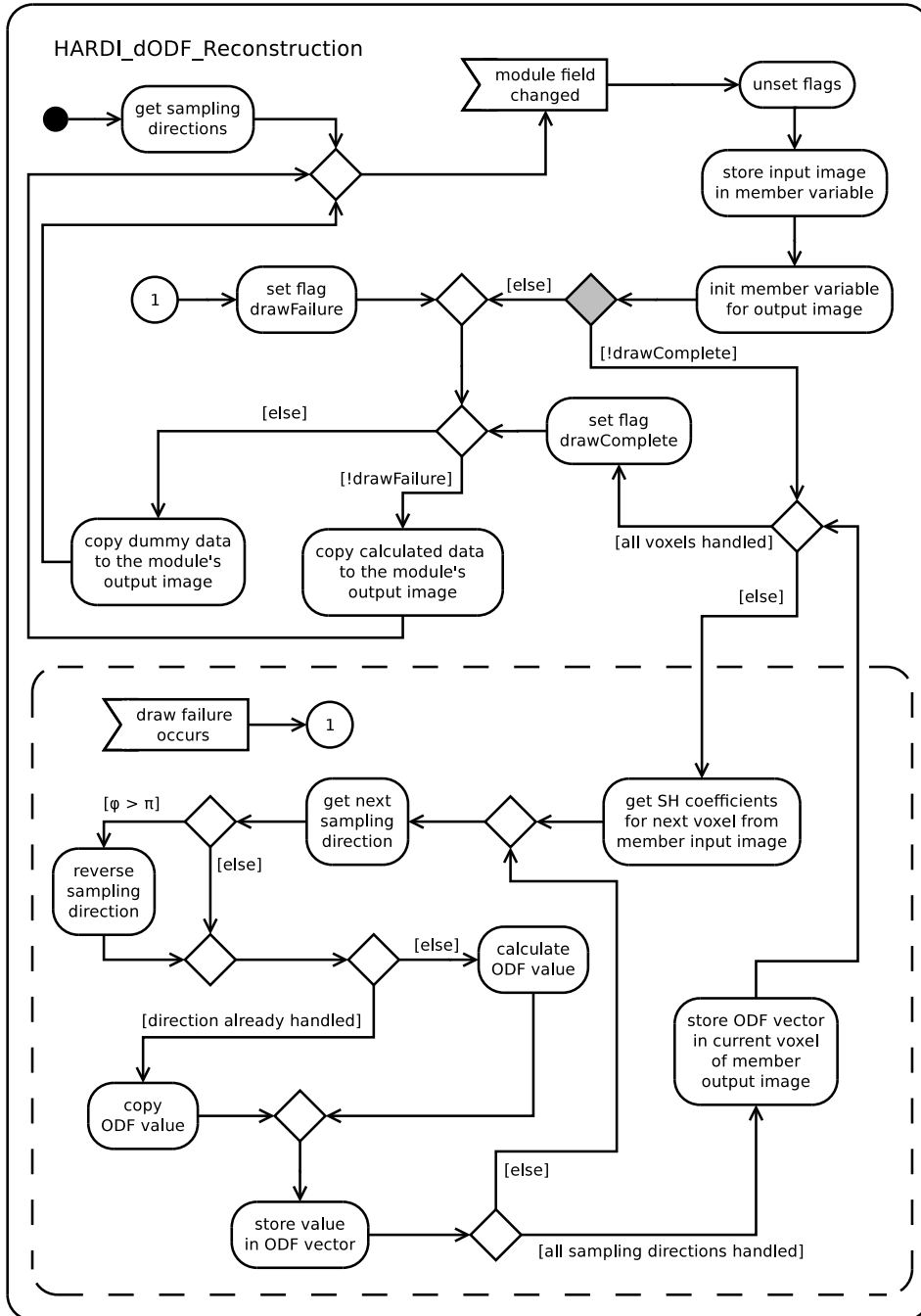


Figure 5.11: Illustration of the control flow in the *HARDI_dODF_Reconstruction* module.

a *VirtualVolume** for the output image is created. The following gray diamond node indicates the various calls of the *calculateOutputSubImage* function. The *drawComplete* flag prevents various computation of the output image data. At the first call of this function the dODF reconstruction for each voxel is performed with the corresponding SH coefficients. As the dODF is symmetric the reconstruction is performed with the sampling directions of only one half sphere. However, the coordinates of the sampling directions are not sorted. Therefore each individual sampling direction has to be checked to lie on half-sphere that is reconstructed. If the azimuthal angle, ϕ , of a sampling direction is above π (i.e. on the half-sphere that is not reconstructed) the sampling direction is reversed to lie on the opposite side of the sphere. If this direction has already been handled a locally stored dODF value exists. This value is copied and stored in the dODF vector. Otherwise, the value has to be computed and is stored locally and in the dODF vector. After processing all sampling directions for a voxel (i.e. reconstructing the whole dODF) the dODF vector is stored in the corresponding voxel of the member output image. The reconstruction continues with the next voxel. As soon as the dODF for all voxels is calculated the *drawComplete* flag is set. The calculated data is copied from the member output image to the module's output image and can be processed by subsequent modules in the network. If an error occurs during reconstruction the calculation is aborted and the *drawFailure* flag is set. In this case "dummy data" is copied to the output image in order not to provide an invalid image at the modules output. The "dummy data" is in this case the highest value of the input image. Consequently, this kind of output results in bright or even completely white images.

5.3.2 HARDI_AnisotropyCriteria

The structure of the *HARDI_AnisotropyCriteria* module is illustrated in Figure 5.12. This module is a subclass of the class *ml::Module*. It allows to calculate anisotropy criteria introduced in section 4.1.3 on input images with HARDI data. Additionally, the GFA (see section 3.4.1) and the standard deviation of the dODF are provided as criteria. The goal of these six criteria is to classify the number of fiber populations per voxel (i.e. 0, 1 or 2 populations) or to separate voxels with isotropic diffusion from voxels with anisotropic diffusion (i.e. 0, or 1 fiber populations). As can be seen in Figure 5.12 the computation and output of the criteria is separated (note the functions starting with *calculate* and *output* respectively). To allow fast switching between the different criteria all needed data for dODF based criteria is precomputed and stored in member variables. Criteria based on the coefficients are calculated on demand. The criterion selection via the GUI-tab (see below) only triggers the corresponding output function.

The *HARDI_AnisotropyCriteria* module has two input image connectors.

5.3. DETAILED MODULE DESCRIPTION

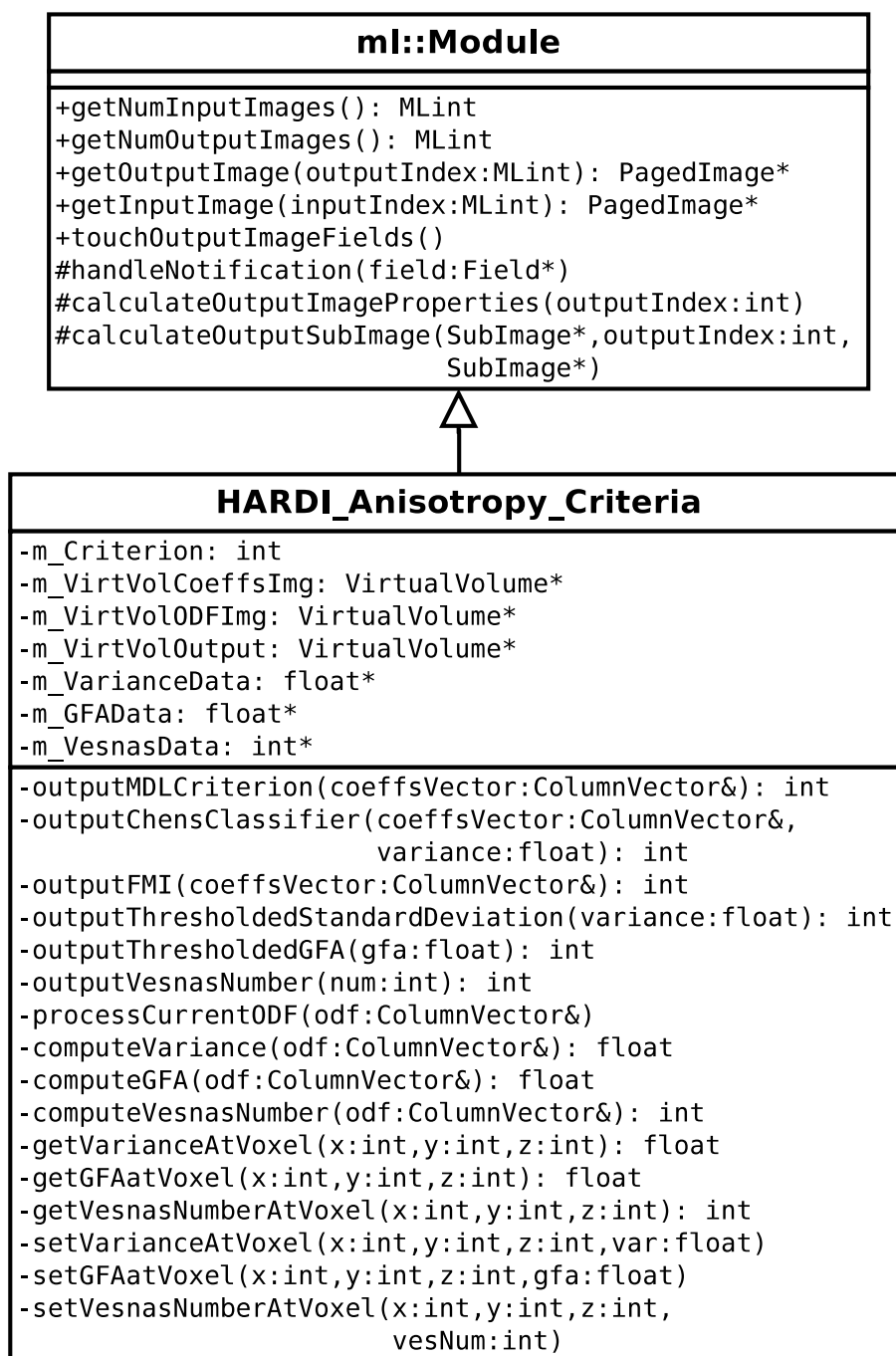


Figure 5.12: Class diagram of the *HARDI_AnisotropyCriteria* module

The number of voxels in both input images has to be indicated by the extents of the x , y , and z dimension. The extent of the dimensions c and u has to be 1. The left input connector is reserved for the SH coefficients im-

5.3. DETAILED MODULE DESCRIPTION

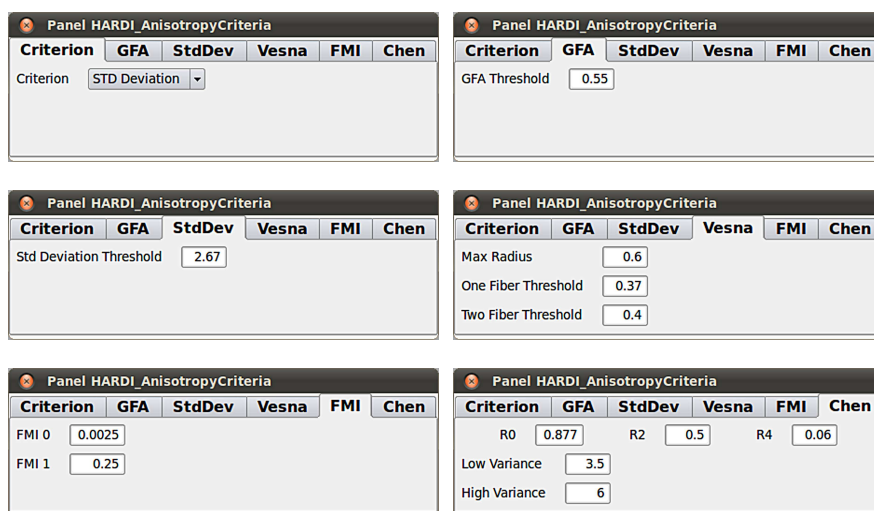


Figure 5.13: GUI-panel of the *HARDI_AnisotropyCriteria* module

age. These coefficients have to be placed in dimension t . An image with reconstructed dODF values in dimension t has to be connected to the right input connector. Both input images have to be provided in order for this module to work properly. The SH coefficients image and the dODF image are both needed to support anisotropy criteria that are calculated on the coefficients (*FMI*, *Chen*, and *MDL*) as well as such that are calculated on the dODF (*GFA*, *stdDev*, *Vesna*, and again *Chen*).

The output image contains the estimated number of fiber populations at each voxel. Typically, the maximum number of fiber populations classified by this module is 2. However, with higher SH orders than 4 the *MDL* criterion can theoretically identify more fiber populations per voxel. The extent of dimension t is set to 1. The extent of the other dimensions is the same as in the input images.

The different anisotropy criteria can be chosen via the *Criterion* tab in the GUI-panel (Figure 5.13). For each of the offered criteria apart from *MDL* a separate GUI tab is provided to define thresholds (*MDL* is a threshold free criterion). For more details on the calculation of the individual values for each criterion see section 4.1.3.

The thresholds for *GFA* and *standard deviation* (in the GUI tab *StdDev*) separate voxels with isotropic diffusion (values below the thresholds) from voxels with anisotropic diffusion (values equal to or above the thresholds). The *GFA* values are normalized to be in $[0, 1]$, however, the standard deviation can take arbitrary values.

The method proposed by Prčkovska *et al.* that was described in section 4.1.3 together with the mentioned extension can be found in the GUI tab *Vesna*. This method counts dODF values above the threshold *Max Radius* in

5.3. DETAILED MODULE DESCRIPTION

the min-max normalized dODF. The resulting value is further transformed to the range $[0, 1]$ and two thresholds are applied to it. Voxels with a value below *One Fiber Threshold* are classified as containing one fiber population. Voxels with a value above *Two Fiber Threshold* are classified as isotropic. Finally, voxels with a value between these two thresholds are classified as containing two fiber populations.

For the *FMI* method two values are calculated: *FMIa* indicating the significance of higher order SH coefficients over the 0th order coefficients and *FMIb* indicating the significance of higher order SH coefficients over the 2nd order coefficients. The classification is carried out according to algorithm 1.

Algorithm 1 FMI criterion classification

```
1: if  $FMIa < FMI0$  then  
2:   0 fiber populations  
3: else  
4:   if  $FMIb < FMI1$  then  
5:     1 fiber population  
6:   else  
7:     2 fiber populations  
8:   end if  
9: end if
```

The criterion in the tab *Chen* demands for the most thresholds. Values R_0 , R_2 , and R_{multi} are calculated as described in section 4.1.3. Additionally, the variance of the dODF is needed for this criterion. The voxels are classified according to algorithm 2. Classification examples are presented in chapter 6.

Algorithm 2 Chen's criterion classification

```
1: if  $R_0 > R0$  or  $variance < Low\ Variance$  then  
2:   0 fiber populations  
3: else  
4:   if  $R_2 > R2$  or  $variance \geq High\ Variance$  then  
5:     if  $R_{multi} > R4$  then  
6:       2 fiber populations  
7:     else  
8:       1 fiber population  
9:     end if  
10:  else  
11:    2 fiber populations  
12:  end if  
13: end if
```

5.3. DETAILED MODULE DESCRIPTION

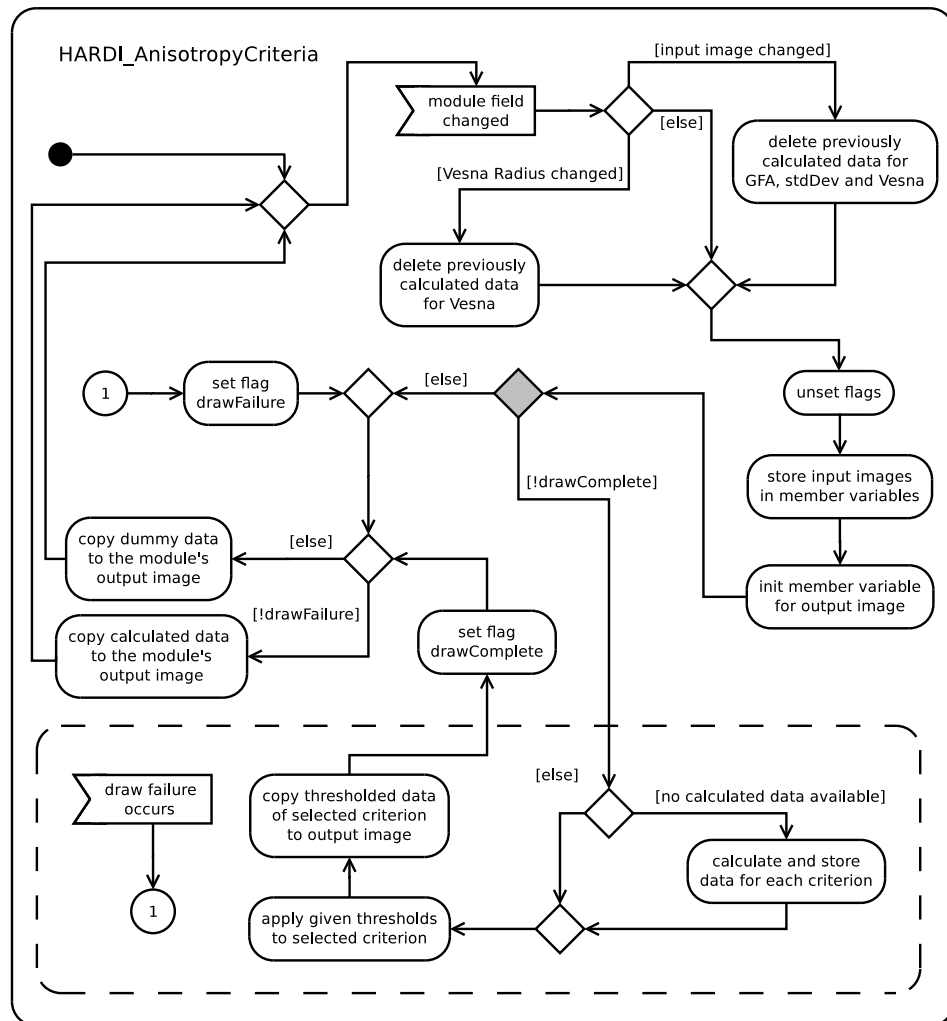


Figure 5.14: Illustration of the control flow in the *HARDI_AnisotropyCriteria* module.

The control flow of this module is illustrated in Figure 5.14. In order to be able to switch between the different criteria without long computation times a majority of data needed for classification is precomputed and stored in member variables. Consequently, on input image changes these data has to be deleted. The *Radius* threshold of the criterion *Vesna* is involved in the data computation for this criterion prior to further thresholding. Therefore, a change of this criterion has to be handled separately. All other field changes (i.e. the various thresholds) can be applied to the calculated data without the need of long computation times. Afterwards, the flags of the module are unset. The input images are copied to member variables and the output image member variable is initialized. The following gray dia-

5.3. DETAILED MODULE DESCRIPTION

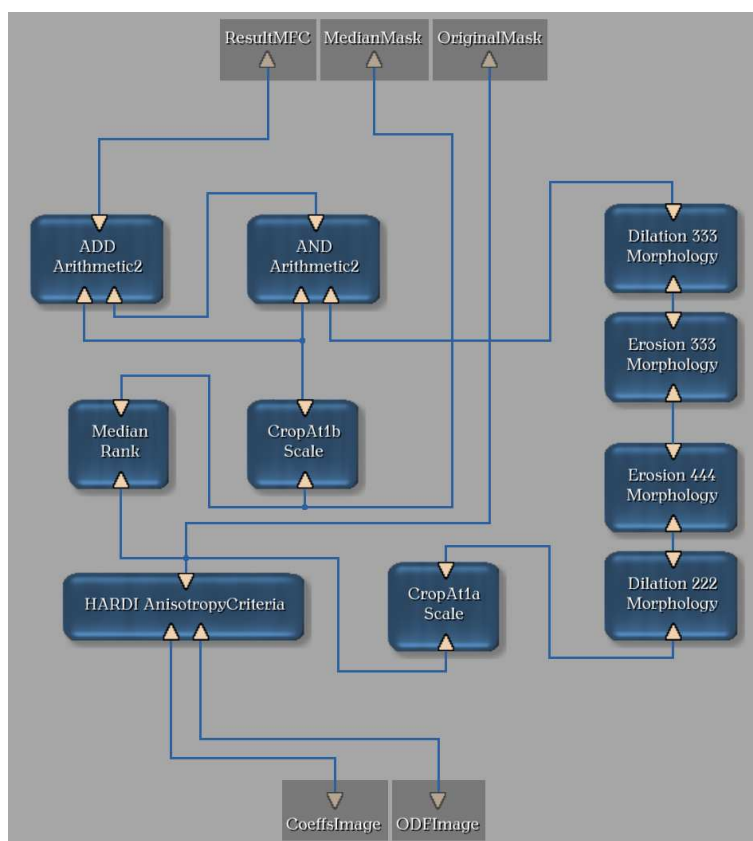


Figure 5.15: Inner network of the *MFC_Macro* module.

mond node indicates again the various calls to the *calculateOutputSubImage* function. Various computation of the output image data are prevented by the *drawComplete* flag. If no precomputed data is available the computation is done in the next step. Otherwise, only the thresholding of the selected criterion is applied to the data. The result is copied to the output image.

5.3.3 MFC_Macro

As the classification results of the *HARDI_AnisotropyCriteria* module for voxels with two fiber populations were insufficient the *MFC_Macro* module was developed to overcome these limitations. This module incorporates an inner network (Figure 5.15) that was built around the *HARDI_AnisotropyCriteria* module. Several other modules of the MeVisLab library, especially morphological operations, were added to achieve the desired functionality. The module's name *MFC* (morphological fiber classification) attributes to these morphological modules. The GUI-panel of this module has exactly the same structure and functionality as the GUI-panel of the *HARDI_AnisotropyCriteria* module (Figure 5.13).

5.3. DETAILED MODULE DESCRIPTION

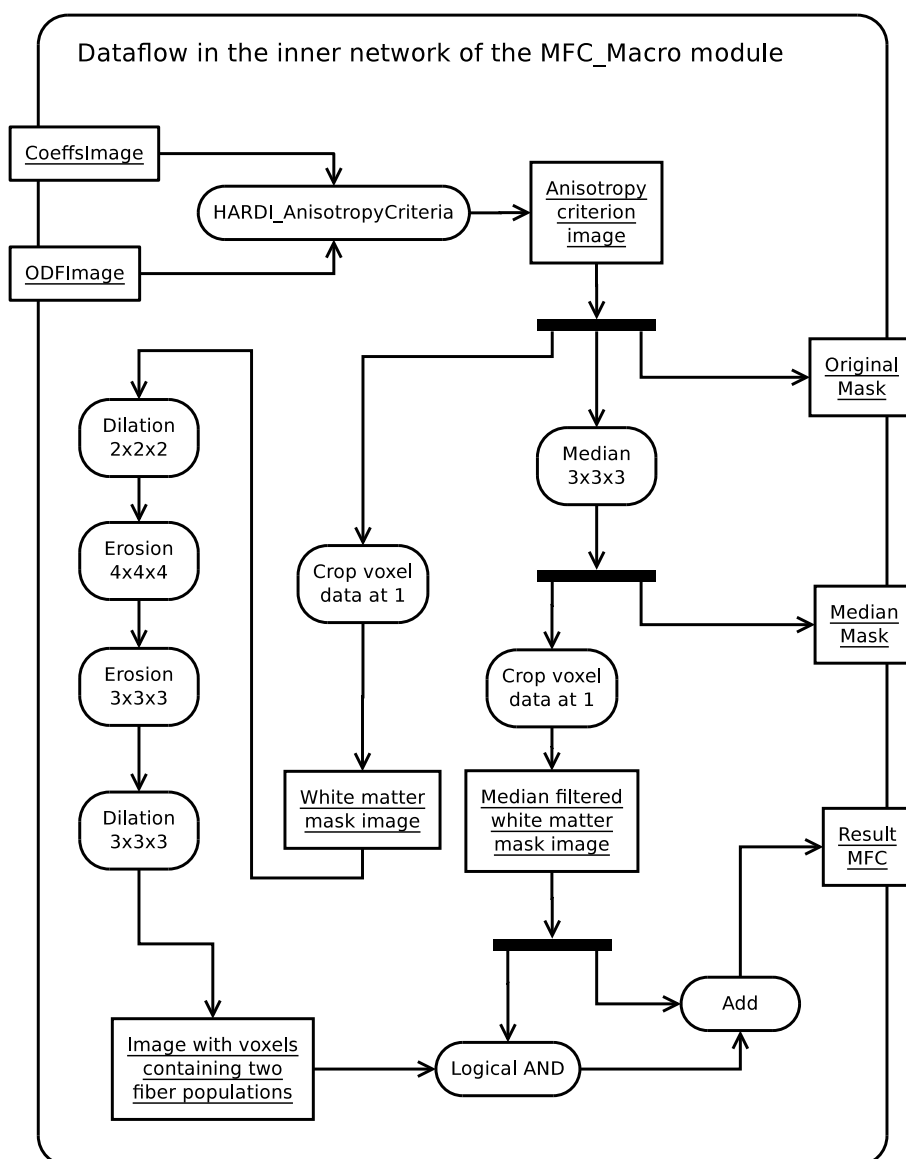


Figure 5.16: Dataflow in the inner network of the *MFC_Macro* module.

The input images of this module have the same properties as the input images of the *HARDI_AnisotropyCriteria* module as they are directly passed to this module. However, the *MFC_Macro* module offers three output images. The image at the rightmost output pin is the output of the *HARDI_AnisotropyCriteria* module. The middle output pin provides a median filtered version of the latter image. A kernel size of $3 \times 3 \times 3$ is used for filtering. The leftmost output pin offers the actual classification result of the MFC.

5.3. DETAILED MODULE DESCRIPTION

The dataflow of the *MFC_Macro* module is illustrated in Figure 5.16. Both input images are passed to the *HARDI_AnisotropyCriteria* module. Its output is an image classifying voxels as containing 0, 1 or 2 fiber populations according to the selected criterion. This image is forwarded to the right output connector (*OriginalMask* in Figure 5.16). The same image is median filtered and passed to the middle output connector. The values of the original output of the *HARDI_AnisotropyCriteria* module as well as the median filtered version of the output are cropped at 1 (i.e. all voxels containing values greater 1 are set to 1). This is necessary as the last step of the *MFC_Macro* module is an addition. By cropping the values at 1 it is ensured that the maximum output value of the MFC result is 2. The *White matter mask image* undergoes several morphological operations. First, small gaps are closed by dilation with kernel size $2 \times 2 \times 2$. Afterwards, white matter tracts are thinned out by applying an erosion with kernel size $4 \times 4 \times 4$. The next operation is morphological opening (erosion followed by dilation) with a kernel size of $3 \times 3 \times 3$. The resulting image contains values 1 only at voxels with two fiber populations. The subsequently applied logical *AND* operation with the median filtered white matter mask eliminates erroneously classified voxels in gray matter as well as noisy voxels. In the final step the median filtered mask image (containing values 1 at voxels with one fiber population) is added to the morphological image. Thus, the resulting image, *ResultMFC*, contains information on voxels with one and two fiber populations respectively. This image is passed to the left output connector.

As the MFC approach provided good results on the Fiber Cup phantom (section 6.3) an article describing this method was written in the course of this thesis. This article was accepted for publication at the *Bildverarbeitung für die Medizin* workshop¹⁰, held in march 2011 in Lübeck, Germany.

5.3.4 HARDI_deterministic_Tractography

The *HARDI_deterministic_Tractography* module is the most complex module that was implemented in the course of this thesis and can therefore be considered as the "heart" of this implementation.

The decision to implement deterministic tracing algorithms instead of probabilistic or global algorithms was made for several reasons. Firstly, the algorithms should be evaluated on the Fiber Cup data and compared to the algorithms that were evaluated at this competition. Since all approaches except one that were presented at the Fiber Cup were deterministic this type of algorithm was chosen for better comparison with the evaluated state-of-the-art algorithms. Secondly, probabilistic algorithms allow to assign probabilities to reconstructed fiber pathways, thus, giving a statement

¹⁰<http://www.charite.de/medinfo/BVM2011/BVM2011.html>

on their plausibility. However, a ground truth is provided for the Fiber Cup dataset. It allows to decide even for deterministically reconstructed pathways whether they are correctly reconstructed or not. Therefore, deterministic algorithms were chosen due to their significantly lower computation times. Lastly, one of the tasks of this thesis is to implement current fiber tracing algorithms and extend them to work on HARDI data. In practice, this means to design these algorithms from scratch as no fiber tracing MeVisLab modules, neither for DTI nor for HARDI data, were previously implemented in the Computer Graphics working group. As was described in chapter 4 probabilistic (and even some global) approaches are based on deterministic streamline propagation. Therefore, it is inevitable to implement deterministic algorithms before proceeding to other methods.

Module Description

The structure of the *HARDI_deterministic_Tractography* module is illustrated in Figure 5.17. For better overview the type name *ColumnVector* was replaced by *CV* in the class diagram. This module is a subclass of the class *ml::Module*. It uses the coordinates of the class *SpherePoints* as possible directions for fiber tracing. The task of this module is to reconstruct white matter fibers from an input image containing a dODF for each voxel. Five different deterministic fiber tracing algorithms are provided for this purpose and will be described below in detail. The reconstruction of fibers for all algorithms is performed bidirectionally starting at seedpoints that can be defined via the GUI-tab.

This module has two input image connectors. The number of voxels in both input images has to be indicated by the extents of the x , y , and z dimension. The extent of the dimensions c and u has to be 1. The left input connector is reserved for an image holding the dODF values in the t dimension. An image indicating the number of fiber populations in each voxel has to be connected to the right input connector. The extent of dimension t of this image has to be 1. The value stored in each voxel is directly interpreted as the number of fiber populations. Isotropic voxels have to be marked with 0. Theoretically, any number of populations can be provided. In practice, however, the maximum number used by this module is 2. Both input images have to be provided in order for this module to work properly.

The output image contains the reconstructed fiber trajectories from all seedpoints. Its structure differs significantly from the other output images generated by the modules described here. The output image is basically a one-dimensional array holding all fiber information in dimension t . The extents of all other dimensions (including x , y and z) is 1. The data is an array of floating point numbers and is organized as shown in Figure 5.18. The first number indicates the quantity of three-dimensional coordinates

5.3. DETAILED MODULE DESCRIPTION

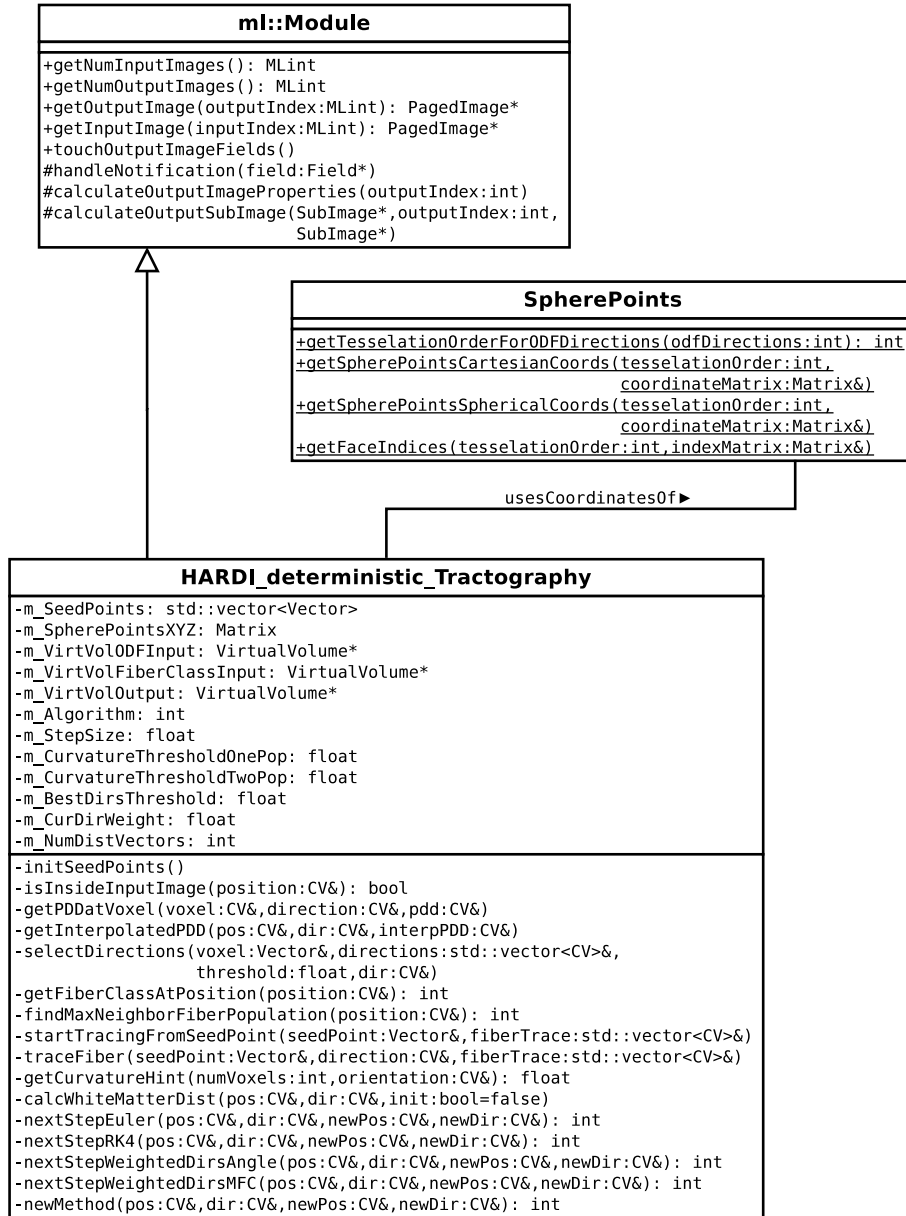


Figure 5.17: Class diagram of the *HARDI_deterministic_Tractography* module

5.3. DETAILED MODULE DESCRIPTION

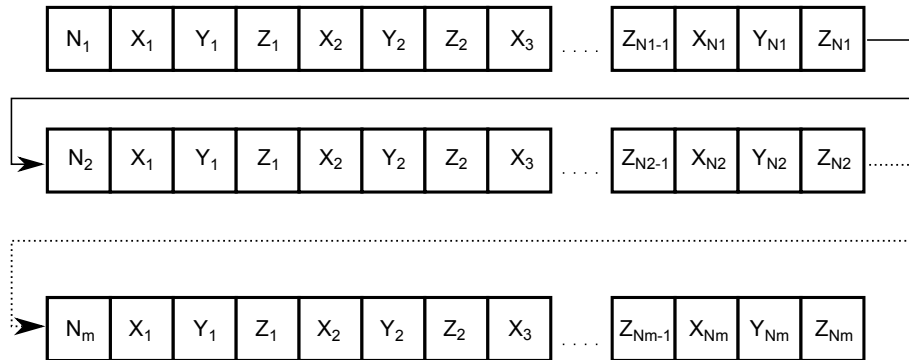


Figure 5.18: Structure of the output image. The array is divided into lines for illustration purposes. Each line represents one fiber.

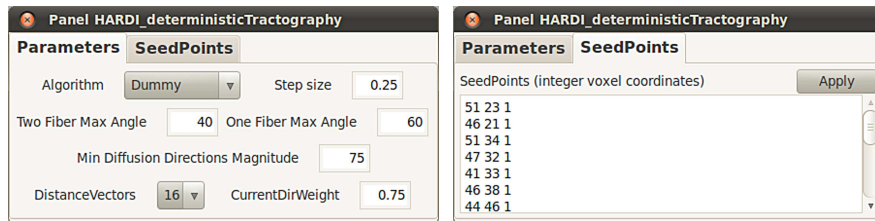


Figure 5.19: GUI-panel of the *HARDI_deterministic_Tractography* module

representing the first fiber. These coordinates are stored in the array directly after the first number. First the x coordinate followed by the y and z coordinates (in this order) of the first vector is placed in the array. Then, the x , y and z coordinates of the second vector and all subsequent vectors follow. If more than one fiber was reconstructed the number of vectors belonging to the next fiber is stored in the array at the position behind the z coordinate of the last vector belonging to the first fiber. Its coordinates follow as described above. The total amount of floating point numbers stored in the fiber image and, thus, the extent of dimension t is $s + v * 3$, where s is the number of seedpoints and v the sum of vertices in all fibers.

The GUI-panel of this module has two tabs (Figure 5.19). The *SeedPoints* tab allows to specify seedpoints that will be used for fiber tracing. The seedpoints have to be integer voxel coordinates indicating the voxel at the center of which the tracing will be started. Each seedpoint has to be entered in a separate line. Its coordinates have to be separated by an arbitrary number of spaces. By clicking on the *Apply* button an event is triggered and fiber tracing is performed starting at the new seedpoints. If the *Apply* button is not clicked the seedpoints will be copied on the next module field change of the *Parameters* tab. Any coordinates outside of the image dimensions will be ignored.

The *Parameters* tab allows to set different parameters for fiber tracing.

5.3. DETAILED MODULE DESCRIPTION

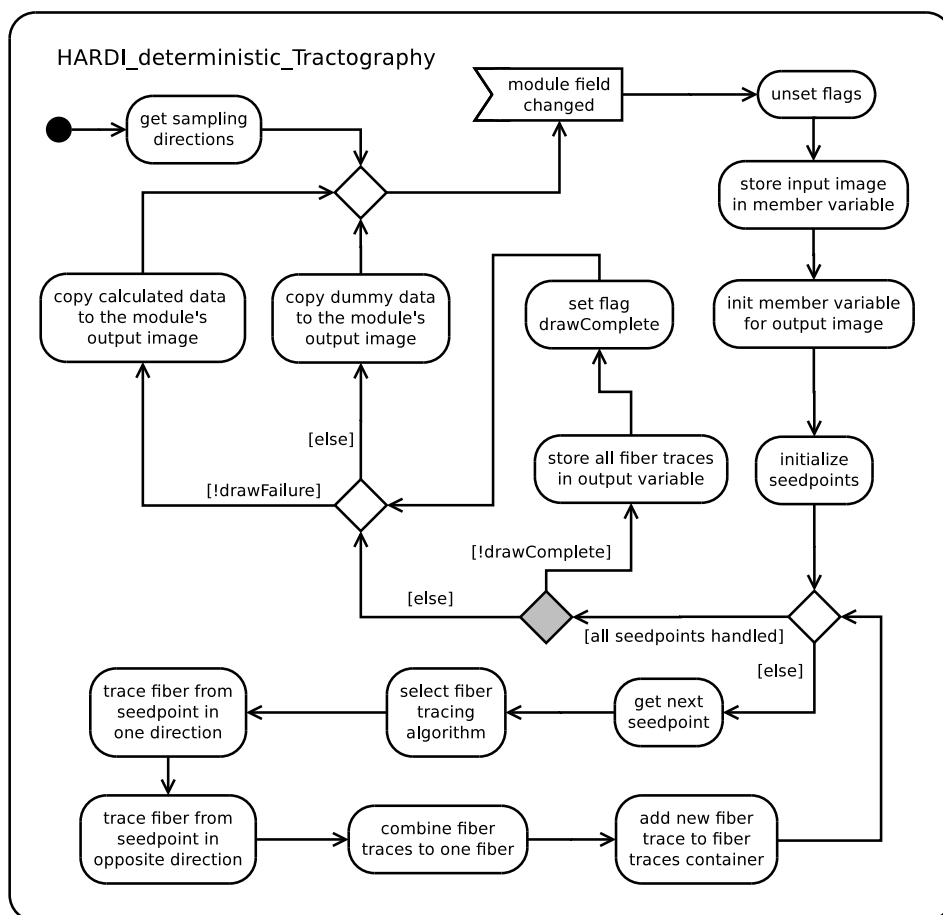


Figure 5.20: Illustration of the control flow in the *HARDI_deterministic_Tractography* module.

The parameter *Algorithm* defines the algorithm that is used. Possible values are *Dummy* (no fiber tracing), *Euler PDD*, *RK4 PDD*, *Weighted Dirs Angle*, *Weighted Dirs MFC* and *Distance Based*. The different algorithms are described in detail in the next section. The parameter *Step size* indicates the integration step size used for fiber tracing. The step size is specified in voxels. Using *One Fiber Max Angle* and *Two Fiber Max Angle* the maximum angle a fiber can bend in every step can be specified for voxels with one and voxels with two fiber populations respectively. The parameter *Min Diffusion Directions Magnitude* is a lower bound for dODF directions that are used for fiber tracing. The given value specifies the percentage of the magnitude of the PDD. The last two parameters only apply to the *Distance Based* algorithm. Via *DistanceVectors* the number of distance vectors that is used to estimate the best tracing direction can be selected. Possible values are 4, 8, and 16. The parameter *CurrentDirWeight* is the weighting factor for

5.3. DETAILED MODULE DESCRIPTION

Algorithm 3 Trace Fiber Algorithm

Input: seed point: sp , tracing direction: $initDir$

Output: list of fiber coordinates: $fiberTrace$

```
1:  $tracePos \leftarrow$  center or voxel  $sp$ 
2:  $traceDir \leftarrow initDir$ 
3:  $continueTracing \leftarrow \mathbf{true}$ 
4: while  $continueTracing = \mathbf{true}$  do
5:   create empty vectors  $newTracePos$  and  $newTraceDir$  for output
6:    $state \leftarrow nextStep(tracePos, traceDir, newTracePos, newTraceDir)$ 
7:   if  $state = \mathbf{VALID\_POSITION}$  then
8:     add  $newPos$  to  $fiberTrace$ 
9:   end if
10:  if  $state = \mathbf{VALID\_POSITION}$  and new voxel entered then
11:    save PDD of new voxel as curvature hint
12:  end if
13:  if  $state = \mathbf{INVALID\_POSITION}$  then
14:     $continueTracing \leftarrow \mathbf{false}$ 
15:  end if
16:  if fiber trajectory of last 4 voxels lies in gray matter then
17:     $continueTracing \leftarrow \mathbf{false}$ 
18:  end if
19:   $tracePos \leftarrow newTracePos$ 
20:   $traceDir \leftarrow newTraceDir$ 
21: end while
```

the current tracing direction of this algorithm. This value has to be between 0 and 1.

Figure 5.20 illustrates the control flow in this module. One important difference to other ML modules presented here is that the calculation of the output data is performed in the *calculateOutputImageProperties* function instead of the *calculateOutputSubImage* function. This is done because MeVis-Lab needs the extension of the output image before the function *calculateOutputSubImage* is called. However, the extent of the output image is known only after performing fiber tracing. After a module field change flags are unset and the input and output images initialized. In the next step seedpoints are copied from the GUI-tab and fiber tracing starts with the selected algorithm. Every algorithm follows a different strategy to reconstruct fibers. More details on the process of fiber tracing are given in the next section. Each fiber is traced in two opposite directions starting at the corresponding seedpoint. After tracing, both trajectories are combined to form a consecutive fiber. The resulting fiber is stored in a member variable. As soon as all seed points are processed the fiber trace output image is generated as described above.

Algorithm Description

Selecting seedpoints and determining whether or not fiber tracing has to be stopped is handled equally for all implemented algorithms. These procedure is shown in Algorithm 3. This algorithm is called twice (with opposite tracing directions *initDir*) for every seedpoint *sp*. The output of the algorithm is a list of coordinates representing the fiber trace.

The tracing position and direction are initialized in lines 1 and 2. Fiber tracing is continued until one of two stopping conditions is met (see below). The streamline is advanced one step according to the selected fiber tracing algorithm in line 6. These algorithms are explained below. If the new position is valid it is added to the output list *fiberTrace*. Further, if a new voxel is entered the PDD of the voxel is saved to be used as curvature hint in some of the tracing algorithms. Tracing is stopped if the new position is invalid (i.e. outside of the dataset) or the fiber trajectory has only been in gray matter in the last four voxels.

Euler PDD and RK4 PDD These two algorithms are shown in Algorithm 4. These algorithms are extension of the classic DTI principal diffusion direction (PDD) tracing approaches to HARDI data. The first line of the algorithm decides which of the two integration methods is used (replace *XXX* by *Euler* or *RK4*). The new position, *newPos*, is obtained by integrating one step in the current direction. If this new position is outside the image the status constant `INVALID_POSITION` is returned. In the other case the number of fiber populations is determined at the current position. This is done in two steps. First, the classification result of the *MFC_Macro* module is read at the voxel in question. Second, if this result is 1 the 26 neighboring voxels are examined. In the case that at least one of these voxels has 2 fiber populations the number of fiber populations at the current position is also considered as 2. This virtual extension of the 2 fiber population areas is necessary as the *MFC_Macro* module sometimes underestimates the extent of areas with multiple fiber populations.

The algorithm continues in line 4 or line 12 depending on the number of found fiber populations. The procedure in both cases is equal except that in the case of less than 2 fiber populations the interpolated PDD is examined as the possible new direction. In the other case the PDD without interpolation is regarded. This new direction is accepted and assigned to the output variable *newDir* only if it is below the curvature threshold specified via the GUI-tab. In the other case tracing continues in the current direction.

The PDD without interpolation is used in the case of multiple fiber populations as the interpolated PDD always smoothly turns the fiber trajectory towards the dominant fiber. Thus, tracing along a nondominant fiber bundle is impossible. The PDD, however, is ignored if its curvature is too high (i.e. in regions of crossings) but can nevertheless influence the fiber trajec-

5.3. DETAILED MODULE DESCRIPTION

Algorithm 4 Euler/RK4 PDD Fiber Tracing

Input: current position: $curPos$, current direction: $curDir$

Output: new position: $newPos$, new direction: $newDir$,
integration state: $state$

```
1:  $newPos \leftarrow$  next position with XXX integration using  $curPos$  and  
    $curDir$   
2: if  $newPos$  is inside input image then  
3:    $fp \leftarrow$  number of fiber populations at  $newPos$   
4:   if  $fp < 2$  then  
5:      $interpPdd \leftarrow$  interpolated PDD at  $newPos$   
6:      $angle \leftarrow$  angle between  $interpPdd$  and  $curDir$   
7:     if  $angle > ONE\_FIBER\_MAX\_ANGLE$  then  
8:        $newDir \leftarrow curDir$   
9:     else  
10:       $newDir \leftarrow interpPdd$   
11:    end if  
12:  else if  $fp = 2$  then  
13:     $pdd \leftarrow$  PDD at  $newPos$   
14:     $angle \leftarrow$  angle between  $pdd$  and  $curDir$   
15:    if  $angle > TWO\_FIBER\_MAX\_ANGLE$  then  
16:       $newDir \leftarrow curDir$   
17:    else  
18:       $newDir \leftarrow pdd$   
19:    end if  
20:  end if  
21:   $state \leftarrow VALID\_POSITION$   
22: else  
23:   $state \leftarrow INVALID\_POSITION$   
24: end if
```

tory in regions of fiber kissings or branchings. Further, following a PDD whenever possible (i.e. where it is below the threshold) in uncertain regions is better than following the current direction as the current direction might not be representing a true diffusion direction in the next voxel.

Weighted Dirs Angle The previous algorithms only used the PDD in each voxel. Their extension to HARDI data was limited to the knowledge of whether or not the current position is inside a multiple fiber area. However, much more information can be extracted from HARDI data and used for fiber tracing. The basic idea of the *Weighted Dirs Angle* algorithm is to consider several possible directions in each voxel (instead of only using the PDD) that might be used to continue the streamline. Different weights

5.3. DETAILED MODULE DESCRIPTION

are assigned to these directions and the resulting direction is calculated. The weighting procedure depends on the scattering angle of the highest diffusion directions in each voxel. The scattering angle is determined in two steps. First, all direction vectors with dODF values of at least 70% the dODF value of the PDD are selected. Then, the maximum angle between two vectors in this list is determined as the scattering angle. High scattering angles might indicate a multiple fiber population. The *MFC* is not used in this approach. The *Weighted Dirs Angle* algorithm is shown in Algorithm 5.

In this algorithm the direction vectors indicating the highest diffusion (at least 70% of the dODF value of the PDD) in the current voxel are obtained and stored in the list of vectors *highestDirs*. Since diffusion is symmetric direction vectors with an angle greater 90 degrees to *curDir* are reversed to form an angle of less than 90 degrees. In these directions the closest (i.e. with the smallest separating angle) to the current direction is found. Subsequently, the scattering angle of the highest directions is determined. The higher this angle the higher the uncertainty associated with the diffusion direction in this voxel. This scattering angle is stored in *scatteringAngle*. In line 4 the curvature threshold is set.

The calculation of the new direction is performed in lines 5 to 31. Tracing is continued in the current direction if the angle between the current direction and the closest one is above the threshold. In the other case weighting factors are calculated (lines 8 and 9) and the current and closest directions are combined with their corresponding weights and stored in *newDir* (line 10). This line also initializes *newDir* in the case that none of the following conditions apply.

The following steps depend on *scatteringAngle*. If it is below 20 degrees tracing is continued along the PDD. In the case that the scattering is above 20 and below 90 degrees a more complicated heuristic is used. The average curvature angle of the last 15 voxels is determined in line 15. The curvature hint is based on the PDDs of the voxels and not the actual fiber course as the average PDD more reliably indicates whether or not a curved fiber is present. The decision of whether or not the PDD itself should be used in the calculation of the new direction is made in lines 16 to 21. The PDD is used in the case that one step along the PDD would result in a curvature more similar to the curvature hint than a step along the closest direction. In this cases the resulting *newDir* is calculated based on the current direction and the PDD. The PDD weighting factor is determined based on the angle between the PDD and the current direction. The resulting direction is normalized in line 32. Finally, the streamline is propagated one step size along the new found direction (line 33) and the corresponding *state* is returned.

5.3. DETAILED MODULE DESCRIPTION

Algorithm 5 Weighted Dirs Angle Fiber Tracing

Input: current position: $curPos$, current direction: $curDir$

Output: new position: $newPos$, new direction: $newDir$,
integration state: $state$

```
1:  $highestDirs \leftarrow$  highest diffusion directions at  $curPos$  using  $curDir$ 
2:  $closestDir \leftarrow$  most similar direction vector to  $curDir$  in  $highestDirs$ 
3:  $scatteringAngle \leftarrow$  scattering angle of  $highestDirs$ 
4:  $threshold \leftarrow$  ONE_FIBER_MAX_ANGLE
5: if angle between  $closestDir$  and  $curDir > threshold$  then
6:    $newDir \leftarrow curDir$ 
7: else
8:    $curDirWeight \leftarrow \frac{scatteringAngle}{2 \cdot threshold}$ 
9:    $closestDirWeight \leftarrow |curDir \cdot closestDir|$ 
10:   $newDir \leftarrow curDir \cdot curDirWeight + closestDir \cdot closestDirWeight$ 
11:   $pdd \leftarrow$  PDD at  $curPos$ 
12:  if  $scatteringAngle < 20$  then
13:     $newDir \leftarrow pdd$ 
14:  else if  $scatteringAngle < 90$  then
15:     $curv \leftarrow$  curvature hint at  $curPos$ 
16:     $tempAnglePDD \leftarrow$  angle between  $curDir$  and  $pdd$ 
17:     $tempAngleClosest \leftarrow$  angle between  $curDir$  and  $closestDir$ 
18:     $usePDD \leftarrow$  false
19:    if  $|curv - tempAnglePDD| < |curv - tempAngleClosest|$  then
20:       $usePDD \leftarrow$  true
21:    end if
22:     $pddWeight \leftarrow |curDir \cdot pdd|$ 
23:    if  $usePDD = \mathbf{true}$  or  $tempAnglePDD < 25$  then
24:       $pddWeight \leftarrow pddWeight^2$ 
25:       $newDir \leftarrow curDir \cdot curDirWeight + pdd \cdot pddWeight$ 
26:    else if  $usePDD = \mathbf{true}$  and  $tempAnglePDD < 35$  then
27:       $pddWeight \leftarrow pddWeight^3$ 
28:       $newDir \leftarrow curDir \cdot curDirWeight + pdd \cdot pddWeight$ 
29:    end if
30:  end if
31: end if
32:  $newDir \leftarrow \frac{newDir}{\|newDir\|}$ 
33:  $newPos \leftarrow curPos + newDir \cdot STEP\_SIZE$ 
34: if  $newPos$  is inside input image then
35:    $state \leftarrow$  VALID_POSITION
36: else
37:    $state \leftarrow$  INVALID_POSITION
38: end if
```

Weighted Dirs MFC Lots of heuristic assumptions on weighting factors and degree limits had to be taken in the design of the last algorithm to allow for reliable streamline propagation through complex regions and regions with high curvature. The algorithm presented in this section tries to avoid some of these assumptions by making use of the classification examples of the *MFC_Macro* module.

The *Weighted Dirs MFC* algorithm is shown in Algorithm 6. Like the last algorithm this approach considers several possible directions for tracing continuation. Different weights are assigned to these directions and the resulting direction is calculated. The weighting depends on the number of fiber populations at the current position (i.e. on the MFC result).

In this algorithm the direction vectors indicating the highest diffusion in the current voxel (line 1), the closest diffusion to the current direction (line 2), and the scattering angle of the highest directions (line 3) is found as described in the last algorithm. In lines 5 to 8 the curvature threshold is set and the weighting factor for the current direction is determined in lines 9 to 13. In the case of two fiber populations the weight of the current direction is doubled to allow streamlines representing nondominant diffusion directions to propagate through complex regions. Subsequently, the weighting factor for the closest direction is determined as the scalar product between the current and the closest direction.

The calculation of the new direction is performed in lines 15 to 27. Tracing is continued in the current direction if the angle between the current direction and the closest one is above the threshold. In the other case these two directions are combined with their corresponding weights and stored in *newDir*. As this result is in most cases only a slightly modified current direction the PDD and its weight are obtained and incorporated into the new direction if the PDD is not above the specified curvature threshold. The streamline is propagated one step size along the resulting direction (line 28) and the corresponding *state* is returned.

Distance Based Fiber Tracing The idea behind the *Distance Based* algorithm is to combine the methods that were used so far. This includes taking into account several directions at each step, applying weighting factors to the new and the current direction, and using the *MFC* for fiber classification. Additionally, distances to white matter borders will be computed at each step and included in the decision process of which direction to take next. Further, this is the only approach that not necessarily takes the PDD of the seed point as the initial tracing direction.

Algorithm 7 illustrates the main steps of this approach. When the algorithm is called the first time (i.e. to make the first step at the seed point) initial distances are calculated (lines 1 to 3). In the first distance calculation an initial tracing direction needs to be found. For this purpose the

5.3. DETAILED MODULE DESCRIPTION

Algorithm 6 Weighted Dirs MFC Fiber Tracing

Input: current position: $curPos$, current direction: $curDir$

Output: new position: $newPos$, new direction: $newDir$,
integration state: $state$

```
1:  $highestDirs \leftarrow$  highest diffusion directions at  $curPos$  using  $curDir$ 
2:  $closestDir \leftarrow$  most similar direction vector to  $curDir$  in  $highestDirs$ 
3:  $scatteringAngle \leftarrow$  scattering angle of  $highestDirs$ 
4:  $fp \leftarrow$  number of fiber populations at  $curPos$ 
5:  $maxAngle \leftarrow$  ONE_FIBER_MAX_ANGLE
6: if  $fp = 2$  then
7:    $maxAngle \leftarrow$  TWO_FIBER_MAX_ANGLE
8: end if
9: if  $fp \leq 2$  then
10:   $curDirWeight \leftarrow \frac{scatteringAngle}{2 \cdot maxAngle}$ 
11: else
12:   $curDirWeight \leftarrow \frac{scatteringAngle}{maxAngle}$ 
13: end if
14:  $closestDirWeight \leftarrow |curDir \cdot closestDir|$ 
15: if angle between  $closestDir$  and  $curDir > maxAngle$  then
16:   $newDir \leftarrow curDir$ 
17: else
18:   $newDir \leftarrow curDir \cdot curDirWeight + closestDir \cdot closestDirWeight$ 
19:   $newDir \leftarrow \frac{newDir}{\|newDir\|}$ 
20:  $pdd \leftarrow$  PDD at  $curPos$ 
21:  $pddWeight \leftarrow |newDir \cdot pdd|$ 
22: if angle between  $pdd$  and  $newDir > maxAngle$  then
23:   $pddWeight \leftarrow 0$ 
24: end if
25:   $newDir \leftarrow pdd \cdot pddWeight + newDir \cdot curDirWeight$ 
26:   $newDir \leftarrow \frac{newDir}{\|newDir\|}$ 
27: end if
28:  $newPos \leftarrow curPos + newDir \cdot STEP\_SIZE$ 
29: if  $newPos$  is inside input image then
30:   $state \leftarrow$  VALID_POSITION
31: else
32:   $state \leftarrow$  INVALID_POSITION
33: end if
```

5.3. DETAILED MODULE DESCRIPTION

Algorithm 7 Distance Based Fiber Tracing

Input: current position: $curPos$, current direction: $curDir$

Output: new position: $newPos$, new direction: $newDir$,
integration state: $state$

```
1: if  $initDistances$  is empty then
2:    $initDistances \leftarrow$  initial orthogonal distances to white matter borders
3: end if
4:  $possibleDirs \leftarrow$  findPossibleDirections (Algorithm 8)
5: if  $possibleDirs$  is empty then
6:    $state \leftarrow$  INVALID_POSITION
7: else
8:    $bestIndex \leftarrow$  findOptimalWhiteMatterDistance (Algorithm 9)
9:   if number of fiber populations at  $curPos = 2$  then
10:     $bestIndex \leftarrow$  findLongDistances (Algorithm 10)
11:   end if
12:    $bestDir \leftarrow$  element at position  $bestIndex$  in  $possibleDirs$ 
13:    $newDir \leftarrow bestDir \cdot (1 - CURRENT\_DIR\_WEIGHT) +$   

 $curDir \cdot CURRENT\_DIR\_WEIGHT$ 
14:    $newDir \leftarrow \frac{newDir}{\|newDir\|}$ 
15:    $newPos \leftarrow curPos + newDir \cdot STEP\_SIZE$ 
16:    $state \leftarrow$  VALID_POSITION
17: end if
```

highest directions (i.e. having at least an dODF value of 70% the dODF of the PDD) of the current voxel are selected. These directions are averaged and the result is added to the list of highest directions. For each of these possible tracing directions 16 (this number can be varied via the parameter *DistanceVectors* in the GUI-tab) orthogonal vectors are generated. Assuming that the tracing direction is parallel to the fiber borders the orthogonal vectors indicate the shortest paths to the edges of the current fiber. The distance to the fiber border for each of the 16 vectors is calculated. The fiber border is reached when a gray matter voxel is entered or the angle between the current direction and the PDD of a voxel crossed by one of the 16 distance vectors is more than ONE_FIBER_MAX_ANGLE. This second condition allows for almost constant fiber widths even in regions with multiple fiber populations. Of all possible tracing directions the one is selected as initial tracing direction that has the smallest maximum distance value. This complicated procedure is necessary as the PDD in the seedpoint might not at all be parallel to the fiber borders and would, thus, lead to an erroneous assumption of the fiber thickness. By selecting the direction with the smallest maximum value it is ensured that the most parallel direction to the fiber borders is selected.

5.3. DETAILED MODULE DESCRIPTION

In the next step (line 4) possible directions are selected to continue the fiber trajectory. The individual steps of this procedure are illustrated in Algorithm 8 and will be explained below in detail. If no possible directions could be found the *state* INVALID_POSITION is returned and the fiber tracing is stopped. In the other case Algorithm 9 is used to find the index of the optimal direction. A direction is optimal if it leads to a position at which the distances to white matter borders are most similar to the initial distances. In this manner a streamline stays at its relative position to the borders of the fiber. More details on this algorithm are given below. If two fiber populations are present in the current voxel the optimal index calculated in Algorithm 9 might be misleading. For example a streamline that was started close to a fiber border will necessarily follow its closest edge at fiber crossings. To avoid these cases Algorithm 10 is used to select a different direction index.

In line 12 the direction vector at the found *bestIndex* in the list of all possible directions is selected. The resulting tracing direction is determined by combining the new direction vector at *bestIndex* and the current direction with the weighting factor *CurrentDirWeight* that can be specified via the GUI-tab. The streamline is advanced one step and the new position is determined.

Algorithm 8 has the purpose to find possible directions to continue the fiber trajectory. First, the directions representing the highest diffusion values are selected. Then, the number of fiber populations of the current voxel is determined. Again, the regions with two fiber populations as found by the *MFC_Macro* module are virtually extended. If the current voxel is in gray matter the task is to return the trajectory to white matter voxels. For this purpose 16 direction vectors forming an angle of 45 degrees with the current direction are generated. Subsequently, it is tested whether one step in the current direction or in the direction of one of the 45 degree vectors leads to white matter. Every direction vector reaching white matter after one step size is added to the list of possible directions. The test is performed again with the double and quadruple step size if no possible direction could be found.

In the case of one or two fiber populations in the current voxel the corresponding curvature threshold is set. After this several directions are tested and added to the list *possibleDirs* if two conditions are met. First, one step in this direction leads to a position that is still inside the dataset. Second, the angle between the current direction and the tested direction is below the threshold. Besides the current direction and the interpolated PDD all directions in *highestDirs* are tested.

The list *highestDirs* with all possible direction vectors is passed to Algorithm 9. This algorithm finds the direction vector that leads to a position at which the distances to white matter borders are most similar to the initial distances. To accomplish this the list of initial distances is grouped in

5.3. DETAILED MODULE DESCRIPTION

Algorithm 8 findPossibleDirections

Input: current position: $curPos$, current direction: $curDir$

Output: list of possible direction vectors: $possibleDirs$

```
1:  $highestDirs \leftarrow$  highest diffusion directions at  $curPos$  using  $curDir$ 
2:  $fp \leftarrow$  number of fiber populations at  $curPos$ 
3: if  $fp = 0$  then
4:    $tempStepSize \leftarrow$  STEP_SIZE
5:    $vectors45Deg \leftarrow$  45 degree vectors around  $curDir$ 
6:   repeat
7:      $tempPos \leftarrow curPos + curDir \cdot tempStepSize$ 
8:     if number of fiber populations at  $tempPos > 0$  then
9:       add  $curDir$  to  $possibleDirs$ 
10:    end if
11:    for all  $tempDir$  in  $vectors45Deg$  do
12:       $tempPos \leftarrow curPos + tempDir \cdot tempStepSize$ 
13:      if number of fiber populations at  $tempPos > 0$  then
14:        add  $tempDir$  to  $possibleDirs$ 
15:      end if
16:    end for
17:     $tempStepSize \leftarrow 2 \cdot tempStepSize$ 
18:  until  $possibleDirs$  is not empty or  $tempStepSize > 4 \cdot STEP\_SIZE$ 
19: else
20:    $maxAngle \leftarrow$  ONE_FIBER_MAX_ANGLE
21:   if  $fp = 2$  then
22:      $maxAngle \leftarrow$  TWO_FIBER_MAX_ANGLE
23:   end if
24:    $tempPos \leftarrow curPos + curDir \cdot STEP\_SIZE$ 
25:   if  $tempPos$  is inside input image then
26:     add  $curDir$  to  $possibleDirs$ 
27:   end if
28:    $interpPdd \leftarrow$  interpolated PDD at  $curPos$ 
29:    $tempPos \leftarrow curPos + interpPdd \cdot STEP\_SIZE$ 
30:   if  $tempPos$  is inside input image and angle between  $curDir$  and
    $interpPdd < maxAngle$  then
31:     add  $interpPdd$  to  $possibleDirs$ 
32:   end if
33:   for all  $tempDir$  in  $highestDirs$  do
34:      $tempPos \leftarrow curPos + tempDir \cdot STEP\_SIZE$ 
35:     if  $tempPos$  is inside input image and angle between  $curDir$  and
      $tempDir < maxAngle$  then
36:       add  $tempDir$  to  $possibleDirs$ 
37:     end if
38:   end for
39: end if
```

5.3. DETAILED MODULE DESCRIPTION

Algorithm 9 findOptimalDistance

Input: list of possible direction vectors: *possibleDirs*

list of initial distances: *initDistances*

Output: index of direction vector with optimal distance: *bestIndex*

```
1: bestDeviance  $\leftarrow \infty$ 
2: initialDistPairsList  $\leftarrow$  pairs of opposite distances from initDistances
3: sort pairs in initialDistPairsList according to their sum
4: for all tempDir in possibleDirs do
5:   tempPos  $\leftarrow$  curPos + tempDir · STEP_SIZE
6:   distanceList  $\leftarrow$  orthogonal distances from tempPos
7:   currentDistPairsList  $\leftarrow$  pairs of opposite distances from
   distanceList
8:   sort pairs in currentDistPairsList according to their sum
9:   deviance  $\leftarrow$  0
10:  for i = 0 to size of currentDistPairsList - 1 do
11:    curPair  $\leftarrow$  element at position i in currentDistPairsList
12:    initPair  $\leftarrow$  element at position i in initialDistPairsList
13:    curFraction  $\leftarrow$   $\frac{\text{minimum of pair } curPair}{\text{sum of pair } curPair}$ 
14:    initFraction  $\leftarrow$   $\frac{\text{minimum of pair } initPair}{\text{sum of pair } initPair}$ 
15:    deviance  $\leftarrow$  deviance +  $|curFraction - initFraction|^2$ 
16:  end for
17:  deviance  $\leftarrow$   $\sqrt{deviance}$ 
18:  if deviance < bestDeviance then
19:    bestDeviance  $\leftarrow$  deviance
20:    bestIndex  $\leftarrow$  index of tempDir
21:  end if
22: end for
```

pairs. Each pair holds the distances of two opposite distance vectors. In the loop starting at line 4 one step is taken along each of the possible directions. At the resulting position distances to white matter borders are calculated. In contrast to the initial distance calculation no tracing direction needs to be found for distance calculation. However, in cases where the average of the highest diffusion directions in the current voxel is closer than 45 degrees to the current direction the mean of the current direction and the average direction is used for distance computations. This allows for better approximation of the diffusion direction in the processed voxel. The found distances are grouped to pairs of opposite distances and sorted according to their sums. In order to compare the distances at the new position to the initial distances one needs to know which two distance vectors correspond to each other. Unfortunately there is no way to do so as it would

5.3. DETAILED MODULE DESCRIPTION

Algorithm 10 findLongDistances

Input: current position: $curPos$, current direction: $curDir$,
list of possible direction vectors: $possibleDirs$
index of optimal white matter distance: $bestIndex$

Output: index of best direction vector for streamline propagation:
 $bestIndex$

- 1: create list to hold indices of vectors with longest frontal white matter distances: $longestDistIndices$
- 2: add $bestIndex$ to $longestDistIndices$
- 3: **for all** $tempDir$ in $possibleDirs$ **do**
- 4: $tempDist \leftarrow$ frontal distance from $curPos$ along $tempDir$
- 5: **if** $tempDist > 2 \cdot STEP_SIZE$ **then**
- 6: add index of $tempDist$ to $longestDistIndices$
- 7: **end if**
- 8: **end for**
- 9: $curvHint \leftarrow$ curvature hint at $curPos$
- 10: $bestAngle \leftarrow TWO_FIBER_MAX_ANGLE$
- 11: **for all** $index$ in $longestDistIndices$ **do**
- 12: $tempDir \leftarrow$ element at position $index$ in $possibleDirs$
- 13: $tempAngle \leftarrow$ angle between $curDir$ and $tempDir$
- 14: **if** $|curvHint - tempAngle| < bestAngle$ **then**
- 15: $bestAngle \leftarrow |curvHint - tempAngle|$
- 16: $bestIndex \leftarrow index$
- 17: **end if**
- 18: **end for**

imply to assign descriptors like *up*, *down*, *left*, and *right* to arbitrary vectors in three-dimensional space. However, assuming that the fiber width does not change much along the fiber the pairs of the initial distances and the current distances are sorted. It is assumed that the distance pairs at the same position in both lists correspond to each other. In lines 10 to 17 the deviance from the initial distances for each possible fiber direction is calculated. A distance fraction is determined for two corresponding distance pairs of both lists. The squared difference is added to the deviation. After summing up all squared deviances the square root is taken and the resulting deviance for a possible direction is obtained. The index of the smallest deviance is assigned to $bestIndex$.

Finally, Algorithm 10 decided whether the direction at the computed $bestIndex$ or at a different position should be taken to propagate the fiber trajectory. For each of the possible directions in $possibleDirs$ the distance from the current position along this direction to the next gray matter voxel is calculated. If this distance is longer than twice the $STEP_SIZE$ the direction's index is added to the list $longestDistIndices$. The index $bestIndex$

5.3. DETAILED MODULE DESCRIPTION

from Algorithm 9 is always added to this list. In lines 11 to 18 the angle between the current direction and each direction indicated by the indices in *longestDistIndices* is compared to the curvature hint angle. The index of the direction that best matches the curvature hint angle is assigned to *bestIndex* and will be used to propagate the streamline one step further.

5.3.5 So_dODF_Visualization

The structure of the *So_dODF_Visualization* module is illustrated in Figure 5.21. This module is a subclass of the class *SoShape* and uses the class *SpherePoints*. The purpose of the *So_dODF_Visualization* module is to visualize the diffusion ODFs and, optionally, to visualize voxel groups if some kind of classification (e.g. anisotropy measure) is applied to the data. This module is useful to examine the data and identify problematic regions. Thus, it is helpful in deriving proper thresholds and heuristics for fiber tracing algorithms. Further, it helps to evaluate classification algorithms (e.g. from the *MFC_Macro* module) as it offers a method to visualize different voxel groups.

The colors of the dODFs are calculated according to equations 3.30. Thus, directions primary oriented along the *x*-axis are rendered in red, directions oriented along the *y*-axis are displayed in green, and directions oriented along the *z*-axis are shown in blue.

This module offers three input connectors. All input images have to store the extent of the data in voxels in the dimensions *x*, *y* and *z*. The main

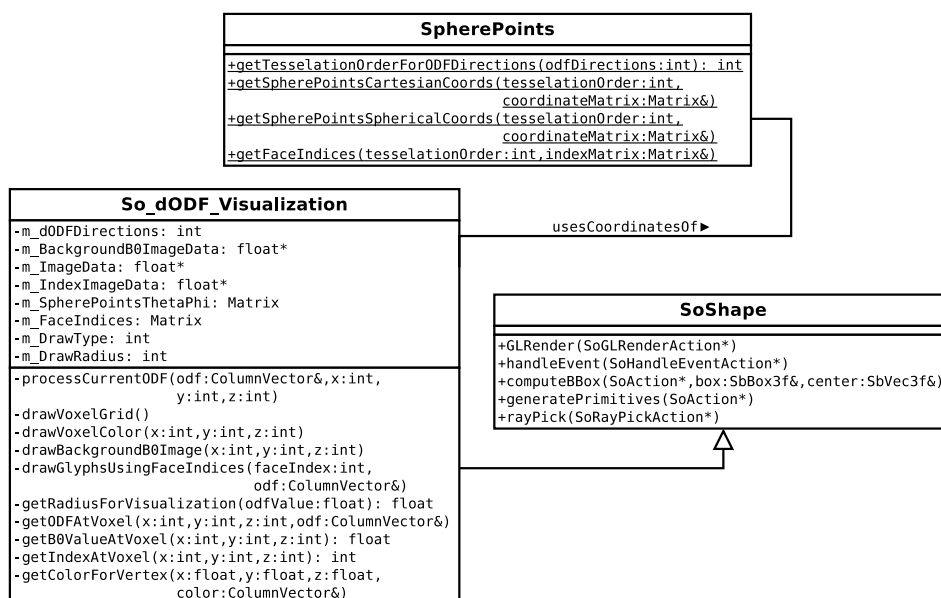


Figure 5.21: Class diagram of the *So_dODF_Visualization* module

5.3. DETAILED MODULE DESCRIPTION







Scalar value	Color name	Sample
0	not handled	-
1	dark green	
2	dark blue	
3	dark red	
4	dark cyan	
5	dark magenta	
> 5	dark yellow	

Table 5.2: The background colors for voxel classification that can be assigned to the voxels by a scalar value in the input image at the right input connector.

purpose of the module is the visualization of dODFs. An image containing the dODFs in dimension t can be connected to the middle input connector. The size of the dimensions c and u has to be 1.

The left and right connectors are optional and do not have to be connected to other modules in order for the *So_dODF_Visualization* module to work properly. At the right connector a voxel classification image can be supplied. The extension of the dimensions c , t and u of this image has to be 1. The value stored in each voxel is considered as an index value representing the class the voxel belongs to. Six different voxel classes can be defined. The corresponding colors are shown in table 5.2. These colors are drawn in the background of the dODFs. The value 0 is not handled and can be used as a rejection class. A practical example of the usage of voxel classes is shown in Figure 5.24a. The displayed classification was calculated by the *MFC_Macro* module. Dark green voxels represent voxels containing one fiber population while dark blue voxels contain two fiber populations. All other voxels were rejected, thus, containing no fiber populations (isotropic diffusion).

A B_0 -image can be connected to the left input connector. The extent of the dimensions c and u has to be 1. The extent of dimension t does not matter as only the value at $t = 0$ is used in this module. The value of every voxel at the coordinate $t = 0$ is interpreted as a grayscale value and is displayed in the background of the other two input images. This is a convenient way of displaying the B_0 -image in the background of the rendered scene. An example for the B_0 image in the background is shown in Figure 5.24b.

The output of this module is a rendered OpenGL scene that can be displayed with the *SoExaminerViewer* module.

The different visualization means offered by this module can be accessed via the GUI-panel (Figure 5.22). The *Icosahedron tessellation* tab is only for testing purposes. It allows to display the coordinates of the class *Sphere-*

5.3. DETAILED MODULE DESCRIPTION

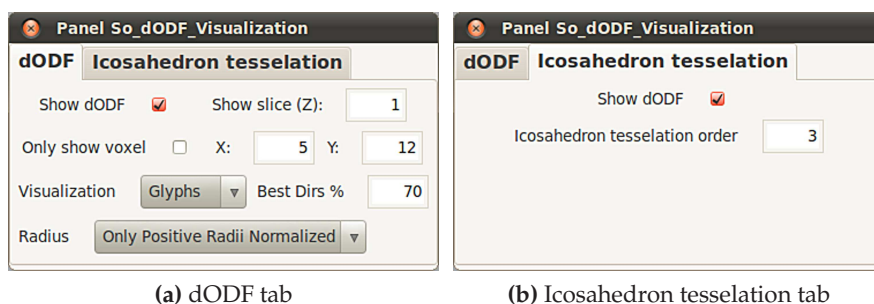


Figure 5.22: GUI-panel of the *So_dODF_Visualization* module

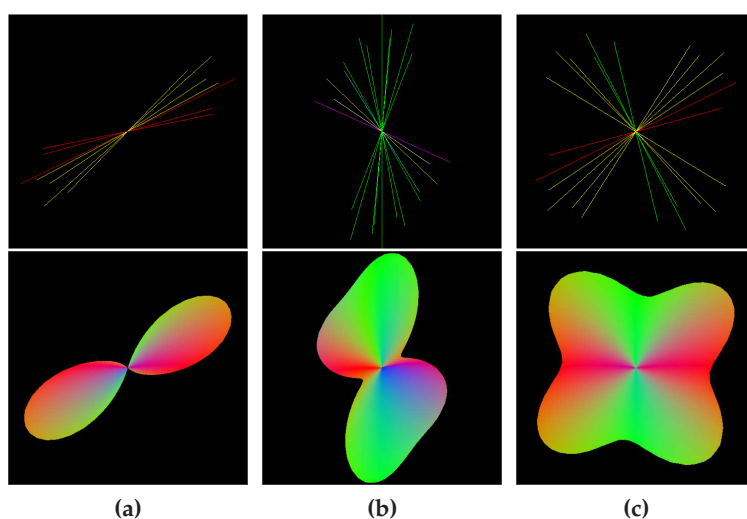
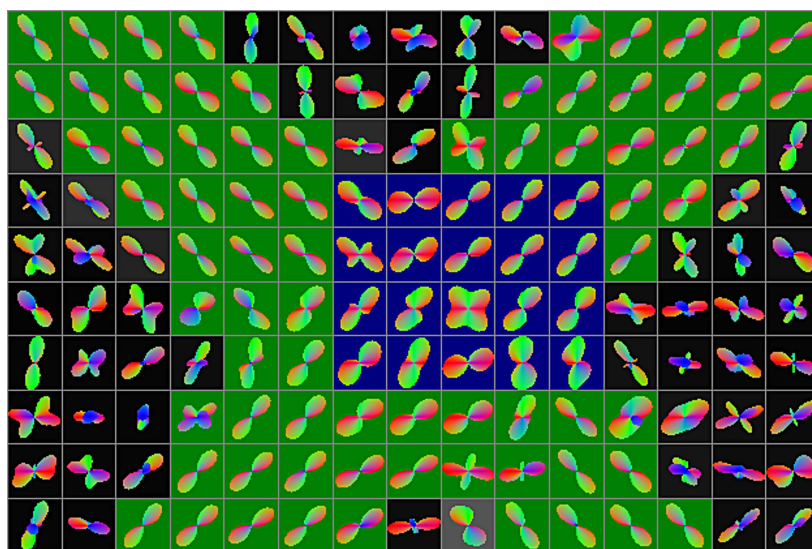


Figure 5.23: Diffusion ODF directions of the 3rd tessellation order having at least 70% of the PDD are shown for 3 different dODFs in the top row. The bottom row shows the same dODFs as glyphs.

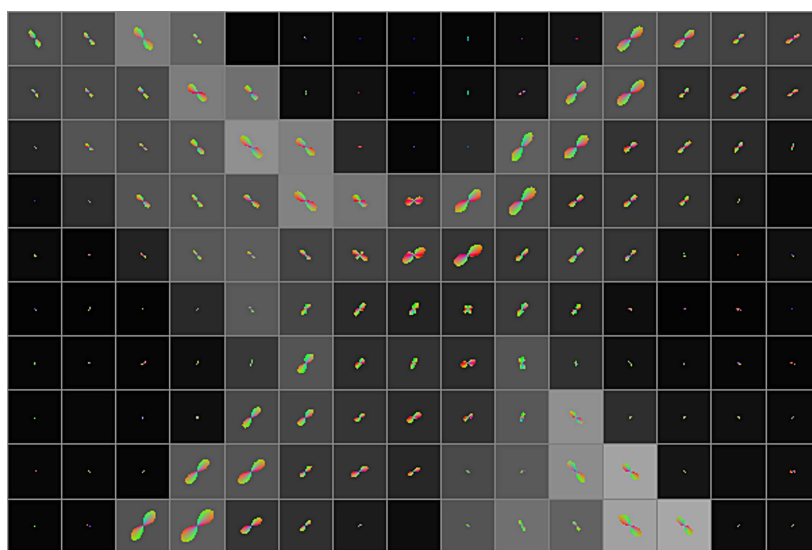
Points as a tessellated icosahedron. The box *Show dODF* has to be unchecked in order to view the icosahedron. The tessellation order can also be chosen in this tab. It only applies to the displayed icosahedron and does not affect the visualized dODFs.

The *dODF* tab allows for multiple ways of dODF visualization. The box *Show dODF* has to be checked in order to view the dODFs from the input image. To examine the data either one whole slice or a single voxel can be shown by checking and unchecking the box *Only show voxel*. The parameters *X* and *Y* hold the coordinates of the voxel to display. The slice can be chosen via the field *Show slice (Z)*.

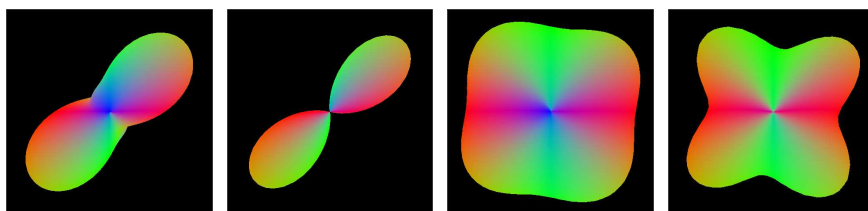
The dODFs can be displayed in several ways selected via the *Visualization* parameter in the GUI-panel. Possible visualization types are points, lines, meshes, or glyphs. Further, the visualization option *Best Dirs* allows to display only the highest values of the dODF as lines. The threshold for



(a) Fiber crossing shown as glyphs with voxel classification. The positive radii are normalized by the maximum value of each dODF.



(b) Fiber crossing shown as glyphs with B_0 -image. The positive radii are normalized by the maximum value in the ROI.



(c) Min-Max

(d) Pos-Max

(e) Min-Max

(f) Pos-Max

Figure 5.24: Visualization examples from the *So_dODF_Visualization* module

these values is determined by the *Best Dirs %* parameter. This parameter takes a value in the range $[0, 100]$. All dODF directions having at least this percentage of the maximum diffusion direction value in the dODF are shown. The bottom row of Figure 5.23 shows three different dODFs as glyphs (tessellation order 5). The 3rd tessellation order sampling directions with magnitude of at least 70% the PDD are shown in the top row for the same dODFs.

The last parameter that can be set via the GUI-panel is the *Radius*. This parameter represents the kind of normalization that is applied to the individual dODF values (these values are referred to as radii as they describe the local radius of the glyph in one direction). A common way to normalize the dODF values is the *Min-max normalization*. The normalized dODF radius, r_n , for a voxel u and each direction i is calculated as

$$r_n = \frac{\Psi(u)_i - \Psi(u)_{min}}{\Psi(u)_{max} - \Psi(u)_{min}} \quad (5.1)$$

where $\Psi(u)_i$ is the i -th direction vector of the dODF of the voxel u . $\Psi(u)_{max}$ and $\Psi(u)_{min}$ are the maximum and minimum values of the dODF of voxel u respectively. The resulting value, r_n , is in the range $[0, 1]$. This kind of normalization is straightforward but has a significant disadvantage. Negative dODF values (representing improbable diffusion directions) are also included in the normalization, thus, inflating the glyphs. An alternative method is to replace all negative values by a small value $\epsilon > 0$ before normalization (replacing by 0 would lead to undefined values in color calculation as each vector is scaled by the respective radius before its color is calculated). This method can be chosen via the option *Normalized By ROI Max* in the *Radius* parameter. A comparison of these two normalization methods is presented in Figures 5.24c through 5.24f for two different glyph shapes. A third option, *Normalized By ROI Max*, is also provided. Here, negative values are also replaced by a small value $\epsilon > 0$, but instead of using $\Psi(u)_{max}$ the maximum dODF value of all voxels in the dataset is used. The resulting glyphs are small in regions with low diffusivity and large in regions with high diffusivity. The fiber crossing area of the Fiber Cup dataset is presented in Figure 5.24a with positive normalized radii and in Figure 5.24b with ROI normalized radii for comparison. Further, *Zero* can be chosen for *Radius* in order to omit the dODF information. This is useful if only the information of the images at the left and right input connectors is desired.

The control flow of the *So_dODF_Visualization* module is illustrated in Figure 5.25. Initially, the sampling directions and the face indices for glyph rendering are copied from the class *SpherePoints* and the module waits for a module field change. As soon as valid input images are present at the modules input the data for each voxel is processed. Subsequently, a voxel grid is drawn, a voxel classification color is set (if the corresponding input

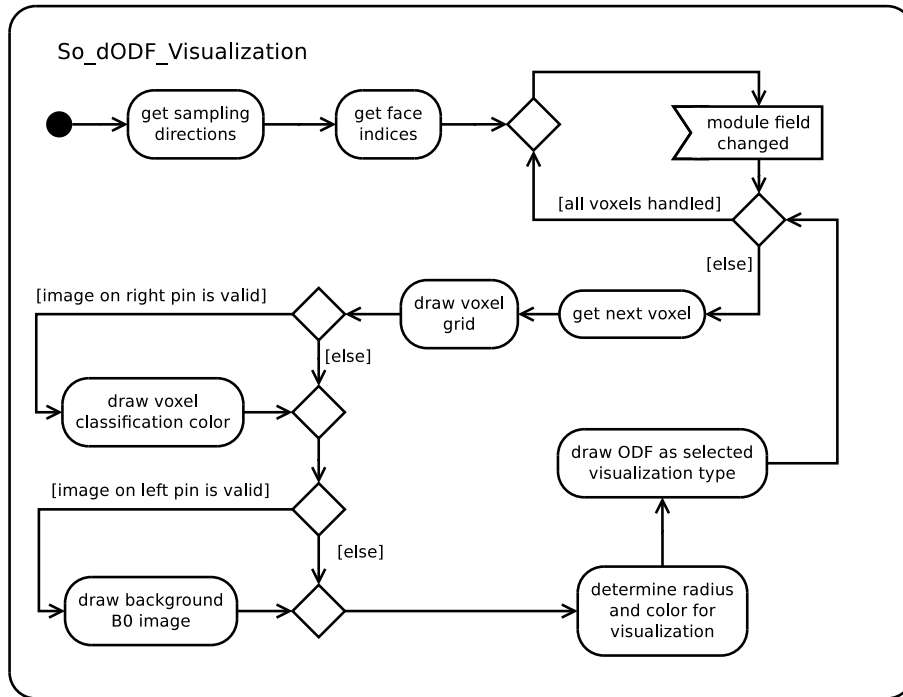


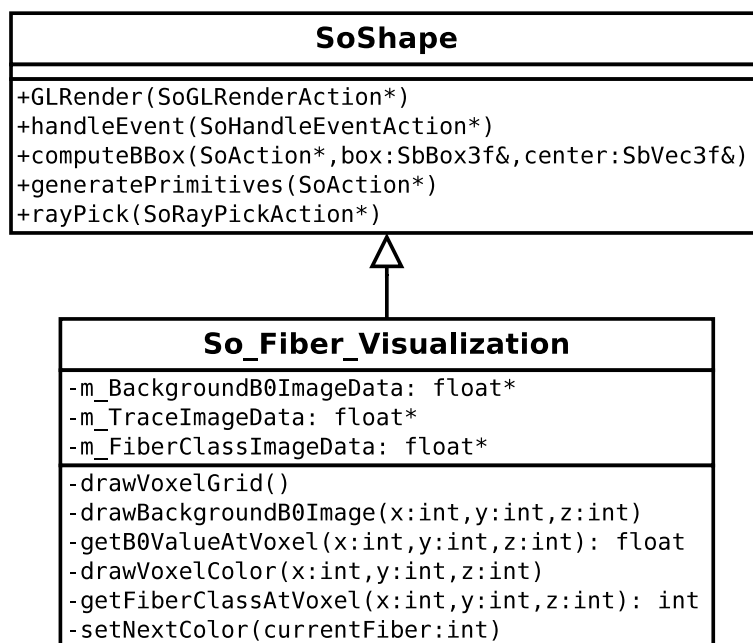
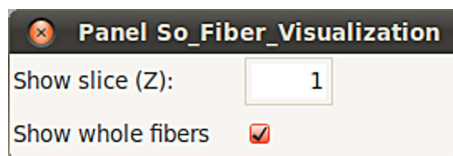
Figure 5.25: Illustration of the control flow in the *So_dODF_Visualization* module.

image is provided), and the background B_0 -image is drawn (again only if a B_0 -image is provided at the corresponding input connector). Then, the color and radius for visualization is determined and the dODF is rendered. After processing all voxels the module waits again for a field change.

5.3.6 So_Fiber_Visualization

The structure of the *So_Fiber_Visualization* module is illustrated in Figure 5.26. This module is a subclass of the class *SoShape*. The purpose of the *So_Fiber_Visualization* module is to visualize the fiber tracts that were reconstructed by the *HARDI_deterministic_Tractography* module.

This module offers three input connectors. The left and the right input connectors have exactly the same properties as in the *So_dODF_Visualization* module. The middle input pin is reserved for a fiber trace image. The extent of all dimensions of this image has to be 1. The only exception is the t dimension. It holds the complete information on the reconstructed fiber tracts. In a way, this input image is a one-dimensional array. The first value in dimension t indicates the number of vertices of the first fiber tract. The vertex coordinates are the subsequent values of dimension t . Then again a value indicating the number of vertices in the next fiber tract follows. For more details on the structure of the fiber trace image please refer to the

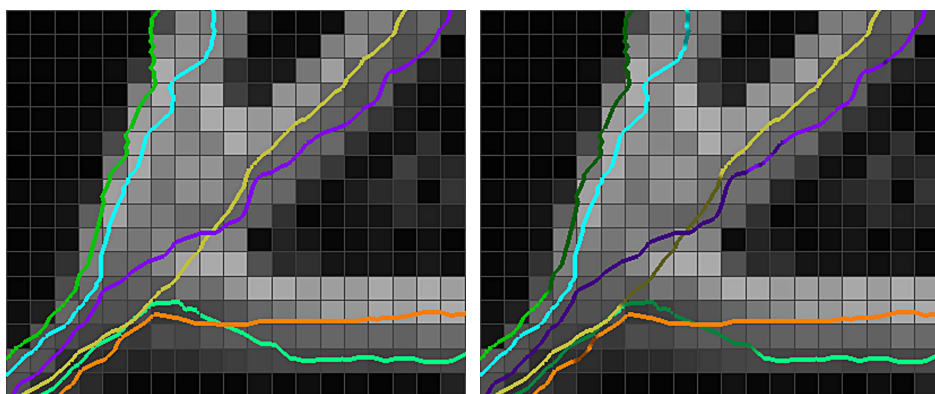
Figure 5.26: Class diagram of the *So_Fiber_Visualization* moduleFigure 5.27: GUI-panel of the *So_Fiber_Visualization* module

description of the *HARDI_deterministic_Tractography* module.

The output of this module is a rendered OpenGL scene that can be displayed with the *SoExaminerViewer* module.

The GUI-panel of this module offers access to two parameters (Figure 5.27). The first parameter, *Show slice (Z)*, allows to switch the displayed slice of the input images. Note that this parameter only influences the input images at the left and at the right input pin. The fiber tracts from the image at the middle input connector are rendered entirely no matter which slice is selected. This is done to be able to view the fibers as a whole and to better assess each individual fiber course. However, by unchecking the parameter *Show whole fibers* only the parts of the fiber trajectories that are in the selected slice are rendered in a bright color. The other parts of the fibers are displayed in a dark color. Example images visualizing the reconstructed fiber pathways are presented in Figure 5.28.

Figure 5.29 illustrates the control flow in the *So_Fiber_Visualization* module. On startup, the module waits for a module field change. When a valid



(a) Parameter *Show whole fibers* is checked, fibers are rendered completely in a bright color. (b) With unchecked parameter *Show whole fibers* only the parts of the fibers in the selected slice are displayed in a bright color.

Figure 5.28: Visualization of reconstructed fibers.

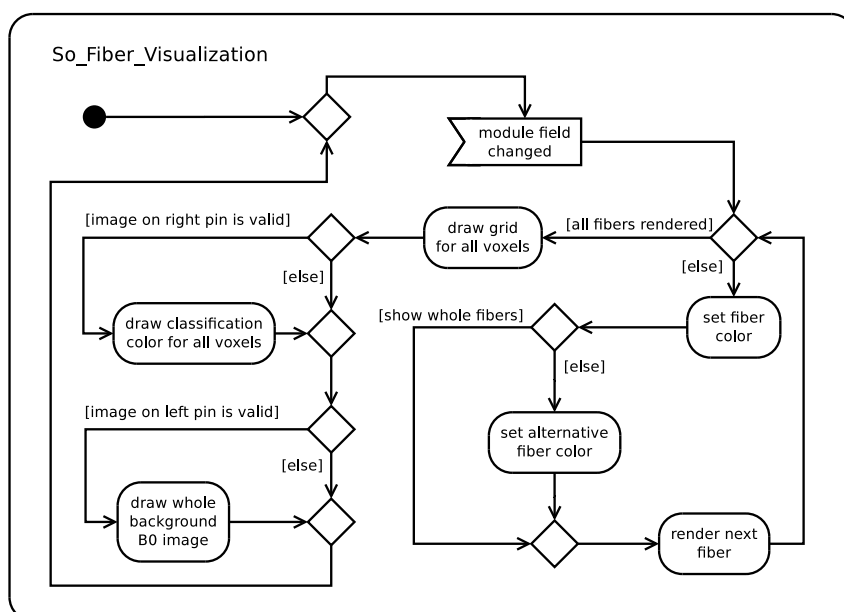


Figure 5.29: Illustration of the control flow in the *So_Fiber_Visualization* module.

input images are provided a color for the first fiber is selected. If *Show whole fibers* is unchecked the same color with half the brightness is used to render parts of the fiber running through other slices. As soon as all fibers are displayed a grid separating the voxels is drawn. Subsequently, if the corresponding input images are provided a classification color is set for all voxels and the background B_0 -image is drawn.

Chapter 6

Evaluation

The objective of this chapter is to evaluate the ML modules described in the previous chapter. As these modules differ strongly in their functionality and purpose different aspects will be evaluated. The evaluation is performed on a Linux notebook with an *Intel Core 2 Duo P8400*¹ CPU @ 2.26 GHz. This notebook is equipped with 2 GiB of RAM.

The evaluation is performed both, on a phantom and on a human brain dataset. The phantom was originally provided by the Laboratoire de Neuroimagerie Assistée par Ordinateur (LNAO, France) for the Fiber Cup, a tractography contest at the MICCAI conference in 2009². The phantom data was acquired with two repetitions and 64 image encoding gradients uniformly distributed over the sphere. The data volume consists of 3 slices each with a resolution of 64×64 voxels with an uniform voxel size of 3 mm. Of the different diffusion sensitizations provided the dataset with b-value $b = 2000 \text{ s/mm}^2$ is used here. No ROI selection is applied to the data, i.e. the whole data volume will be used for evaluation. All measurements of calculation times were performed on the Fiber Cup dataset.

The human brain dataset is taken from [PPAM06]. It consists of 60 slices each with a resolution of 128×128 voxels. The in-slice voxel size is 1.875×1.875 mm, the slice thickness is 2 mm. The acquisition was performed with 41 image encoding gradients at a b-value of $b = 700 \text{ s/mm}^2$. For the evaluation in this chapter a ROI with the size of $43 \times 7 \times 40$ voxels was selected in the region of the corpus callosum (CC) including the corticospinal tract (CST).

6.1 HARDI_dODF_Reconstruction

This module has the purpose to reconstruct the dODF from SH coefficients. The quality and accuracy of this reconstruction algorithm as described in

¹<http://ark.intel.com/Product.aspx?id=35569>

²<http://www.lnao.fr/spip.php?rubrique79>

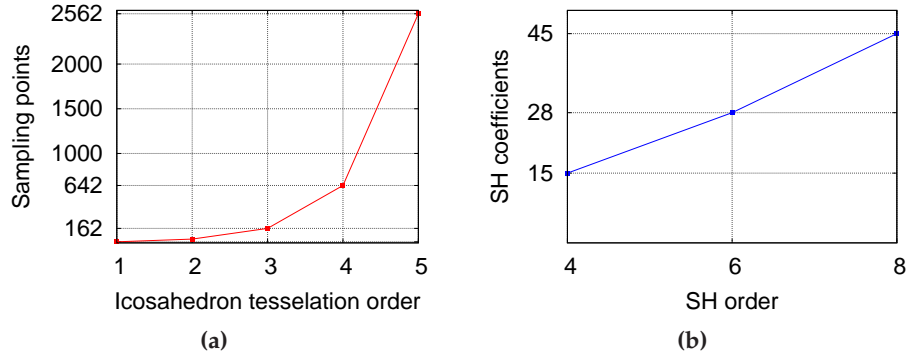


Figure 6.1: Number of sampling points and SH coefficients at different orders

Tessellation order	Reconstruction time [s]		
	SH order 4	SH order 6	SH order 8
1	1	2	16
2	4	7	25
3	14	27	57
4	55	103	184
5	219	411	688

Table 6.1: Durations of dODF reconstruction for different parameters

section 3.2.3 was evaluated by its authors in [DAFD07]. Therefore, only the computation time of the presented implementation will be evaluated in this section.

Two aspects influence the computation time of this module:

- the tessellation order of the icosahedron
- the order of the provided SH coefficients

These two parameters have different influences on the complexity of dODF reconstruction. Figure 6.1a shows how the number of sampling directions increases with higher tessellation orders (this Figure is a plot of table 5.1). This exponential growth is also expected for the computation time of the dODFs. Further, Figure 6.1b shows how the number of coefficients increases with higher SH orders. Although the number of SH coefficients does not increase as rapidly as the number of sampling points it has nevertheless a strong effect on the computation time as each coefficient needs to be multiplied with a SH basis function for each sampling point (equation 3.15).

The duration of a dODF reconstruction for different parameters are presented in Figure 6.2 and table 6.1. As expected, the reconstruction time

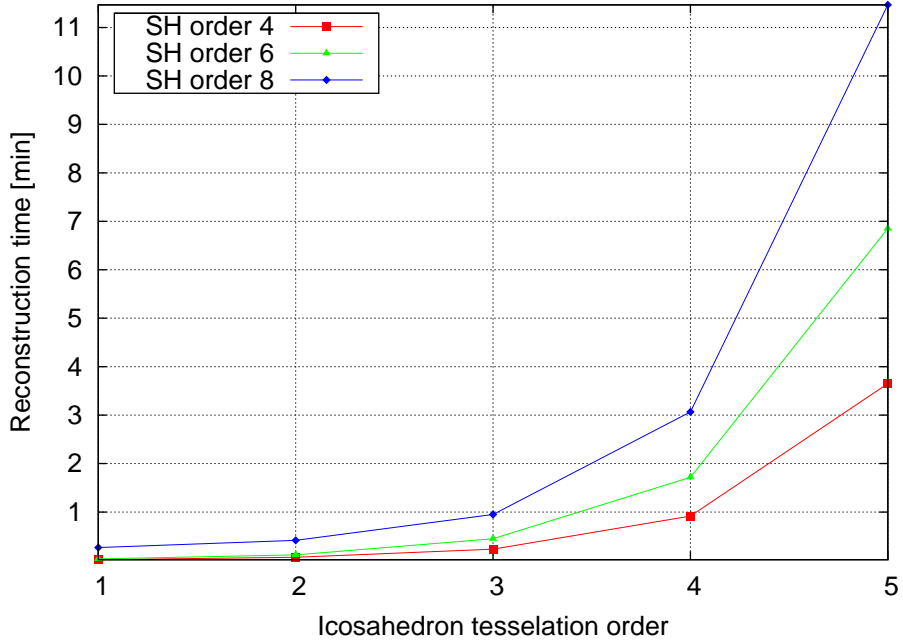


Figure 6.2: Illustration of dODF reconstruction durations

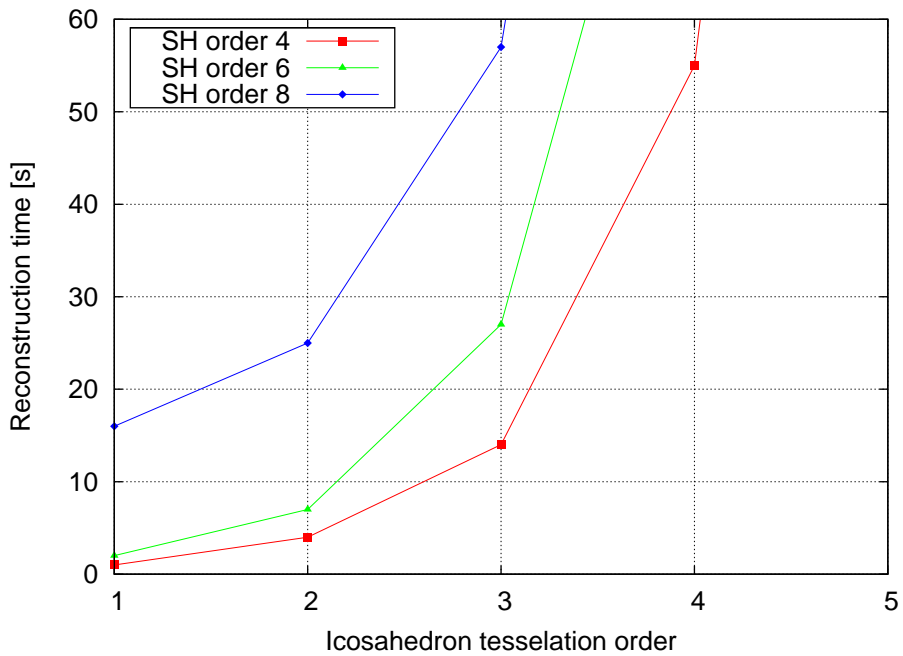


Figure 6.3: Illustration of dODF reconstruction durations in the time span of one minute

increases exponentially. Figure 6.3 shows the parameter combinations that lead to clinically feasible reconstruction times of one minute and less. While the reconstruction time for all evaluated SH orders for tessellation order 3 lies in this period of time dODF with tessellation order 4 can only be reconstructed from SH coefficients of order 4 in the same time span. The reconstruction time of approximately 3 minutes for SH order 8 and tessellation order 4 can still be considered as acceptable. The employment of higher SH orders is currently not practical as noise disturbances become an issue. Further, Figure 5.9 raised the suspicion that the additional precision arising from tessellation order 5 compared to orders 3 and 4 is marginal. However, the impact of the chosen tessellation order on the accuracy of fiber tracing needs to be evaluated. Even if this precision should be needed dODF with tessellation order 5 and SH order 4 is sufficient to identify up to 2 fiber populations per voxel and can be reconstructed in a time period far below 4 minutes.

For further tests tessellation orders 1 and 2 will not be used as they are practically useless.

6.2 HARDI_Anisotropy_Criteria

The *HARDI_Anisotropy_Criteria* module allows to calculate anisotropy criteria introduced in section 4.1.3. The evaluation is performed in two steps. In the first section the computation time for different parameters will be examined. The second section evaluates the quality of the module's classification results.

6.2.1 Computation Times

Computation times for several different tasks are relevant in the *HARDI_Anisotropy_Criteria* module. These tasks are:

- Precalculation of data needed for dODF based criteria
- Calculation of output data for a dODF based criterion
- Calculation of output data for a SH coefficients based criterion
- Calculation of output data for the MDL criterion

The precalculation of data needed for dODF based criteria is performed when an input image or the *Max Radius* parameter in the criterion *Vesna* changes. This computation depends only on the tessellation order of the dODF and not on the SH order. The times needed to precompute the dODF are illustrated in Figure 6.4 for different tessellation orders. Again, tessellation order 5 demands for very long computation times whereas the other orders lie in an acceptable time span.

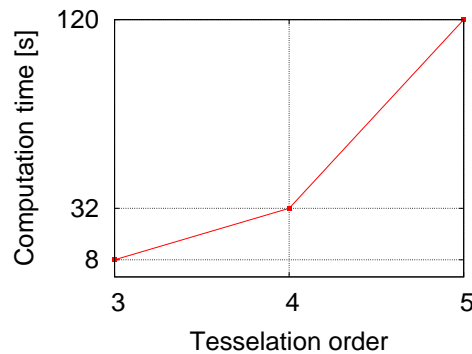


Figure 6.4: Illustration of anisotropy criteria precalculation durations

SH order	Computation time [s]
4	0.5
6	1
8	16

Table 6.2: Durations of MDL criterion calculation for different SH orders

Once the data for dODF based criteria (i.e. *GFA*, *stdDev*, *Vesna*), is pre-computed, changing thresholds or selecting another criterion hardly needs any time. For any of these criteria this computation time is in the order of 10 ms independent of the tessellation order or the used SH order.

Criteria completely (i.e. *FMI*, *MDL*) or partly (i.e. *Chen*) based on the SH coefficients are computed on demand and are, except for *Chen*, independent of the tessellation order. Calculating an output image for *FMI* and *Chen* takes less than 30 ms for SH order 3, and less than 50 ms for SH order 4. However, this time strongly increases for SH order 8 and is on the order of 14 seconds.

The *MDL* criterion has the highest calculation times. These times are presented in table 6.2.

6.2.2 Classification on the Fiber Cup Phantom

The classification results of this module are evaluated on the Fiber Cup phantom dataset in this section. All thresholds were chosen carefully to achieve the best possible classification results. The B_0 -image showing the middle slice of the Fiber Cup phantom is shown in Figure 6.5. Bright voxels represent anisotropic diffusion while dark voxels represent isotropic diffusion. An anisotropy criterion is considered as sound if it is able to separate these two voxel classes properly.

The presented classification results show the middle slice of the dataset. Green voxels represent anisotropic diffusion with one fiber population. Blue

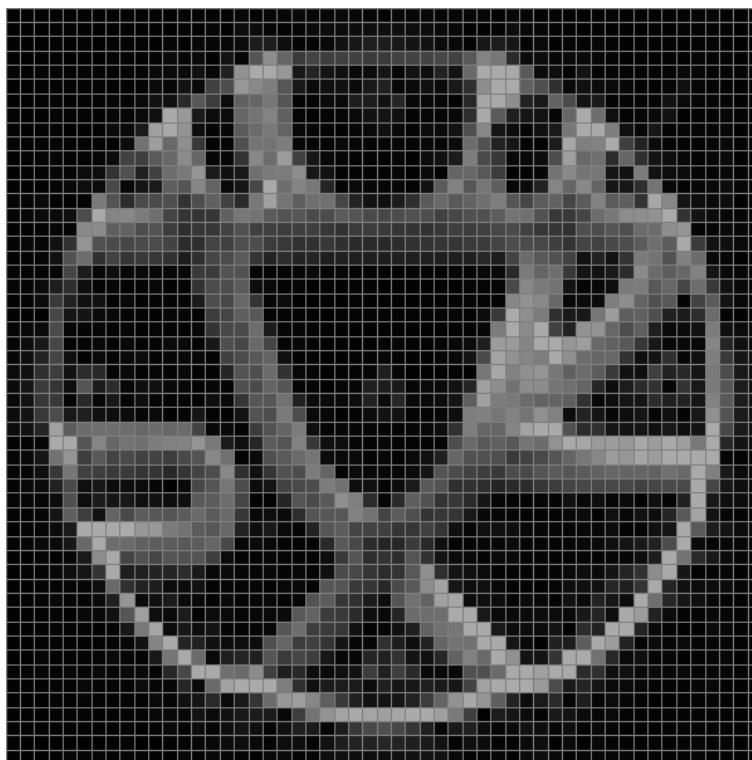


Figure 6.5: B_0 -image of the Fiber Cup phantom

voxels stand for anisotropic diffusion with two fiber populations per voxel. Each row of the following Figures shows the results for a distinct SH order (SH). Each column illustrates results for a tessellation order (T).

GFA

Figure 6.6 shows the classification results of the *GFA* criterion for all relevant tessellation orders and SH orders. The classification results are overlaid on the B_0 -image from Figure 6.5. *GFA* classifies most of the white matter voxels as anisotropic. However, even with a carefully chosen threshold a lot of gray matter voxels are classified as anisotropic as well. Further, for higher SH or tessellation orders no improvement in the results can be found. One possible explanation for this bad results is that the examined dataset is of low quality. It should be noted again that Q-ball reconstruction works best on datasets acquired with a b-value of at least 3000. However, a maximum b-value of only 2000 is available for this dataset. Further, the data is very noisy and shows only little differences between isotropic and anisotropic voxels. In conclusion, the *GFA* criterion is useless on the Fiber Cup phantom.

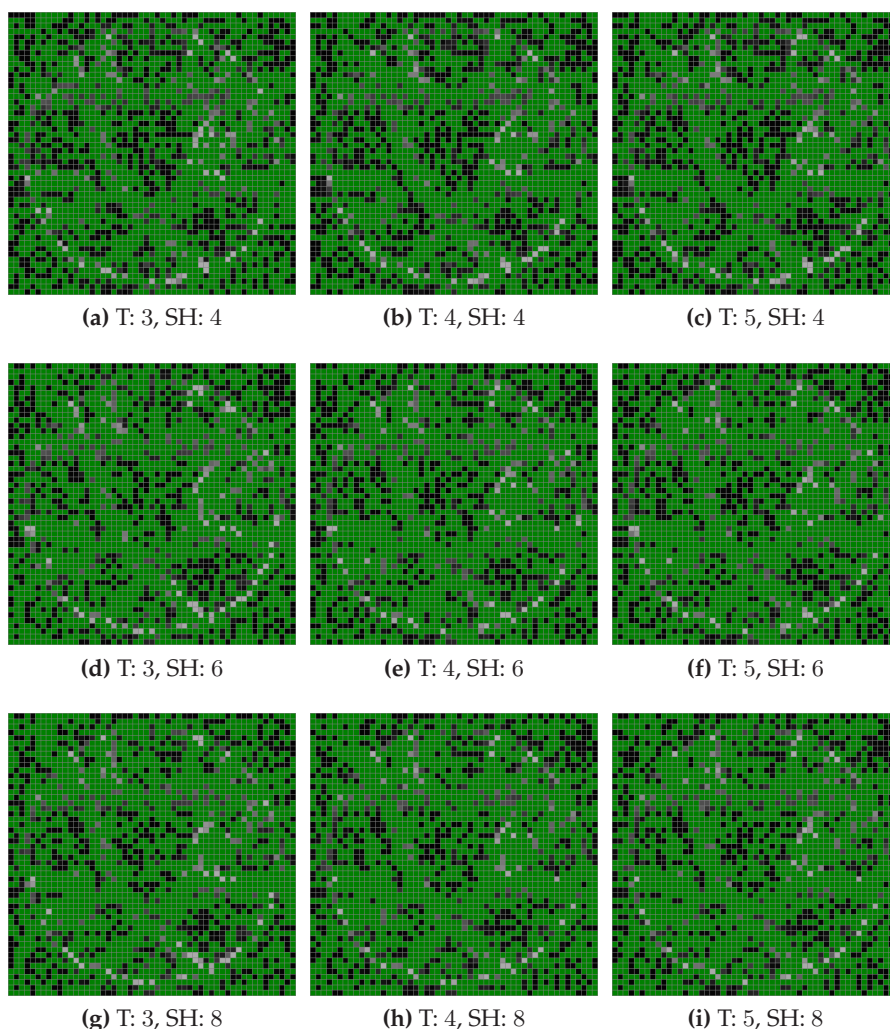


Figure 6.6: Classification results of the *GFA* criterion (threshold 0.55).

stdDev

Figure 6.7 shows the classification results of the *stdDev* criterion. This criterion is able to separate gray and white matter voxels very well as can be seen by comparing the results to the B_0 -image. The results for different tessellation orders show almost no differences. Further, a marginal increase of erroneously classified gray matter voxels as having anisotropic diffusion can be observed for higher SH orders. This observation is consistent with the fact that higher SH orders are more susceptible to noise. Overall, these results are promising and can be used as a mask to separate gray and white matter voxels. The choice of a proper threshold, however, is more time-consuming as the standard deviation can take arbitrary values and is not restricted to a certain range (as e.g. is the *GFA*).

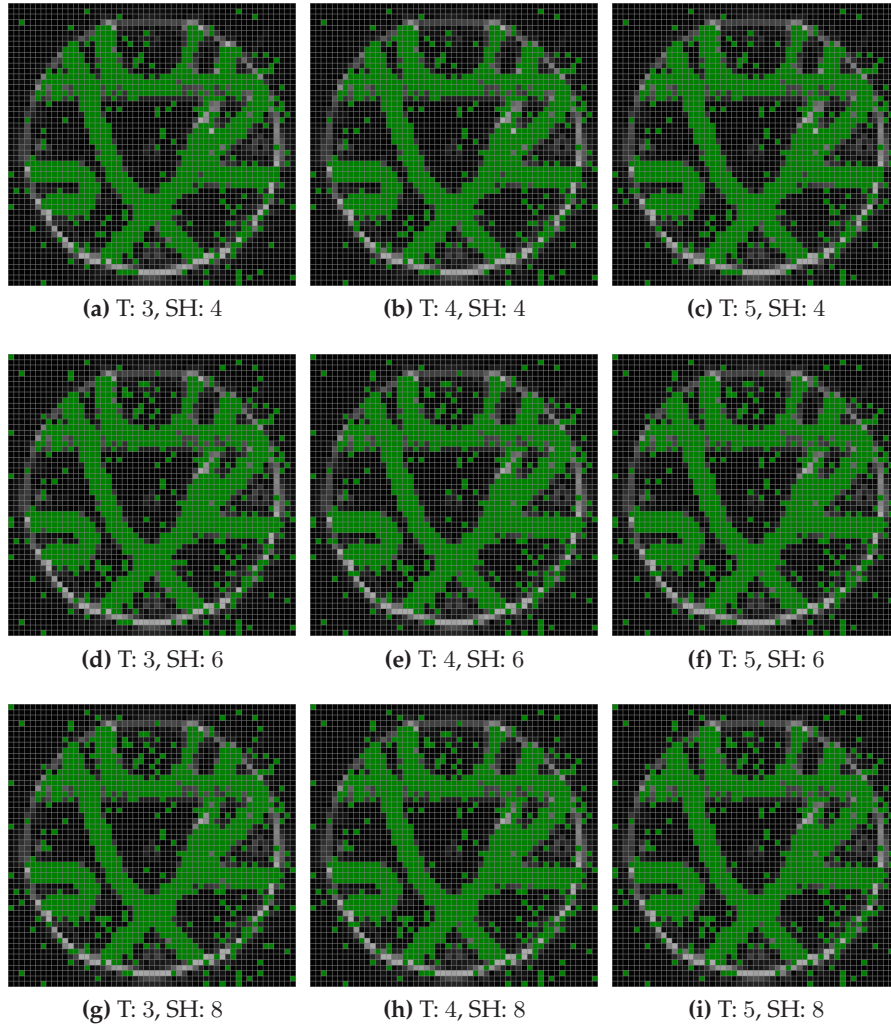


Figure 6.7: Classification results of the *stdDev* criterion (threshold 2.67).

Vesna

Figure 6.8 shows the classification results of the *Vesna* criterion. The chosen thresholds were 0.6 (*Max Radius*), 0.023 (*One Fiber Threshold*), 0.026 (*Two Fiber Threshold*) for Figures 6.8d and 6.8g. Further, 0.6 (*Max Radius*), 0.1 (*One Fiber Threshold*), 0.11 (*Two Fiber Threshold*) for Figures 6.8e and 6.8h. The thresholds 0.6 (*Max Radius*), 0.37 (*One Fiber Threshold*), 0.4 (*Two Fiber Threshold*) were applied to the remaining Figures. The classification results of voxels with two fiber populations (blue) were useless for all configurations. As in the last criterion, no significant differences can be observed when varying the tessellation order while the number of erroneously classified voxels increased with the SH order. The results are not as bad as of the *GFA* criterion, but still too noisy to be used in practice.

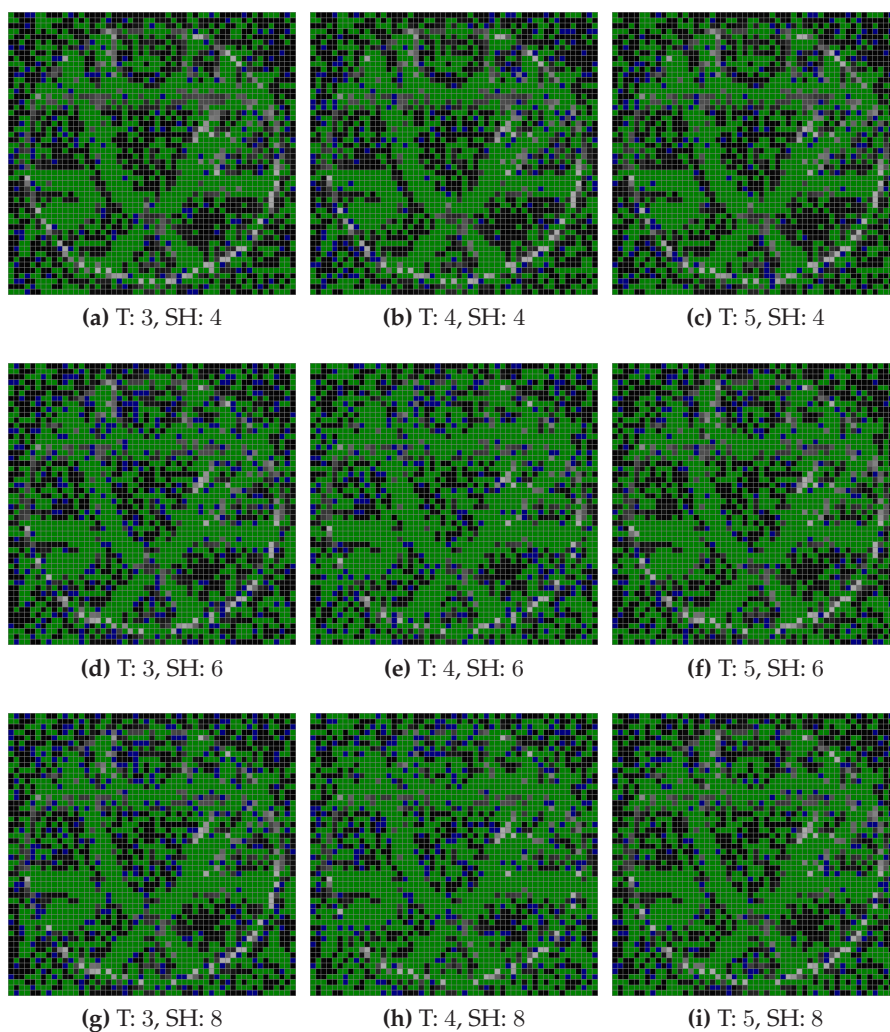


Figure 6.8: Classification results of the *Vesna* criterion.

Chen

Figure 6.9 shows the classification results of the *Chen* criterion. Again, different thresholds had to be chosen here to achieve similar results throughout all configuration. In general, the chosen thresholds were 0.877 ($R0$), 0.5 ($R2$), 0.1 ($R4$), 3.5 (*Low Variance*), and 6 (*High Variance*). For SH orders 6 and 8 $R4$ was modified to 0.857. Similar to the *Vesna* criterion the results for two fiber populations were useless. As before, no differences in the results for varying tessellation orders could be observed and noise increased with the SH order. However, when only considering the green marked voxels the results can be useful to separate gray and white matter voxels.

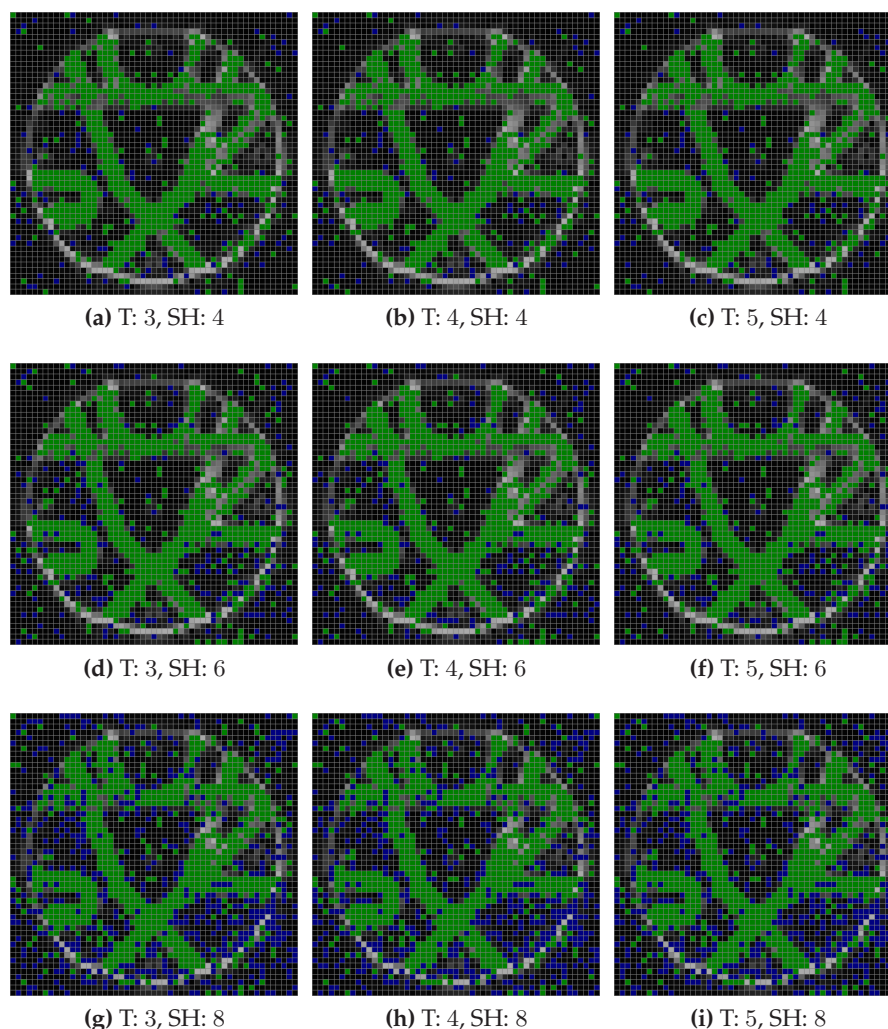


Figure 6.9: Classification results of the *Chen* criterion.

FMI

The *FMI* criterion only depends on the choice of the SH order. Figure 6.10 presents its results. The results for two fiber populations were useless as in all other criteria. On the other hand, voxels with one fiber population were identified well. The choice of higher SH orders introduced only marginal increase in noise but did not allow for better classification results.

MDL

The results of the *MDL* criterion appear like randomly distributed colors (Figure 6.11). In theory, *MDL* can detect a maximum number of fiber populations per voxel corresponding to $l/2$ where l is the chosen SH order. Prac-

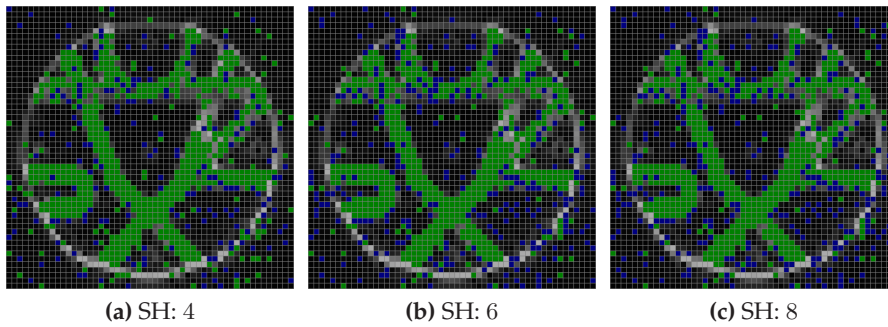


Figure 6.10: Classification results of the *FMI* criterion (thresholds: 0.0025 (*FMI 0*), and 0.25 (*FMI 1*)). Green voxels represent one fiber population, while blue voxels represent two fiber populations.

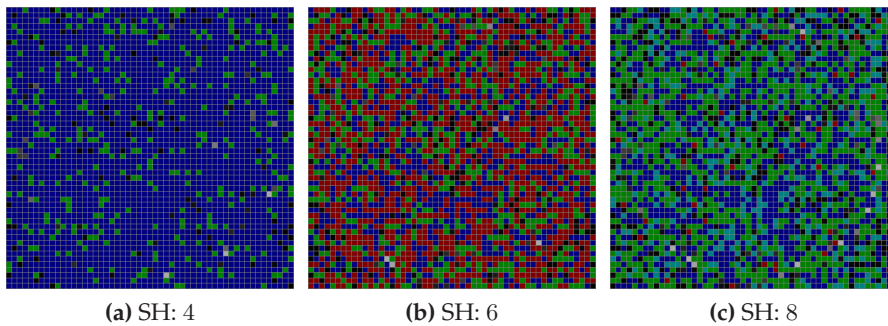


Figure 6.11: Classification results of the *MDL* criterion. Fiber populations: 1 (green), 2 (blue), 3 (red), 4 (dark cyan).

tically, the results are useless. These bad results are not surprising as this criterion was, so far, only tested on simulated data where all parameters (e.g. noise, SNR) were adjustable.

Summary of Results on Phantom Data

Higher tessellation or SH orders did not provide better results on this dataset for any of the criteria. The results of the criteria *GFA*, *Vesna*, and *MDL* were either completely useless or too noisy to be used in practice. None of the four criteria theoretically capable of classifying voxels with two fiber populations was able to provide useful results. However, the criteria *stdDev*, *Chen*, and *FMI* are able to separate isotropic and anisotropic voxels very well. Of all these criteria the selection of a proper threshold is easiest for the *stdDev* criterion as only one single threshold has to be found.

6.2.3 Classification on Human Brain Data

The classification results of the *HARDI_AnisotropyCriteria* module are evaluated on the human brain dataset in this section. Only the criteria that provided best results on the phantom dataset are used for this evaluation. The b-value of 700 is challenging as it is too low for optimal Q-ball reconstruction. However, as no HARDI datasets with higher b-values are available the evaluation is carried out on this dataset.

stdDev

Figure 6.12 shows the classification results of the *stdDev* criterion for some combinations of tessellation and SH orders. The applied threshold is 260. Although many voxels were erroneously classified as anisotropic the separation of gray and white matter voxels is acceptable. Selecting higher SH or tessellation orders leads to marginal changes in the classification, but does not improve the results.

Chen

The classification results of the *Chen* criterion are presented in Figure 6.13. The chosen thresholds were 0.9 (*R0*), 0.5 (*R2*), 0.06 (*R4*), 73000 (*Low Variance*), and 80000 (*High Variance*). The threshold *R4* was modified to 0.1 for Figure 6.13c. White and gray matter voxels were separated with slightly less false positives than with the *stdDev* criterion. Again no significant changes can be observed for different SH or tessellation orders.

FMI

Figure 6.14 illustrates the classification results of the *FMI* criterion for SH orders 4 and 6. The applied thresholds were 0.003 (*FMI 0*) and 0.4 (*FMI 1*). The classification results for both SH orders are almost identical. White and gray matter separation works well with this criterion.

Summary of Results on Human Brain Data

The classification results of these three criteria on the human brain dataset confirm that selecting higher SH and tessellation orders does not improve the classification results. All criteria were able to separate isotropic and anisotropic voxels well. The *stdDev* results were the best on phantom data. However, on the human brain data all three criteria perform similar. The classification results of the *Chen* and *FMI* criteria for voxels with two fiber populations were useless as before on phantom data.

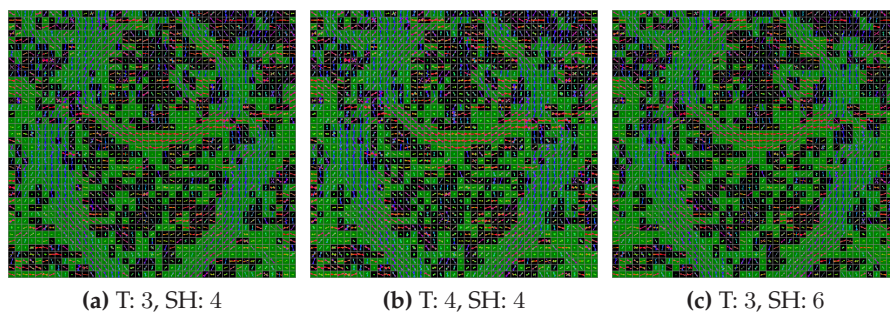


Figure 6.12: Classification results of the *stdDev* criterion on human brain data (threshold 260) for some combinations of tessellation (T) and SH orders (SH).

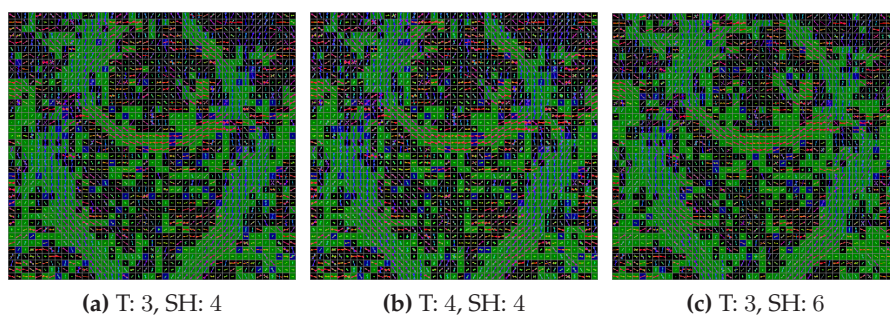


Figure 6.13: Classification results of the *Chen* criterion on human brain data for some combinations of tessellation (T) and SH orders (SH).

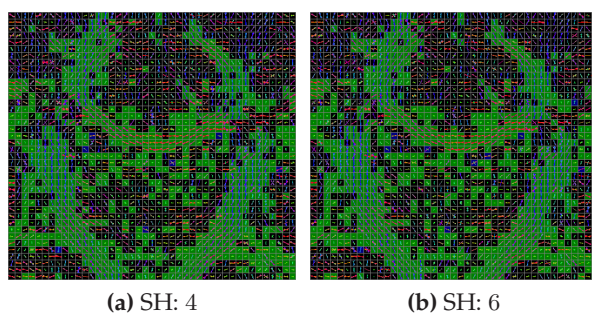


Figure 6.14: Classification results of the *FMI* criterion on human brain data for SH orders 4 and 6. Green voxels represent one fiber populations while blue voxels stand for two fiber populations.

6.3 MFC_Macro

This module was developed to improve the classification results of voxels with two fiber populations as the established criteria (*FMI*, *Chen*) provided useless results. The *MFC* computations are based on an mask separating gray and white matter voxels obtained from one of the criteria in the *HARDI_Anisotropy_Criteria* module. As only the criteria *Chen*, *FMI*, and *stdDev* perform this separating task well only these three criteria will be evaluated here. This evaluation is limited to SH order 4 and tessellation order 3.

MFC on Phantom Data

Figure 6.15 shows the classification results for different input masks on phantom data. Voxels marked blue were classified by this module as containing two fiber populations. When not stated differently the same thresholds were applied as in the evaluation of the *HARDI_Anisotropy_Criteria* module. Figures 6.15a through 6.15c show the results obtained from the *stdDev* criterion for different thresholds. In Figure 6.15c all relevant areas of fiber crossing, fiber kissings, and fiber branchings were identified. For lower thresholds *MFC* merges identified regions or even produces false positives. The results from the *FMI* and *Chen* white matter mask are not as good as of the *stdDev* mask, but still identify lots of relevant areas correctly.

Although some areas were identified as too small (especially the fiber crossing region in the lower part of the phantom) these results improve the classification of multiple fiber populations significantly.

MFC on Human Brain Data

The *MFC* results on human brain data are presented in Figure 6.16. Same threshold as in the evaluation of the *HARDI_AnisotropyCriteria* module on human brain data were used for the *stdDev* and *Chen* criteria. The thresholds of the *FMI* criterion were modified to 0.0026 (*FMI* 0) and 0.4 (*FMI* 1).

Ideally, only the parts where the corpus callosum merges into the corticospinal tract and the left and right branchings of this tract would be classified as containing two fiber populations. Lots of false positives can be found in the classification results of all three criteria. The good classification results on phantom data were due to similar fiber tract widths in the whole dataset. However, this is not true for the human brain data. One possible solution would be to use different mask sizes for the morphological operations and combine the results.

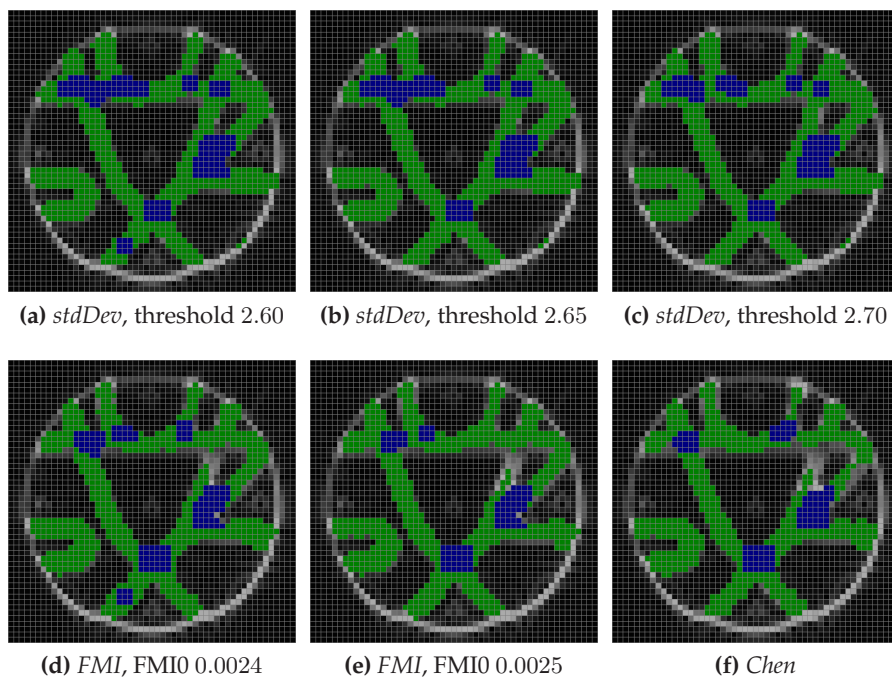


Figure 6.15: MFC classification results on phantom data

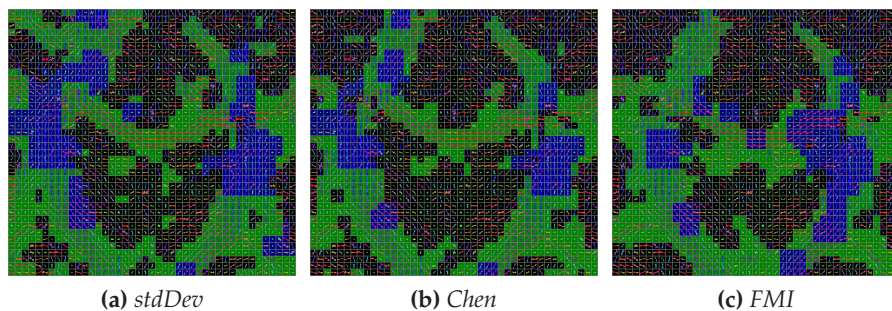


Figure 6.16: MFC classification results on human brain data

6.4 HARDI_deterministic_Tractography

This section evaluates different tractography algorithms that were implemented in the course of this thesis for the *HARDI_deterministic_Tractography* module. In the first section the fiber tracing times of each algorithm will be examined. The second section evaluates the quality of the obtained results.

6.4.1 Fiber Tracing Durations

The times needed by the different algorithms for fiber reconstruction are presented in table 6.3. The shown times were averaged for all different parameter configurations plotted in the next section. These times highly depend on the lengths of fibers that were recovered (i.e. an algorithm that aborts tracing after a few steps needs little time even if it is computationally expensive). The *Weighted Dirs Angle* algorithm stopped many trajectories too early resulting in the lowest computation time. All other tracing times can be considered as representative for the individual algorithm. The significantly higher computation time of the *Distance Based* algorithm indicates its high complexity. However, 16 seconds are still acceptable in practice.

Algorithm	Reconstruction time [s]
Euler PDD	2.1
RK4 PDD	3.4
Weighted Dirs Angle	1.4
Weighted Dirs MFC	2.3
Distance Based	16.4

Table 6.3: Times needed for fiber tract reconstruction by different algorithms.

6.4.2 Tractography Results on Phantom Data

To evaluate the tractography results common seedpoints and ground truth were provided for the Fiber Cup phantom (Figure 6.17). The phantom itself represents the most challenging fiber configurations: Fiber crossings at different angles, fiber branching, and a fiber kissing, as well as a sharp turn were designed. The seedpoints were placed in the middle slice of the dataset. An application was provided at the Fiber Cup website to evaluate the results. This application requires the coordinates of a fiber ground truth and the coordinates of the corresponding candidate fiber as input (in form of a *txt*-file). The fiber trajectories reconstructed by the *HARDI_deterministic_Tractography* module were resampled using interpolating coordinates with the application provided for evaluation. These resampled coordinates were the input data for the evaluation process.

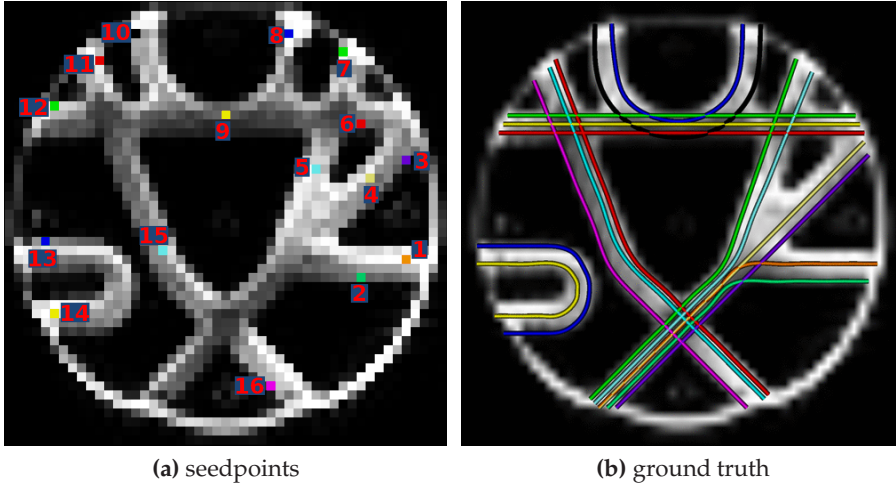


Figure 6.17: Seedpoints and ground truth of the Fiber Cup phantom

In the evaluation process each fiber was resampled with 1000 points and these points were approximated with splines. The output of the application is the mean difference between the ground truth fiber and the candidate fiber computed at each point according to a selected metric. Three metrics were used for evaluation: $l2$, tan , and $curv$.

The $l2$ metric is the squared euclidean distance between two points

$$l2(\mathbf{g}, \mathbf{c}) = g_x \cdot c_x + g_y \cdot c_y + g_z \cdot c_z \quad (6.1)$$

where \mathbf{g} is the ground truth point and \mathbf{c} the corresponding point of the candidate fiber.

The tan metric is the squared angle in degrees between the tangent vectors to the two evaluated points

$$tan(\mathbf{g}, \mathbf{c}) = (\arccos(\mathbf{g}_o \cdot \mathbf{c}_o))^2 \quad (6.2)$$

where \mathbf{g} and \mathbf{c} are defined as before and g_o and c_o are the vectors from the origin to these points.

The $curv$ metric measures the curvature of the spline resulting from interpolated fiber coordinates

$$curv(\mathbf{g}, \mathbf{c}) = \left(\frac{\|\mathbf{vder}_g \times \mathbf{vder2}_g\|}{\|\mathbf{vder}_g \times \mathbf{vder2}_g\|^3} - \frac{\|\mathbf{vder}_c \times \mathbf{vder2}_c\|}{\|\mathbf{vder}_c \times \mathbf{vder2}_c\|^3} \right)^2 \quad (6.3)$$

where \mathbf{vder}_x is the vector from the origin to the value of the first derivative and $\mathbf{vder2}_x$ is the vector from the origin to the value of the second derivative of the spline at point x .

The $l2$ metric reliably indicates whether or not a fiber trajectory is close to the ground truth. The other two metrics, however, provide bad results

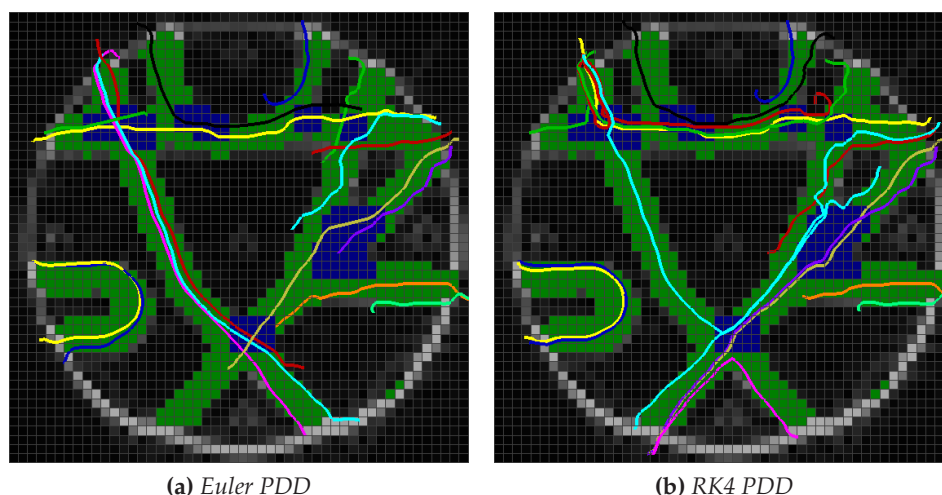


Figure 6.18: Tracing results of the *Euler PDD* and *RK4 PDD* algorithm for a curvature threshold of 35 degrees in voxels with two fiber populations.

for fiber trajectories that are not smooth or not perfectly straight lines even if they are very close to the ground truth. Further, if each fiber trajectory is approximated by straight lines representing the PDD of the corresponding seedpoint and completely disregarding the underlying fiber data the results of *tan* and *curv* are still very good. Thus, an evaluation based on these three metrics lays more weight on smoothness than on the spatial correctness of the fiber trajectories. Nevertheless, all metrics will be used for evaluation to allow for a better comparison to the Fiber Cup results.

For a better comparison of the different algorithms the *y*-axes of the plots are scaled equally for one metric over all algorithms. *RMS* in the caption of the *y*-axes stands for *root mean square*. SH order 4 and tessellation order 3 was used for all presented approaches.

Euler PDD

Quantitative experiments were performed to find the optimal step size and curvature threshold for voxels with one fiber population. The experiments consisted of visually inspecting the tracing results and comparing the fiber trajectories to the ground truth. The step size and curvature threshold with the most correctly reconstructed streamlines (i.e. matching the ground truth in their course) were used for the evaluation. The best results were achieved with an integration step size of 0.25 voxels (0.75 mm) and a curvature threshold of 60 degrees. Increasing the curvature threshold further did not lead to any changes in the reconstructed fiber trajectories.

Using the fixed step size and curvature for one fiber populations the curvature threshold for voxels with two fiber populations was varied. Fig-

ure 6.18a shows the reconstructed fibers for a two fiber curvature threshold of 35 degrees. The results of the evaluation metrics for different values of this threshold are shown in Figure 6.19. The blue line corresponds to Figure 6.18a. Fibers 9 (yellow at the top), and 13 to 16 (blue, yellow, cyan and magenta in the left part) were reconstructed correctly in the presented image. This is confirmed by the low deviance from the ground truth indicated by the $l2$ metric. Thus, fibers in regions of high curvature (13, 14) can be reconstructed and nondominant pathways (15, 16) can be traced through fiber crossings with this method. Further, the courses of fibers 4, 10, and 11 are close to the ground truth. The \tan and curv indicate best results for fibers 13 to 16 as they are smoothly curved and almost matching the ground truth.

RK4 PDD

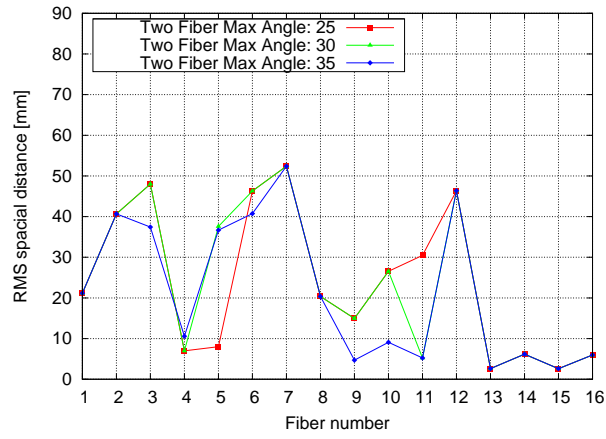
Optimal step size and curvature threshold for voxels with one fiber population for the *RK4 PDD* algorithm were the same as for the *Euler PDD* method (0.25 voxels and 60 degrees respectively). Again, increasing the curvature threshold further did not lead to any changes in the reconstructed fiber trajectories.

Using this step size and curvature for one fiber populations the curvature threshold for voxels with two fiber populations was varied. Figure 6.18b shows the reconstructed fibers for a two fiber curvature threshold of 35 degrees. The results of the evaluation metrics for different values of this threshold are shown in Figure 6.20. The green line corresponds to Figure 6.18b. The *RK4 PDD* method performed worse than the *Euler PDD* algorithm according to the metrics. However, the same number of fiber (i.e. 5) could be reconstructed correctly: fibers 13 and 14 (blue and yellow in the left part), fibers 2, 3 (khaki and violet), and fiber 10 (black). These fiber correspond to regions with high curvature, dominant fiber pathways (2, 3), and a fiber kissing region (10), a very challenging fiber configuration.

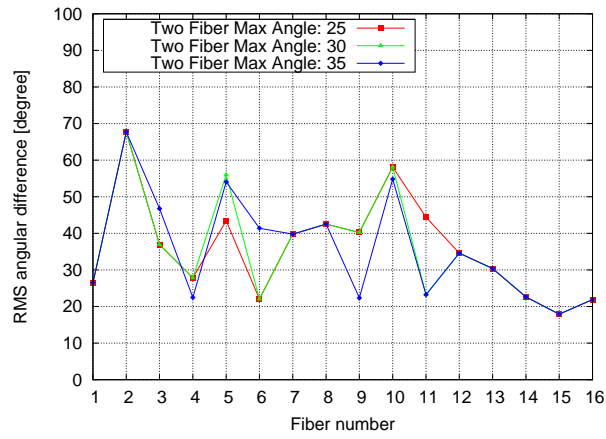
Weighted Dirs Angle

This algorithm is the only one of the five implemented that does not use the *MFC* results to classify voxels with two fiber populations. Instead, a median filtered *stdDev* mask to separate white and gray matter is used. Quantitative experiments showed 0.25 voxels to be the optimal step size for this algorithm.

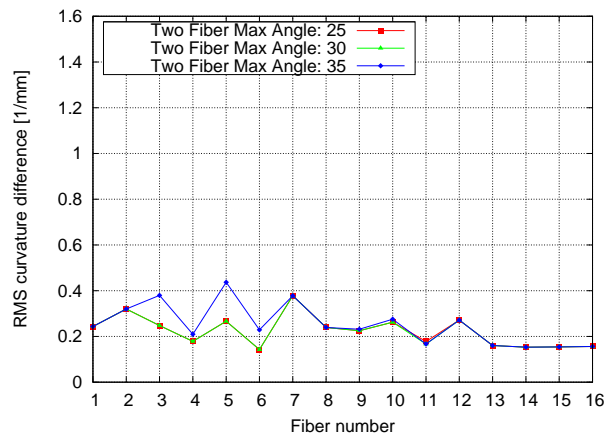
Using this step size the curvature threshold was varied (only one curvature threshold was applied as no *MFC* result was used). Figure 6.22a shows the reconstructed fibers for a curvature threshold of 30 degrees. The results of the evaluation metrics for different values of this threshold are shown in Figure 6.21. The red line corresponds to Figure 6.22a. Four fibers, 4 (khaki), 11, 15, and 16 (red, cyan and magenta on the left side), were reconstructed



(a) l_2 metric

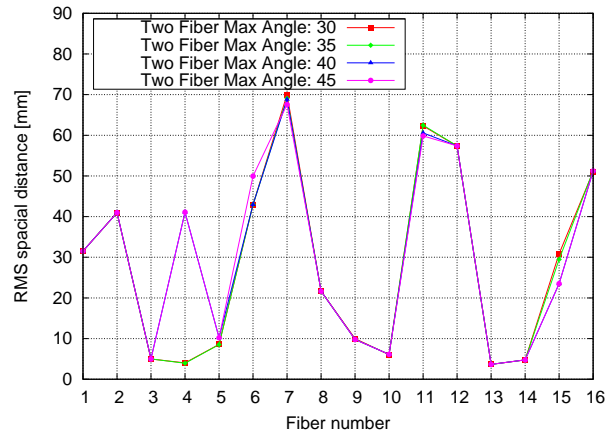


(b) \tan metric

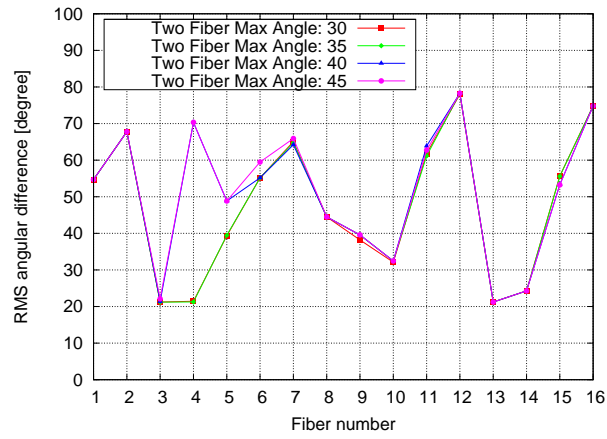


(c) $curv$ metric

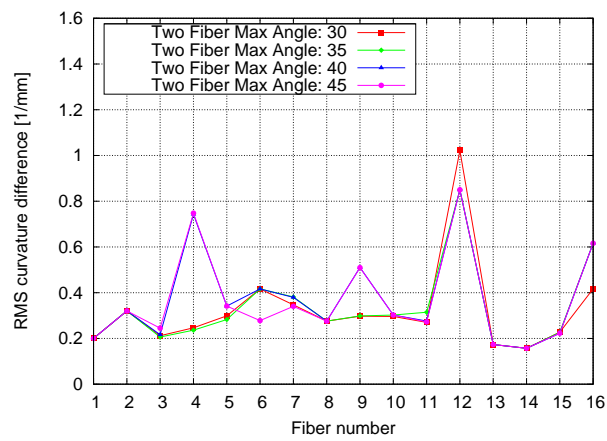
Figure 6.19: Evaluation metrics computed for *Euler PDD* fiber tracing



(a) l_2 metric

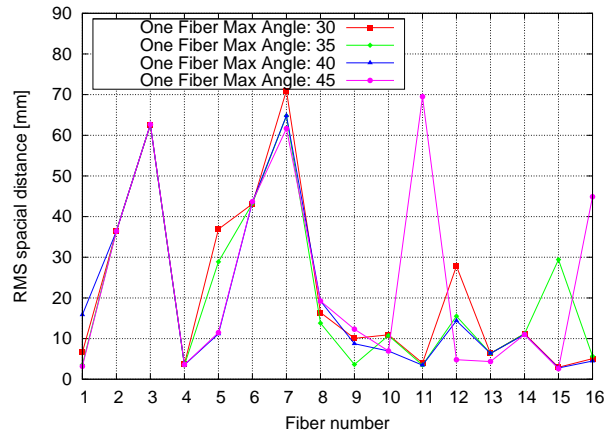


(b) \tan metric

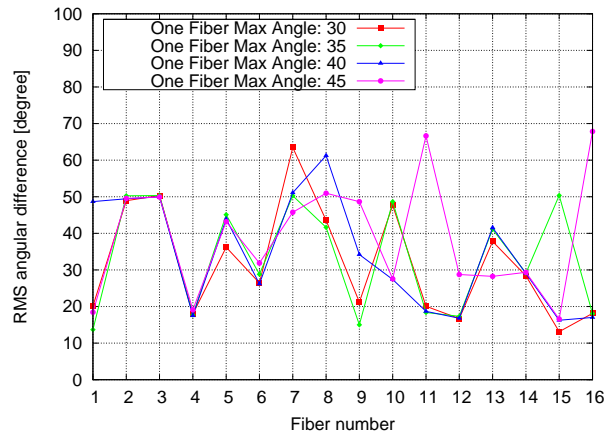


(c) $curv$ metric

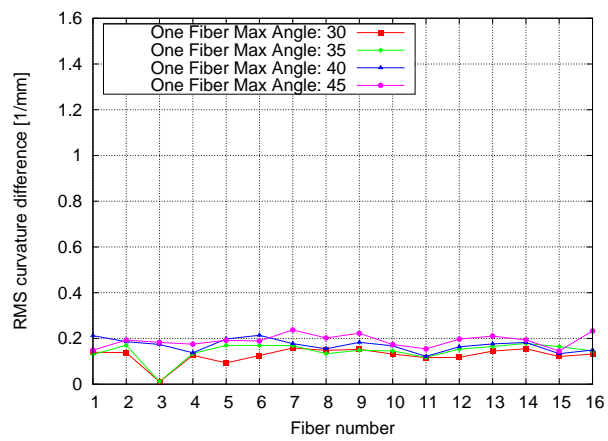
Figure 6.20: Evaluation metrics computed for $RK4$ PDD fiber tracing



(a) l_2 metric



(b) \tan metric



(c) $curv$ metric

Figure 6.21: Evaluation metrics computed for *Weighted Dirs Angle* fiber tracing

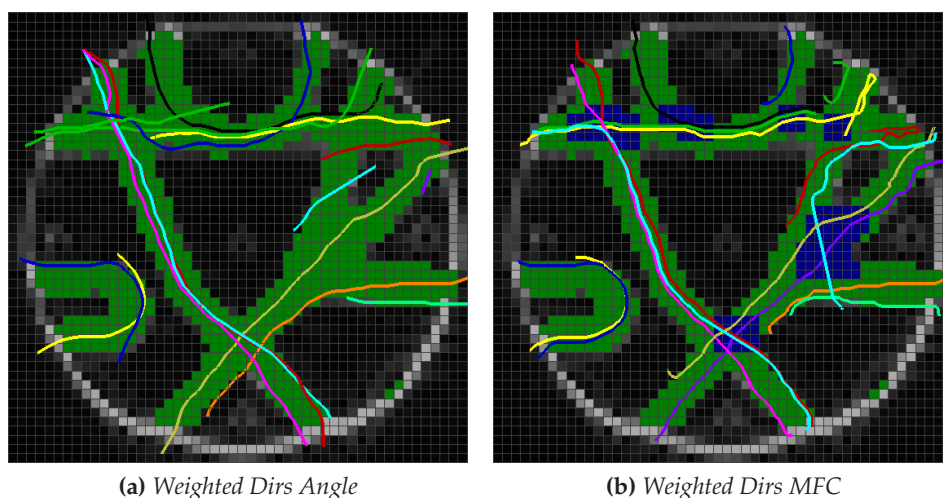


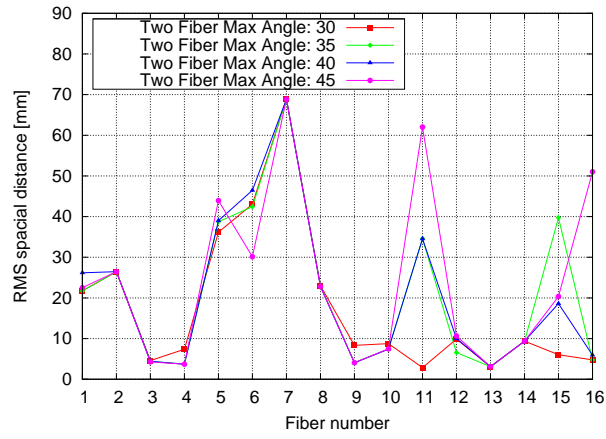
Figure 6.22: Tracing results of the *Weighted Dirs Angle* and *Weighted Dirs MFC* algorithm for a curvature threshold of 30 degrees in voxels with one and two fiber populations respectively.

correctly. Further, the fibers 2 (orange), and 9 (yellow at the top) almost matched the ground truth trajectory. These trajectories belong to both, the dominant and nondominant tract of the fiber crossing at the lower central part of the phantom. The *curv* metric suggests that this algorithm produces the smoothest fiber tracts so far. However, this results are misleading. As can be seen on fiber 3 (violet on the right part) fiber tracing is aborted after only a few steps. The fiber trajectory does not at all match the corresponding ground truth fiber. Nevertheless, this fiber gets the best result in the *curv* metric.

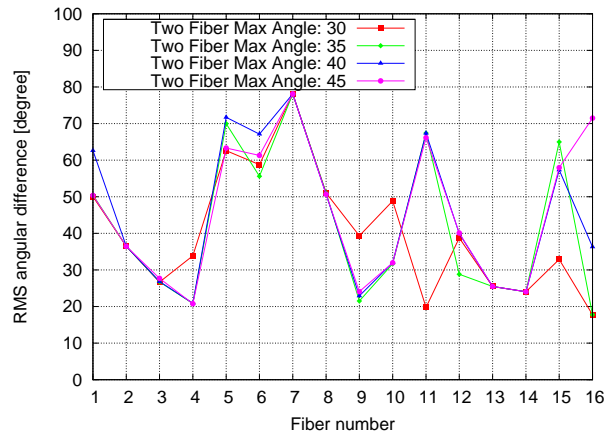
Weighted Dirs MFC

The optimal step size for this algorithm is 0.1 voxels (0.3 mm). The best curvature threshold for voxels with one fiber population is 60 degrees. Using this step size and curvature for one fiber population the curvature threshold for voxels with two fiber populations was varied. Figure 6.22b shows the reconstructed fibers for a two fiber curvature threshold of 30 degrees. The results of the evaluation metrics for different values of this threshold are shown in Figure 6.23. The red line corresponds to Figure 6.22b. In total, four fibers were reconstructed correctly: 3 (violet), 11, 13, and 16 (red, blue, and magenta on the left part). Further, five other fibers almost matched the course of the ground truth (4 (khaki), 9, 12 (yellow and green horizontal fibers at the top), and 14 (yellow on the left part)).

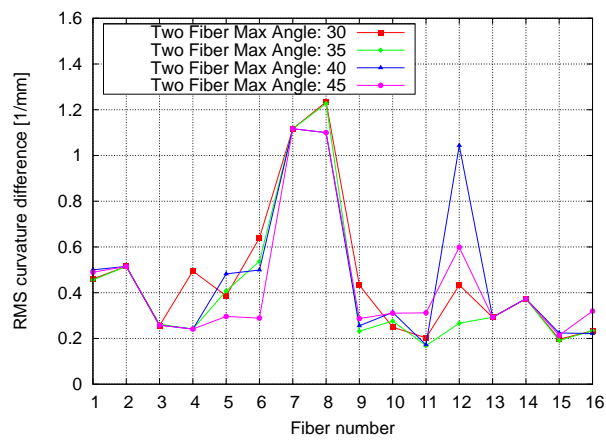
6.4. HARDI_DETERMINISTIC_TRACTOGRAPHY



(a) l_2 metric



(b) \tan metric



(c) $curv$ metric

Figure 6.23: Evaluation metrics computed for *Weighted Dirs MFC* fiber tracing

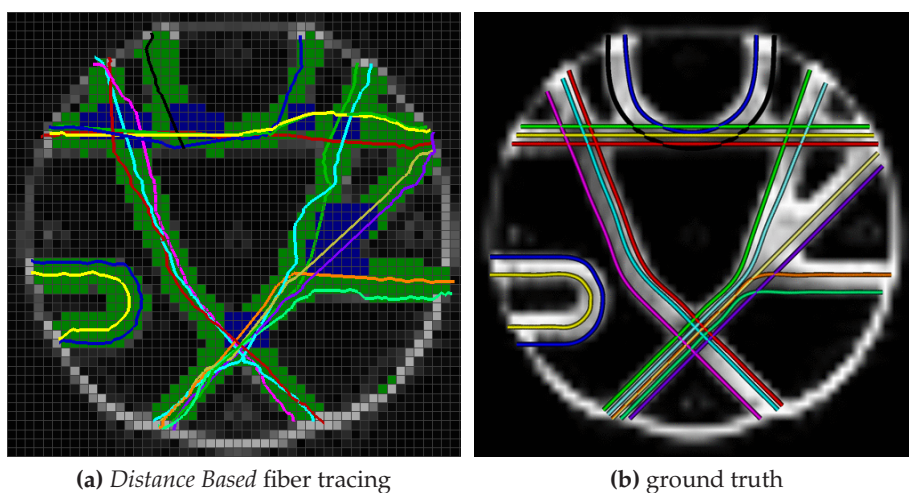


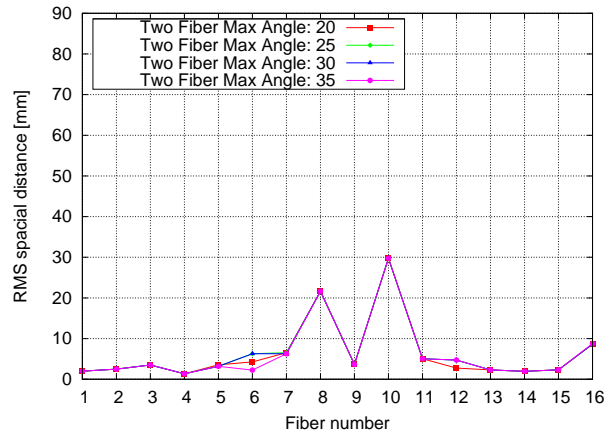
Figure 6.24: Tracing results of the *Distance Based* algorithm for a curvature threshold of 20 degrees in voxels with two fiber populations. The ground truth is shown for comparison.

Distance Based Fiber Tracing

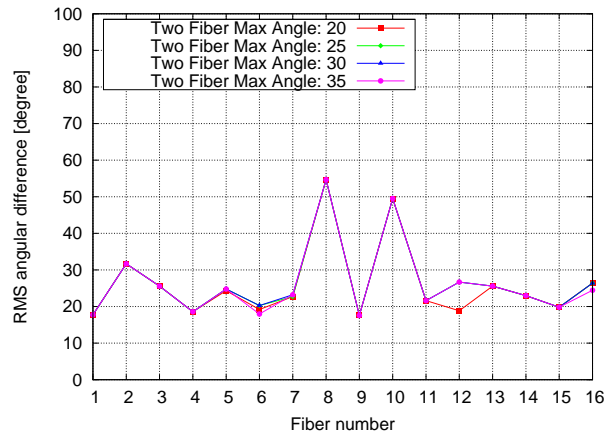
Quantitative experiments show that the optimal step size for this algorithm is 0.5 voxels (1.5 mm). The optimal curvature threshold for voxels with one fiber population is 70 degrees. These are the highest values for these parameters of all presented algorithms. Using the fixed step size and curvature for one fiber populations the curvature threshold for voxels with two fiber populations was varied. Figure 6.24a shows the reconstructed fibers for a two fiber curvature threshold of 20 degrees. The results of the evaluation metrics for different values of this threshold are shown in Figure 6.25. The red line corresponds to Figure 6.24a.

All fibers except fibers 8 and 10 (blue and black at the top) were reconstructed correctly. These two fibers represent the fiber kissing configuration. This type of fiber configuration is similar to fiber crossings at a low angle. However, unlike in a crossing area it is not possible to limit the range of possible directions to a small angular value and propagate the trajectory in one of the directions. In fiber kissings the fiber has to bend constantly inside the multiple fiber area. In fiber crossing areas this would result in taking the wrong “exit” (i.e. following a wrong fiber bundle). This conflict of bending and straight propagation makes fiber kissing regions most challenging. The course of fiber 8 (blue) can be recovered correctly with a step size of 0.1 voxels as lower step sizes are more sensitive to curvature. However, the reconstruction of a majority of other fibers fails with a step size this low.

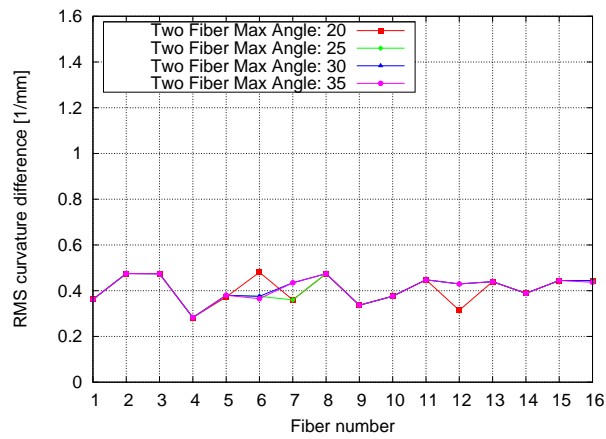
As can be seen in Figure 6.25 the results of this algorithm do not vary much for different curvature thresholds for voxels with two fiber popula-



(a) l_2 metric



(b) \tan metric



(c) $curv$ metric

Figure 6.25: Evaluation metrics computed for *Distance Based* fiber tracing

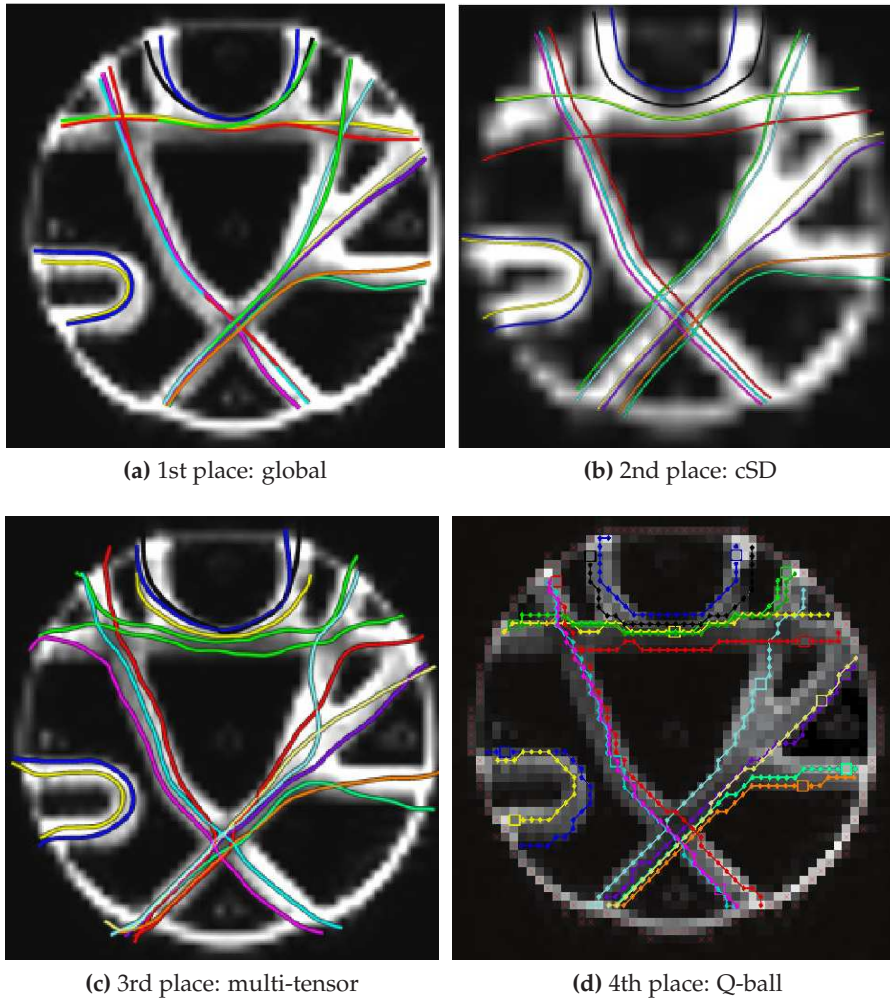


Figure 6.26: Results of first four places at the Fiber Cup.

tions. Even for a non-optimal curvature threshold for voxels with one fiber population of 65 and 60 degrees 12 fibers are reconstructed correctly for most values of the second curvature threshold. This is an indication for the stability of this approach. The overall results according to the l_2 and \tan metrics are better than in the other algorithms. Even the $curv$ metric indicates acceptable results. However, further smoothing can be applied to the reconstructed trajectories to obtain even better results in this metric.

Comparison to the Fiber Cup Results

Figure 6.26 shows the reconstruction results of the first four places (out of ten) at the Fiber Cup. The number of incorrectly reconstructed fiber trajectories for these places is 1 (first place), 0 (second place), and 3 (third and

6.4. HARDI_DETERMINISTIC_TRACTOGRAPHY

Fiber number	1st place (global)	2nd place (cSD)	3rd place (multi-tensor)	4th place (Q-ball)	<i>Distance Based</i> (Q-ball)
1	2.24	6.91	3.01	4.39	1.98
2	2.37	7.83	3.22	4.37	2.48
3	4.98	2.01	4.74	2.62	3.50
4	2.18	2.25	4.36	1.25	1.35
5	1.98	3.75	4.77	1.81	3.55
6	4.25	6.61	54.05	7.7	4.26
7	5.62	3.61	67.03	67.82	6.54
8	2.11	4.91	4.63	4.16	21.65
9	2.61	5.73	18.42	4.15	3.67
10	5.78	5.37	6.45	4.38	29.75
11	3.36	3.27	17.25	2.63	5.04
12	17.02	4.22	4.86	8.24	2.75
13	4.66	1.65	3.78	2.15	2.29
14	2.56	2.73	2.34	3.83	1.95
15	2.16	2.73	2.89	5.52	2.32
16	5.81	5.31	6.01	6.04	8.64

Table 6.4: Comparison of the *Distance Based l2* metric to the results of the Fiber Cup.

fourth places). With only 2 incorrectly reconstructed fiber pathways the *Distance Based* algorithm achieves similar results as these participants at the Fiber Cup.

Tables 6.4, 6.5, and 6.6 compare the best four Fiber Cup results to the result of the *Distance Based* algorithm in all metrics. Compared to the Fiber Cup result the *Distance Based* approach performs well in the *l2* metric. The results of this metric are plotted in Figure 6.27 for better comparison. The results of the other two metrics, however, have room for improvement.

The scoring procedure at the Fiber Cup was organized as follows: The three metrics were computed for each of the 16 fibers resulting in total in 48 ratings. For each rating the candidate with the least deviance from the ground truth won three points. Two points were given to the second best result and one point to the third place. Assuming an imaginary participation of the *Distance Based* algorithm at the Fiber Cup the Fiber Cup score can be recalculated. These fictitious scores are presented in table 6.7. According to this evaluation the *Distance Based* algorithm would be placed second in the *l2* metric and third in the total score of the Fiber Cup. Almost all points for this algorithm result from the *l2* metric. Therefore, future work should not only improve the fiber reconstruction, but also concentrate on regularizing and smoothing the reconstructed fiber trajectories.

6.4. HARDI_DETERMINISTIC_TRACTOGRAPHY

Fiber number	1st place (global)	2nd place (cSD)	3rd place (multi-tensor)	4th place (Q-ball)	<i>Distance Based</i> (Q-ball)
1	9.33	10.83	13.42	31.1	17.74
2	12.38	15.18	16.28	32.98	31.68
3	6.35	9.92	9.11	28.32	25.57
4	5.37	9.65	19.77	19.26	18.58
5	6.2	6.96	22.62	29.71	24.33
6	8.1	11.87	56.62	46.86	19.17
7	11.44	8.32	67.49	65.66	22.72
8	8.26	12.34	18.29	33.27	54.69
9	6.73	11.7	46.99	35.27	17.57
10	12.1	15.21	13.01	34.69	49.52
11	4.85	8.00	45.92	29.69	21.62
12	46.74	12.02	18.91	43.58	18.85
13	12.75	10.94	15.76	24.89	25.58
14	14.74	13.18	16.19	27.67	22.99
15	4.25	7.39	11.97	36.70	19.82
16	7.54	5.44	25.49	27.59	26.39

Table 6.5: Comparison of the *Distance Based tan* metric to the results of the Fiber Cup.

Fiber number	1st place (global)	2nd place (cSD)	3rd place (multi-tensor)	4th place (Q-ball)	<i>Distance Based</i> (Q-ball)
1	0.025350	0.030870	0.070594	0.209874	0.361998
2	0.039980	0.054298	0.076433	0.274584	0.475126
3	0.021249	0.437153	0.086352	0.235227	0.474395
4	0.014065	0.044516	0.079855	0.162110	0.282796
5	0.020442	0.060609	0.091297	0.282577	0.372536
6	0.026327	0.043200	0.126881	0.484764	0.480412
7	0.022992	0.048882	0.128838	0.381329	0.35991
8	0.043299	0.049991	0.066174	0.238658	0.474324
9	0.016785	0.041078	0.109614	0.322867	0.336436
10	0.026207	0.054317	0.091980	0.232870	0.375708
11	0.012756	0.021897	0.110738	0.296971	0.447673
12	0.032568	0.038112	0.115901	0.278794	0.314087
13	0.070287	0.061247	0.085933	0.203200	0.43945
14	0.068706	0.256229	0.082921	0.209025	0.388423
15	0.010659	0.021063	0.078300	0.306166	0.444443
16	0.020795	0.023347	0.114350	0.383827	0.444116

Table 6.6: Comparison of the *Distance Based curv* metric to the results of the Fiber Cup.

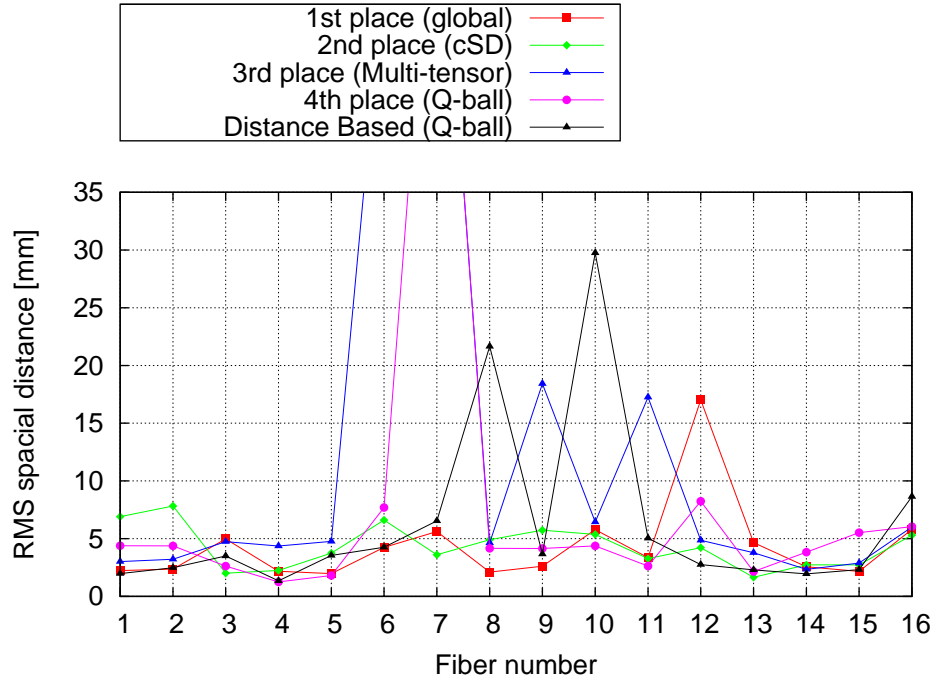


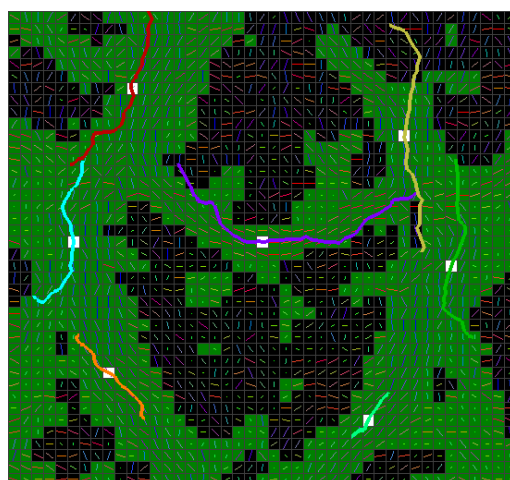
Figure 6.27: Comparison of the l_2 metric results to the Fiber Cup participants.

	1st place (global)	2nd place (cSD)	3rd place (multi-tensor)	4th place (Q-ball)	<i>Distance Based</i> (Q-ball)
l_2	25	18	7	17	20
\tan	41	35	10	0	4
curv	47	28	6	0	0
total	113	81	23	17	24

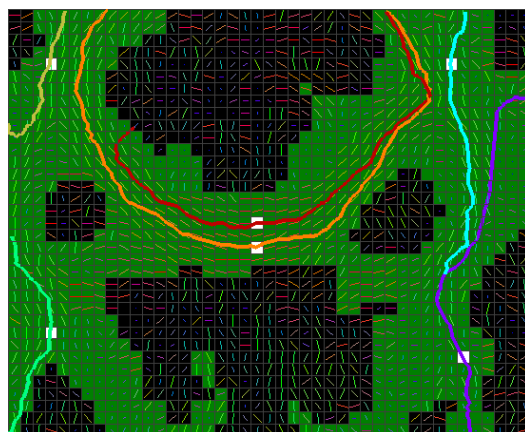
Table 6.7: Fictitious Fiber Cup results with participation of the *Distance Based* fiber tracing approach.

6.4.3 Tractography Results on Human Brain Data

Tractography evaluation was performed on a coronal slice (XZ -plane) in the region of the CC and the CST. None of the fiber tracing algorithms provided good results on this data set. The fiber trajectories presented in Figure 6.28a were reconstructed with the *Distance Based* algorithm. The chosen thresholds were 0.5 (Step size), 65 (One Fiber Max Angle), 8 (DistanceVectors), 0.5 (CurrentDirWeight). The median filtered stdDev mask was used to separate anisotropic and isotropic diffusion voxels. Anisotropic voxels are shown in green on the slice containing the seed point in Figure 6.28a. The *MFC* was not used due to its bad classification results. Note that the fiber trajectories in Figure 6.28a are projected on the slice containing the



(a)



(b)

Figure 6.28: Tracing results on human brain data. In (a): coronal slices in the region of the corpus callosum (CC) and the corticospinal tract (CST). In (b): axial slices in the region of the CC and lateral fibers. Seed voxels are marked white.

seed points. The true trajectories cross several slices and, therefore, do not resemble the PDDs of the shown slice in all voxels.

There are several possible reasons for the bad fiber tracing results on this data. First, as was pointed out before, a data set with a b-value of 700 is not suitable for Q-ball reconstruction. Further, only 41 diffusion encoding directions were used to obtain this dataset. Usually, 60 and more gradient directions are used for HARDI data. A third problem is more fundamental. The *So_dODF_Visualization* and *So_Fiber_Visualization* modules support only axial slice (XY -plane) visualization so far. However, for the selected ROI the data had to be viewed in the XZ -plane. The MeVis-

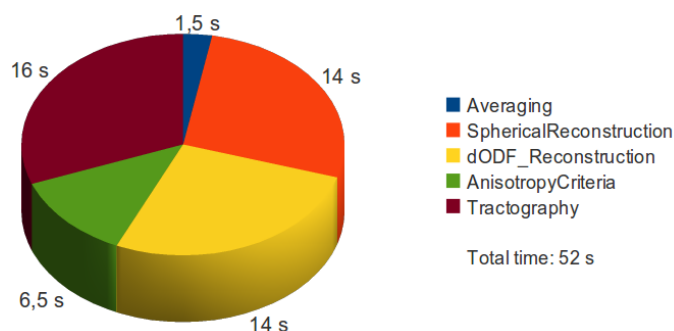


Figure 6.29: The fractions of the total computation time contributed by each module.

Lab library provides the module *SwapFlipDimension* to swap the axis of the dataset. Though, the swapped dataset was corrupted: The diffusion direction in the CST was perpendicular to the anatomically well-known direction (in the coronal view the CST diffusion direction was aligned with the y -axis instead of the anatomically correct z -axis). This orientation had to be corrected manually in the *So_dODF_Visualization* module. However, it remains still unclear why the swapping of the dimension axis introduced errors and whether all manually added corrections were applied properly. Seed point placement and fiber reconstruction was made on the manually corrected data. As the tracing results were not acceptable at all the suspicion remains that the dataset got corrupted at some step of the image processing pipeline.

Therefore, a second ROI was selected to evaluate the tracing results on human brain data. The second ROI was placed on axial slices in the region of the CC including the lateral connected fibers. Thus, no potentially erroneous axis flipping was necessary for this ROI. The tracing results were obtained with the same parameters (Figure 6.28b). Since the fiber trajectories in the second ROI are properly reconstructed it can be stated that the *Distance Based* fiber tracing algorithm works on human brain data. However, further evaluation on other ROIs (and with proper data on the first ROI) is necessary.

6.5 Total Computation Time

This section illustrates the fraction of the total computation time needed by the individual modules of the *HARDI_Tractography* image processing network as presented in 5.2.3. Time measurements were performed with SH order 4 and tessellation order 3. The *Distance Based* algorithm is used for fiber tracing. The results are presented in Figure 6.29. The total computation time of 52 seconds is clinically acceptable.

Part IV

Conclusion

Chapter 7

Summary

This chapter provides a short summary of this thesis. The first section summarizes the theoretical background that forms the basis of this work. In the second section a short review of the implementation is provided and the results are discussed.

7.1 Theoretical Part

The theoretical part of this thesis provided an overview on MRI and diffusion MRI. Background knowledge on signal formation and its physical basis was given. Different types of images, such as relaxation and diffusion weighted images, were introduced. Their individual purposes and significance for specific tissue was explained. Further, background knowledge on diffusion was provided to allow for an assessment of the measured quantities and the associated uncertainty.

Based on this knowledge an introduction to diffusion tensor imaging (DTI), a widely used modality in medical practice, was given. DTI plays an important role in the reconstruction of white matter fiber bundles that connect different functional areas of the brain. The reconstruction of these fibers provides insight into brain structure and allows to analyze the development of the brain from childhood to adulthood. Pathological changes caused by diseases such as multiple sclerosis (MS) and Alzheimer's disease can be investigated and improve our understanding of the brain. Further, white matter fiber reconstruction plays an important role in neurosurgical planning.

Although being a valuable modality DTI has severe limitations as it is not able to resolve multiple fiber populations per voxel. Since lots of brain regions have complicated structures the inability of resolving several fiber populations per voxels prevents DTI from correctly reconstructing the white matter pathways. In regions with fiber crossings, branchings, or kissings often no directional information on the fiber trajectories can be

obtained. These limitations stem from the low number (often 6 to 20) of diffusion encoding directions used in DTI.

High angular resolution diffusion imaging (HARDI) techniques have been developed to overcome these limitations. In HARDI often 60 and more diffusion encoding gradients are applied. At the cost of a higher image acquisition time higher angular resolution is provided that allows to identify even complicated structures in the brain. Several methods have been developed to process the acquired data and reconstruct a diffusion orientation distribution function (dODF) representing the diffusivity directions in each voxel. Of the methods presented analytical Q-ball reconstruction was chosen as basis for the implementation. This reconstruction method is computationally light and has been widely studied by several research groups. One of the disadvantages of the Q-ball method, however, is its demand for a b-value of at least 3000 during the acquisition of the data. In other words, to reconstruct sharp dODF the Q-ball approach demands for high sensitivity on diffusion. Lower b-values lead to loss of angular information and result in blurred dODFs.

Based on diffusion tensors or dODFs fiber trajectories can be reconstructed. This process is called fiber tracing or tractography. Early DTI based approaches used Euler or RK4 integration with a constant step size to propagate a streamline along the direction of highest diffusivity often called the principal diffusion direction (PDD). The FACT (fiber assignment by continuous tracking) approach also follows the PDD but uses a varying step size for integration. Following approaches like TEND (tensor deflection) suggested to use the whole diffusion tensor instead of only using the first eigenvector resembling the PDD. Thus, streamlines were deflected along the PDD in regions with low directional uncertainty and propagated in straight lines when uncertainty was high. Many HARDI approaches examine several directions in each voxel. Often the local maxima above a certain threshold are selected and the streamline continues along the maximum forming the smallest angle with the incoming direction. Some approaches split the streamline when the angle between local maxima is sufficiently high and follow multiple pathways.

These approaches are called *deterministic* as they deterministically follow diffusion and always produce same results if applied with same parameters to the same datasets. Probabilistic approaches were developed to account for the uncertainty of the measured data and allow to assess reconstructed pathways. For these approaches first a function representing the uncertainty needs to be generated. Often bootstrapping and Bayesian methods are applied for this purpose. Once the uncertainty is represented a high number of streamlines is propagated by drawing samples from the uncertainty distribution at each step. Voxels crossed by many streamlines are considered as being the most probable fiber pathway.

These two groups of tractography algorithms (deterministic and prob-

abilistic) are local approaches as they estimate the direction of highest diffusion at each step locally. However, two brain regions are not necessarily connected by a fiber that follows the highest diffusion direction in each step. Therefore, global tractography approaches try to find a globally optimal diffusion path between two regions. Locally seen this path might not align with the PDD along the whole trajectory. However, out of all possible paths the reconstructed fiber trajectory is the path of maximum diffusivity between the regions in question.

7.2 Implementation and Results

Several modules have been implemented for the MeVisLab medical image processing framework in the course of this thesis. These modules perform the tasks of reconstructing the dODF and apply several criteria to separate voxels with isotropic diffusion from voxels with anisotropic diffusion. Based on this separation a module was developed to classify voxels as containing one or two fiber populations. Further, two visualization modules were developed to examine the data and display the reconstructed fiber pathways.

The main focus of the implementation lies on the tractography module. Five different algorithms were implemented to deterministically reconstruct white matter fibers. The first two algorithms (*Euler PDD* and *RK4 PDD*) were basic extensions of classic PDD based algorithms with Euler and RK4 integration respectively. The PDD was interpolated among neighboring voxels and the fiber trajectory followed the resulting PDD directions. No other directions were considered for possible path continuation. The extension to HARDI data was limited to using the classification results that provided the number of fiber populations in each voxel. Based on this number and the angle of the incoming direction either the PDD, the interpolated PDD, or the current direction was used to continue the streamline. Tracing was stopped in voxels with isotropic diffusion.

The third and fourth algorithm (*Weighted Dirs Angle* and *Weighted Dirs MFC*) were inspired by state-of-the-art tractography algorithms that identify the closest local maximum to the incoming direction. A high threshold was used to filter the possible directions in each voxel. All remaining directions were considered as representing high diffusivity without determining local maxima. The *Weighted Dirs Angle* algorithm reconstructs fibers on a white matter mask without identifying regions with two fiber populations. Based on the scattering angle of possible directions (i.e. the uncertainty of the PDD) in the current voxel and the angle between the streamline and the PDD a direction to continue the streamline was determined. This direction is a weighted average of the current direction, the closest direction of the thresholded dODF, and the PDD. However, the PDD can be

ignored in some cases. The *Weighted Dirs MFC* algorithm is similar but does not use the scattering angle of the dODF to determine the weights for the individual direction vectors. Instead, the information about the number of fiber populations per voxel is used.

The last algorithm (*Distance Based*) is a completely new approach that has not been used before. In voxels with one fiber population the streamline tries to keep a constant distance to the white matter borders. In complex regions (i.e. with two fiber populations) one of the possible directions is chosen that allows for a long trajectory continuation. The decision which direction to choose in these cases is based on the resulting streamline angle and a curvature hint calculated from last voxels.

The tractography results of these algorithms were evaluated on a phantom dataset with the most challenging fiber configurations. This dataset was originally used for tractography evaluation at the Fiber Cup during the MICCAI 2009 conference. A ground truth and common seedpoints were provided for the evaluation. The *Euler PDD* algorithm was able to reconstruct 5 out of 16 fibers correctly (several other fibers were close to the ground truth). The *RK4 PDD* algorithms performed similar (5 fibers) but reconstructed different fiber trajectories correctly than the first algorithm. Since the fiber configurations in the dataset are very challenging these bad results for PDD based approaches were not surprising.

The approaches based on direction averaging reconstructed 4 fibers correctly. In general, the *Weighted Dirs MFC* approach provided better results. Many trajectories of the *Weighted Dirs Angle* approach were stopped too early as they left the dataset. A simple averaging of directions is, therefore, insufficient since an average direction does not necessarily represent a diffusion direction. The additional information provided by the MFC helped to improve the results.

Best results were achieved by the *Distance Based* algorithm. 14 out of 16 fibers were reconstructed correctly. The remaining two fibers represented a kissing fiber configuration. Further work is needed to enable this algorithm to reconstruct even this configuration. One possible extension would be to use smaller step sizes in regions with high curvature. In regions with multiple fiber populations the maximum tracing distances inside white matter could be determined for each potential tracing direction. By selecting the local maxima of these distances that best fit the current curvature of the fiber the selection of the next direction could be improved. Kissing fiber configuration could be handled this way. The calculation of the lateral distances could benefit from a continuous white matter mask as the discrete voxel borders (especially in diagonal fibers) result in erroneous distance estimations. Further, the calculation of the curvature hint can be improved.

Nevertheless, the results of this algorithm are very similar to the best algorithms presented at the Fiber Cup. However, the resulting fiber trajectories have to be smoothed and interpolated to achieve better results at all

metrics used for evaluation at the Fiber Cup.

The *Distance Based* algorithm was also evaluated on human brain data. Due to difficulties with data handling the results for one of two regions of interest (ROIs) can not be trusted and needs further evaluation. The fibers in the second ROI were reconstructed reliably.

The morphological fiber classification (MFC) was developed to improve the tractography results. The development of this global classification approach for multiple fiber configuration was necessary as the usual methods (e.g. FMI) did not provide good results. However, so far the MFC works only on the phantom dataset. The reason is obvious. The MFC uses fixed mask sizes for its morphological operations. The fiber bundles of the phantom dataset are also of almost constant width. If both, the mask size and fiber width fit each other well MFC provides good results. For the human brain dataset MFC proved useless since the fiber width in the brain varies greatly. Therefore, MFC needs to be extended to use variable mask sizes for different regions of the brain. Alternatively, the MFC results obtained with constant but different mask sizes could be combined.

In conclusion, a fiber tracing algorithm was developed that is able to reconstruct fibers in most of the complicated fiber configurations. This algorithm needs more evaluation on human brain data. Also a dataset with a sufficiently high b-value is necessary for a proper evaluation. Alternatively, this algorithm could be applied to data from other reconstruction methods. Especially the constrained spherical deconvolution (cSD) reconstruction approach provides sharp ODFs and resolves even fiber crossings at low angles. Sharp ODFs could also make the classification of fiber populations per voxel unnecessary as the number of fibers would be equal to the number of ODF maxima.

Chapter 8

Outlook

This chapter gives a brief outlook to fiber tracing in challenging regions (e.g. regions with edema or tumors) in the first section. The second and third sections briefly discuss possibilities for clustering and visualization.

8.1 Fiber Tracing in Challenging Regions

Regions with edema and tumors are challenging because these regions feature isotropic diffusion. Axonal fibers can be infiltrated, surrounded, or pushed aside by these pathologies. In the last case it is likely that the affected fiber bundles are deformed but still visible in diffusion weighted images due to their anisotropic diffusivity. In the former two cases, however, the anisotropic diffusion in the fibers is concealed by the isotropic diffusion of the pathologies. Data reconstruction from the measured signal should explicitly consider these two types of diffusion.

One possible data reconstruction method might be the multi compartment model. By explicitly modeling isotropic and anisotropic diffusion these two diffusion types could be distinguished in a voxel. If this separation is successful the orientation of the anisotropic compartment can indicate the underlying fiber orientation. A second possibility to reconstruct the concealed fiber orientation might be spherical deconvolution. The measured signal can be deconvolved with different kernels varying from highly anisotropic to completely isotropic diffusion. The results of each deconvolution can be compared and could indicate a hidden fiber structure. As high b-values are more sensitive to diffusion than low b-values their application might reveal even small diffusion inhomogeneities in challenging region and help identify occluded fibers.

However, it might not be possible at all to reveal the hidden structure. In any case the applied fiber tracing method should be probabilistic (or even global) to handle the high uncertainty in such regions. If any directional data inside the tumor can be gained and fiber tracing appears rea-

sonable tracing should be started from two ROIs. These ROIs should be placed on both sides of the region in question where the fiber is known to enter or exit the tumor.

8.2 Clustering of Fiber Populations

In many cases, especially in probabilistic fiber tracing approaches, several streamlines are generated. These streamlines need to be clustered in order to know which of the streamlines represent the same fiber population. In regions where only one fiber population exists the solution is trivial: All reconstructed trajectories in this region of question belong to the same fiber population. Clustering fibers in regions of fiber crossings at high angles can be accomplished by examining the fiber angles at a selected point or region. Fiber passing this region and having a low angle to one another most probably belong to the same fiber population. On the other side, fibers forming an angle higher than the chosen threshold belong to the crossing (or crossed) fiber population.

Regions of fiber branchings, kissings, and crossings at low angles are more challenging. In these cases it might be beneficial to define several ROIs close to the region in question. Boolean operators can then be applied to these ROIs to identify fiber populations (e.g. all fiber passing through ROI 1 *and* ROI 2 belong to the same fiber population).

8.3 Visualization

Fiber trajectories are usually visualized by streamlines (i.e. rendered lines). These visualization can be enhanced by rendering three-dimensional tubes (streamtubes). Further, these tubes can be overlaid with glyphs of the corresponding dODF in challenging regions. In this way one can assess whether or not the fiber trajectory has taken the best possible way indicated by the data.

Uncertainty in the fiber trajectories reconstructed by probabilistic algorithms can be visualized with semi-transparent fibers. The more fibers choose a particular pathway (and, thus, overlay each other) the more opaque the trajectories becomes. In cases where the main purpose is to visualize the probabilities associated with the different reconstructed streamlines the width of the streamtubes can be varied. More probable connections result in wider tubes than connections with low probabilities.

Appendix A

Module Documentation

This appendix provides a user-level documentation for the modules that were implemented in the course of this thesis.

HARDI_dODF_Reconstruction

Description Given spherical harmonics coefficients this module reconstructs the diffusion orientation distribution function (dODF) for every voxel in the input image. The spherical coordinates for the dODF sampling are derived from a tessellation of an icosahedron. Tessellation orders of 1 through 5 are supported and can be chosen via a parameter in the GUI-panel.

GUI-panel The GUI-panel of the module *HARDI_dODF_Reconstruction* is shown in Figure A.1. It provides access to the following module fields:

- **Tessellation order:** Determines the tessellation order of an icosahedron and thus the number of sampling direction for the dODF reconstruction. Valid values are integers from 1 to 5. The sampling directions belonging to each tessellation order are shown in table A.1.
- **Isotropy weight:** Weighting factor for the SH coefficient representing isotropic diffusion. By default, this factor is set to zero to deemphasize isotropic diffusion and, thus, reveal the directional information of the higher order SH coefficients.



Figure A.1: GUI-panel of the *HARDI_dODF_Reconstruction* module

Tessellation order	Sampling directions
1	12
2	42
3	162
4	642
5	2562

Table A.1: The number of sampling directions belonging to each tessellation order of an icosahedron.

Input This module has one input image parameter. The input is an image of extension $(x, y, z, c, t, u) = (X, Y, Z, 1, R, 1)$. X, Y, Z is the size of the image in voxels. The t dimension holds $R = 0.5(L + 1)(L + 2)$ coefficients that are determined from the measured MRI signal and spherical harmonics order L . The size of dimensions c and u must be 1.

Output This module has one output image parameter. The output is an image of extension $(x, y, z, c, t, u) = (X, Y, Z, 1, N, 1)$ holding the dODF values in the t dimension. X, Y, Z are the same as in the input image. N is the number of used sampling directions for dODF reconstruction.

HARDI_AnisotropyCriteria

Description This module allows to calculate six different anisotropy criteria on input images with HARDI data. The goal of these criteria is to classify the number of fiber populations per voxel (i.e. 0, 1, or 2 populations) or to separate voxels with isotropic diffusion from voxels with anisotropic diffusion.

Algorithm 11 FMI criterion classification

```

1: if  $FMIa < FMI0$  then
2:   0 fiber populations
3: else
4:   if  $FMIb < FMI1$  then
5:     1 fiber population
6:   else
7:     2 fiber populations
8:   end if
9: end if

```

GUI-panel The tabs of the GUI-panel of the *HARDI_AnisotropyCriteria* module are shown in Figure A.2. The tab *Criterion* allows to choose the cri-

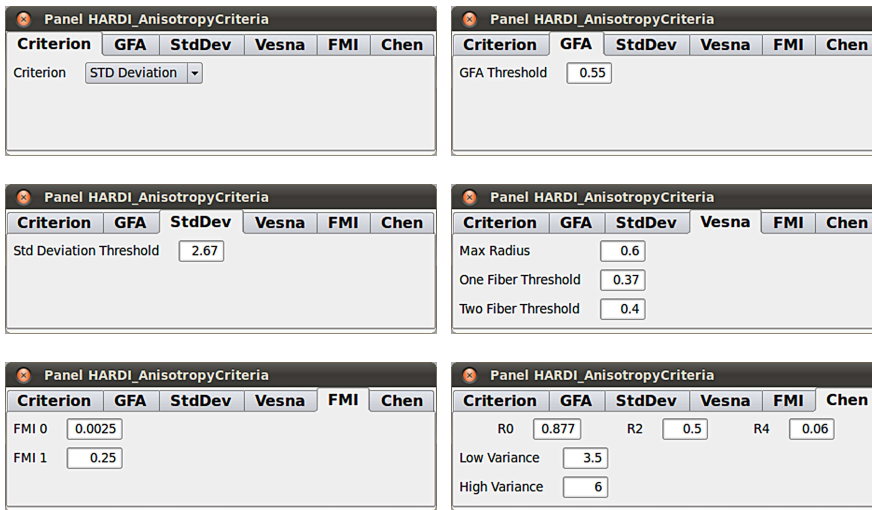


Figure A.2: GUI-panel of the *HARDI_AnisotropyCriteria* module

terion that is applied to the input image. Every criterion has a separate tab where the corresponding thresholds can be set. An exception is the *MDL* criterion as no thresholds need to be applied to this classification criterion. The thresholds for each tab are explained in the following list.

- **GFA:** The general fractional anisotropy (GFA) separates voxels with isotropic diffusion (values below the threshold) from voxels with anisotropic diffusion (values equal to or above the threshold). The GFA values are normalized to be in the range $[0, 1]$.
- **StdDev:** The standard deviation of the dODF separates voxels with isotropic diffusion (values below the threshold) from voxels with anisotropic diffusion (values equal to or above the threshold). The standard deviation can take arbitrary values and highly depends on the dataset.
- **Vesna:** This method counts dODF values above the threshold *Max Radius* in the min-max normalized dODF for each voxel. This value is further transformed to the range $[0, 1]$ and two thresholds are applied to it. Voxels with a value below *One Fiber Threshold* are classified as containing one fiber population. Voxels with a value above *Two Fiber Threshold* are classified as isotropic (zero fiber populations). Finally, voxels with a value between these two thresholds are classified as containing two fiber populations.
- **FMI:** For this method two values are calculated: *FMI_a* indicating the significance of higher order SH coefficients over the 0th order coefficients and *FMI_b* indicating the significance of higher order SH co-

Algorithm 12 Chen's criterion classification

```
1: if  $R_0 > R0$  or  $\text{variance} < \text{Low Variance}$  then
2:   0 fiber populations
3: else
4:   if  $R_2 > R2$  or  $\text{variance} \geq \text{High Variance}$  then
5:     if  $R_{\text{multi}} > R4$  then
6:       2 fiber populations
7:     else
8:       1 fiber population
9:     end if
10:  else
11:    2 fiber populations
12:  end if
13: end if
```

efficients over the 2nd order coefficients. The classification is carried out according to algorithm 11.

- **Chen:** This criterion demands for the most thresholds. Values R_0 , R_2 , and R_{multi} are calculated from SH coefficients. Additionally, the variance of the dODF is needed for this criterion. The voxels are classified according to algorithm 12.

Input This module has two input image parameters. Both input images are needed for correct functionality of this module.

- **Left input connector:** An image of extension $(x, y, z, c, t, u) = (X, Y, Z, 1, R, 1)$. X, Y, Z is the size of the image in voxels. The t dimension holds $R = 0.5(L + 1)(L + 2)$ coefficients that are determined from the measured MRI signal and spherical harmonics order L . The size of dimensions c and u must be 1.
- **Right input connector:** An image of extension $(x, y, z, c, t, u) = (X, Y, Z, 1, N, 1)$ holding the dODF values in the t dimension. X, Y, Z is the image size in voxels. N is the number of used sampling directions for dODF reconstruction.

Output The output is an image of extension $(x, y, z, c, t, u) = (X, Y, Z, 1, 1, 1)$. X, Y, Z are the same as in the input images. The scalar value at each voxel indicates the number of fiber populations that were identified by the selected anisotropy criteria. The number of identified fiber populations can be 0, 1, or 2. In theory, however, more than 2 fiber populations can be identified by the MDL criterion offered by this module.

MFC_Macro

Description This macro module incorporates an inner network that was built around the *HARDI_AnisotropyCriteria* module. Several other modules of the MeVisLab library, especially morphological operations, were added to the internal network. The module's name *MFC* (morphological fiber classification) attributes to these morphological modules. The purpose of this module is to improve the classification results of the *HARDI_AnisotropyCriteria* by applying morphological operations to the anisotropy criteria offered by this module.

GUI-panel The GUI-panel of this module has exactly the same structure and functionality as the GUI-panel of the *HARDI_AnisotropyCriteria* module (Figure A.2).

Input The input images of this module have the same properties as the input images of the *HARDI_AnisotropyCriteria* module as they are directly passed to this module internally.

Output The *MFC_Macro* module offers three output images with the same properties and structure. The image at the rightmost output pin is the output of the *HARDI_AnisotropyCriteria* module. The middle output pin provides a median filtered version of the latter image. A kernel size of $3 \times 3 \times 3$ is used for filtering. The leftmost output pin offers the actual classification result of the MFC. The MFC always classifies voxels as having 0, 1, or 2 fiber populations independent of the chosen criterion of the *HARDI_AnisotropyCriteria* module.

HARDI_deterministic_Tractography

Description This module reconstructs white matter fibers on an input image containing the diffusion orientation distribution function (dODF) for every voxel. Different deterministic algorithms can be chosen for this reconstruction. Fiber tracing is starting at seedpoints defined via the GUI-panel.

GUI-panel The GUI-panel of this module has two tabs (Figure A.3). The *SeedPoints* tab allows to specify seedpoints that will be used for fiber tracing. The seedpoints have to be integer voxel coordinates indicating the voxel at the center of which the tracing will be started. Each seedpoint has to be entered in a separate line. Its coordinates have to be separated by an arbitrary number of spaces. By clicking on the *Apply* button an event

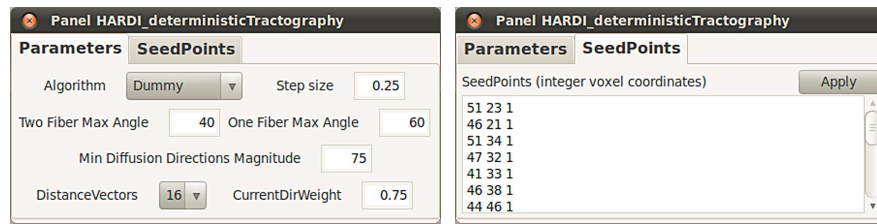


Figure A.3: GUI-panel of the *HARDI_deterministicTractography* module

is triggered and fiber tracing is performed starting at the new seedpoints. If the *Apply* button is not clicked the seedpoints will be copied on the next module field change of the *Parameters* tab. Any coordinates outside of the image dimensions will be ignored. The functionality of the *Parameters* tab is explained below.

- **Algorithm:** Defines the algorithm that is used for fiber tracing. Possible values are *Dummy* (no fiber tracing), *Euler PDD*, *RK4 PDD*, *Weighted Dirs Angle*, *Weighted Dirs MFC* and *Distance Based*.
- **Step size:** Indicates the integration step size used for fiber tracing. The step size is specified in voxels.
- **One Fiber Max Angle:** The maximum angle a fiber can bend in every step in regions with one fiber population per voxel.
- **Two Fiber Max Angle:** The maximum angle a fiber can bend in every step in regions with two fiber populations per voxel.
- **Min Diffusion Directions Magnitude:** Lower bound for dODF directions that are used for fiber tracing. The given value specifies the percentage of the magnitude of the PDD.
- **DistanceVectors:** Number of distance vectors used to estimate the best tracing direction. Possible values are 4, 8, and 16. This parameter only affects the *Distance Based* algorithm.
- **CurrentDirWeight:** Weighting factor for the current tracing direction. The specified value has to be between 0 and 1. This parameter only affects the *Distance Based* algorithm.

Input This module has two input image connectors. Both input images are needed for correct module functionality.

- **Left input connector:** An image of extension $(x, y, z, c, t, u) = (X, Y, Z, 1, N, 1)$ holding the dODF values in the t dimension. X, Y, Z is the image size in voxels. N is the number of used sampling directions for dODF reconstruction.

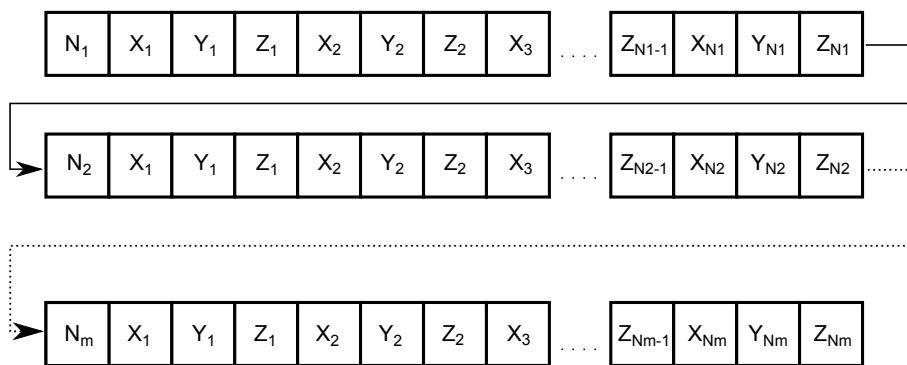


Figure A.4: Structure of the output image. The array is divided into lines for illustration purposes. Each line represents one fiber.

- **Right input connector:** An image of extension $(x, y, z, c, t, u) = (X, Y, Z, 1, 1, 1)$. X, Y, Z is the image size in voxels. The scalar value at each voxel indicates the number of fiber populations in this voxel. The number of identified fiber populations can be arbitrary. However, the fiber tracing algorithms use a maximum number of 2 fiber populations per voxel.

Output The output image contains the reconstructed fiber trajectories and is basically a one-dimensional array. The extent of every dimension except t is 1. The dimension t holds the information of the fibers reconstructed from all seedpoints. Its extent is $s + v * 3$, where s is the number of seedpoints and v the sum of vertices in all fibers. The fiber data in dimension t is an array of floating point numbers and is organized as is shown in Figure A.4. The first number indicates the quantity of three-dimensional coordinates representing the first fiber. These coordinates are stored in the array directly after the first number. First, the x coordinate, followed by the y and z coordinates (in this order) of the first vector is placed in the array. Then, the coordinates of the second vector follow, etc. If more than one fiber was reconstructed the number of vectors belonging to the next fiber is stored in the array at the position behind the last coordinate of the first fiber. Its coordinates follow as described above.

So_dODF_Visualization

Description This module visualizes the diffusion orientation distribution functions (dODF) of an input image in several ways. Supported visualization modes are points, lines, meshes or glyphs. Further, only the highest values of the dODF can be displayed up to an user defined threshold. The

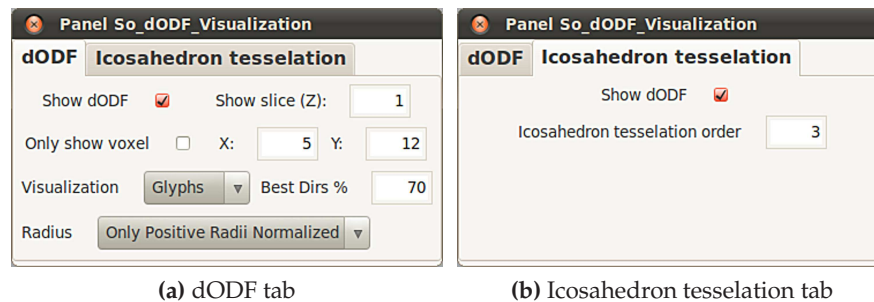


Figure A.5: GUI-panel of the *So_dODF_Visualization* module

displayed dODFs are normalized for better overview. Three different normalization modes are supported (see description of the GUI-panel for details). Every voxel can be rendered in a special background color to visualize specific properties of voxels. Furthermore, the corresponding B_0 -image of the visualized input image can be displayed in the background.

GUI-panel Both tabs of the GUI-panel of the *So_dODF_Visualization* module are shown in Figure A.5. The tab *dODF* provides access to the following module fields:

- **Show dODF:** If checked, the visualization of the dODF from the input image is enabled. This box should be checked when using the visualization option in the *dODF* tab and unchecked when using the visualization option in the *Icosahedron tessellation* tab.
- **Show slice (Z):** The slice of the input images that is displayed.
- **Only show voxel:** If checked, only one voxel of the selected slice is displayed. This voxel is determined by the values X and Y .
- **X:** The x-coordinate of the voxel to display in the case the box *Only show voxel* is checked.
- **Y:** The y-coordinate of the voxel to display in the case the box *Only show voxel* is checked.
- **Visualization:** The type of visualization. The user can choose between a visualization of the dODF as points, lines, meshes or glyphs. The last visualization option, *Best Dirs*, allows to display only the highest values of the dODF as lines. The threshold for these values is determined by *Best Dirs %*.
- **Best Dirs %:** Threshold for the selection of directions in the range $[0, 100]$. All dODF directions having at least this percentage of the







Scalar value	Color name	Sample
0	not handled	-
1	dark green	
2	dark blue	
3	dark red	
4	dark cyan	
5	dark magenta	
> 5	dark yellow	

Table A.2: The background colors for voxel classification that can be assigned to the voxels by a scalar value in the input image at the right input connector.

maximum diffusion direction value in the dODF are shown if *Visualization* is set to *Best Dirs*.

- **Radius:** The type of dODF radius normalization. Supported types are *Min Max Normalized*, *Normalized By ROI Max*, and *Only Positive Radii Normalized*. Further, *Zero* can be chosen to omit the dODF visualization.

The tab *Icosahedron tessellation* shows different tessellations of an icosahedron for general testing purposes of visualization. This tab provides access to the following module fields:

- **Show dODF:** Same as in the tab *dODF*. This box should be unchecked when using the visualization of the tab *Icosahedron tessellation*.
- *Icosahedron tessellation order:* If *Show dODF* is unchecked, this value determines the tessellation order of the icosahedron that is displayed in place of every voxel. This value is automatically set to the tessellation order of the dODF in the input image as soon as the box *Show dODF* is checked. All selectable option in *Visualization* in the tab *dODF* apart from *Best Dirs* also apply here.

Input This module has three input connectors. The main purpose of the module is the visualization of dODFs. An image containing the dODFs can be connected to the middle input connector. The right input connector serves as input to a voxel classification image. The left input connector is reserved for the B_0 -image that can be displayed in the background. The following description provides more details on the input images:

- **Middle input connector:** An image of extension $(x, y, z, c, t, u) = (X, Y, Z, 1, N, 1)$ holding the dODF values in dimension t . X, Y, Z is the size of the image in voxels. N is the number of used sampling directions for dODF reconstruction.

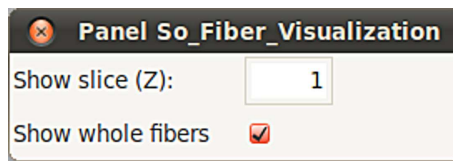


Figure A.6: GUI-panel of the *So_Fiber_Visualization* module

- **Right input connector:** This is an optional input. An image of extension $(x, y, z, c, t, u) = (X, Y, Z, 1, 1, 1)$. X, Y, Z are the same as in the input image at the middle input connector. The scalar value at each coordinate allows to classify the corresponding voxel as belonging to a certain group (e.g. a fiber or an anisotropy measure). Six different voxel groups can be indicated by this classification. Table A.2 shows the resulting color for the different classifications scalars. The classification color is drawn in the background of the dODF from the image at the middle input connector.
- **Left input connector:** This is an optional input. An image of extension $(x, y, z, c, t, u) = (X, Y, Z, 1, N, 1)$. X, Y, Z are the same as in the input image at the middle input connector. The size of dimension t does not matter as only the value at $t = 0$ is used in the module. The value of every voxel at the coordinate $t = 0$ is interpreted as a grayscale value and is displayed in the background of the other two input images. This is a convenient way of displaying the B_0 -image in the background of the rendered scene.

Output A rendered OpenGL scene that can be displayed with the *SoExaminerViewer* module.

So_Fiber_Visualization

Description This module visualizes the white matter fibers reconstructed by the *HARDI_deterministic_Tractography* module. Every voxel can be rendered in a special background color to visualize specific properties of voxels. Furthermore, the corresponding B_0 -image of the visualized input image can be displayed in the background.

GUI-panel The GUI-panel of the module *So_Fiber_Visualization* is shown in Figure A.6. It provides access to the following module fields:

- **Show slice (Z):** The slice of the input images that is displayed.
- **Show whole fibers:** If checked, every fiber trajectory is rendered in a bright color. If unchecked, only the parts of the fiber trajectories that

are in the selected slice are rendered in a bright color. The other parts of the fibers are displayed in a dark color.

Input This module has three input connectors. The main purpose of the module is the visualization of reconstructed fibers. An image containing the reconstructed fibers can be connected to the middle input connector. The right input connector serves as input to a voxel classification image. The left input connector is reserved for the B_0 -image that can be displayed in the background. The following description provides more details on the input images:

- **Middle input connector:** An image of extension $(x, y, z, c, t, u) = (1, 1, 1, N, 1)$ holding the reconstructed white matter fibers as an array of size N .
- **Right input connector:** This is an optional input. An image of extension $(x, y, z, c, t, u) = (X, Y, Z, 1, 1, 1)$. X, Y, Z is the number of voxels in the dimensions x, y , and z . The scalar value at each coordinate allows to classify the corresponding voxel as belonging to a certain group (e.g. a fiber or an anisotropy measure). Six different voxel groups can be indicated by this classification. Table A.2 shows the resulting color for the different classifications scalars. The classification color is drawn in the background of the fiber trajectories from the image at the middle input connector.
- **Left input connector:** This is an optional input. An image of extension $(x, y, z, c, t, u) = (X, Y, Z, 1, N, 1)$. X, Y, Z are the same as in the input image at the right input connector. The size of dimension t does not matter as only the value at $t = 0$ is used in the module. The value of every voxel at the coordinate $t = 0$ is interpreted as a grayscale value and is displayed in the background of the other two input images. This is a convenient way of displaying the B_0 -image in the background of the rendered scene.

Output A rendered OpenGL scene that can be displayed with the *SoExaminerViewer* module.

Bibliography

- [AB05] ALEXANDER, DC. ; BARKER, GJ.: Optimal imaging parameters for fibre-orientation estimation in diffusion MRI. In: *NeuroImage* (2005)
- [ABA02] ALEXANDER, D.C. ; BARKER, G.J. ; ARRIDGE, S.R.: Detection and Modeling of Non-Gaussian Apparent Diffusion Coefficient Profiles in Human Brain Data. In: *Magnetic Resonance in Medicine* 48 (2002)
- [AFRB04] ASSAF, Yaniv ; FREIDLIN, Raisa Z. ; ROHDE, Gustavo K. ; BASSER, Peter J.: New modeling and experimental framework to characterize hindered and restricted water diffusion in brain white matter. In: *Magnetic Resonance in Medicine* 52 (2004)
- [Ale05] ALEXANDER, Daniel C.: Multiple-Fiber Reconstruction Algorithms for Diffusion MRI. In: *Annals of the New York Academy of Sciences* 1064 (2005)
- [ALS09] AGANJ, Iman ; LENGLET, Christophe ; SAPIRO, Guillermo: ODF Reconstruction In Q-ball Imaging With Solid Angle Consideration. In: *IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2009. ISBI '09.* (2009)
- [And05] ANDERSON, Adam W.: Measurement of Fiber Orientation Distributions Using High Angular Resolution Diffusion Imaging. In: *Magnetic Resonance in Medicine* 54 (2005)
- [BAH⁺03] BATCHELOR, P.G. ; ATKINSON, D. ; HILL, D.L.G. ; CALAMANTE, F. ; CONNELLY, A.: Anisotropic noise propagation in diffusion tensor MRI sampling schemes. In: *Magnetic Resonance in Medicine* 49 (2003)
- [Bas97] BASSER, Peter J.: New histological and physiological stains derived from diffusion-tensor MR images. In: *Annals of the New York Academy of Sciences-Paper* (1997)

BIBLIOGRAPHY

- [BCM⁺08] BERMAN, Jeffrey I. ; CHUNG, SungWon ; MUKHERJEE, Pratik ; HESS, Christopher P. ; HAN, Eric T. ; HENRY, Roland G.: Probabilistic streamline q-ball tractography using the residual bootstrap. In: *NeuroImage* 39 (2008)
- [Bea02] BEAULIEU, Christian: The basis of anisotropic water diffusion in the nervous system a technical review. In: *NMR In Biomedicine* (2002)
- [BJB]⁺07a] BEHRENS, T.E.J. ; JOHANSEN-BERG, H. ; JBABDI, S. ; RUSHWORTH, M.F.S. ; WOOLRICH, M.W.: Probabilistic diffusion tractography with multiple fibre orientations: What can we gain? In: *NeuroImage* 34 (2007)
- [BJB]⁺07b] BEHRENS, T.E.J. ; JOHANSEN-BERG, H. ; JBABDI, S. ; RUSHWORTH, M.F.S. ; WOOLRICH, M.W.: Probabilistic diffusion tractography with multiple fibre orientations: What can we gain? In: *NeuroImage* 34 (2007)
- [BML94a] BASSER, Peter J. ; MATTIELLO, James ; LEBIHAN, Denis: Estimation of the Effective Self-Diffusion Tensor from NMR Spin Echo. In: *Journal of Magnetic Resonance* (1994)
- [BML94b] BASSER, Peter J. ; MATTIELLO, James ; LEBIHAN, Denis: MR Diffusion Tensor Spectroscopy and Imaging. In: *Biophysical Journal* 66 (1994)
- [BP96] BASSER, Peter J. ; PIERPAOLI, Carlo: Microstructural and Physiological Features of Tissues Elucidated by Quantitative-Diffusion-Tensor MRI. In: *Journal of Magnetic Resonance* (1996)
- [BPP⁺00] BASSER, Peter J. ; PAJEVIC, Sinisa ; PIERPAOLI, Carlo ; DUDA, Jeffrey ; ALDROUBI, Akram: In Vivo Fiber Tractography Using DT-MRI Data. In: *Magnetic Resonance in Medicine* (2000), Nr. 44
- [BWJ]⁺03] BEHRENS, T.E. ; WOOLRICH, M.W. ; JENKINSON, M. ; JOHANSEN-BERG, H. ; NUNES, R.G. ; CLARE, S. ; MATTHEWS, P.M. ; BRADY, J.M. ; SMITH, S.M.: Characterization and propagation of uncertainty in diffusion-weighted MR imaging. In: *Magnetic Resonance in Medicine* 50 (2003)
- [Cal94] CALLAGHAN, Paul T. ; PRESS, Oxford U. (Hrsg.): *Principles of nuclear magnetic resonance microscopy*. Oxford University Press, 1994
- [Cam04] CAMPBELL, Jennifer S. W.: *Diffusion Imaging of White Matter Fibre Tracts*, McGill University, Montreal, Canada, Diss., 2004

BIBLIOGRAPHY

- [CCC⁺08] CHAO, Yi-Ping ; CHEN, Jyh-Horng ; CHO, Kuan-Hung ; YEH, Chun-Hung ; CHOU, Kun-Hsien ; LIN, Ching-Po: A multiple streamline approach to high angular resolution diffusion tractography. In: *Medical Engineering & Physics* 30 (2008)
- [CGZ⁺04a] CHEN, Y. ; GUO, W. ; ZENG, Q. ; YAN, X. ; HUANG, F. ; ZHANG, H. ; HE, G. ; VEMURI, B. C. ; LIU, Y.: Estimation, smoothing, and characterization of apparent diffusion coefficient profiles from high angular resolution DWI. In: *Computer Vision and Pattern Recognition* 1 (2004)
- [CGZ⁺04b] CHEN, Yunmei ; GUO, Weihong ; ZENG, Qingguo ; HE, Guojun ; VEMURI, Baba ; LIU, Yijun: Recovery of intra-voxel Structure from HARD DWI. In: *IEEE International Symposium on Biomedical Imaging* (2004)
- [CLC⁺99] CONTURO, Thomas E. ; LORI, Nicolas F. ; CULL, Thomas S. ; AKBUDAK, Erbil ; SNYDER, Abraham Z. ; SHIMONY, Joshua S. ; MCKINSTRY, Robert ; BURTON, Harold ; RAICHLE, Marcus E.: Tracking neuronal fiber pathways in the living human brain. In: *Proc. Natl. Acad. Sci. USA* 96 (1999)
- [CP54] CARR, H. Y. ; PURCELL, E. M.: Effects of Diffusion on Free Precession in Nuclear Magnetic Resonance Experiments. In: *Physical Review* 94 (1954), Nr. 3
- [CSR⁺05] CAMPBELL, Jennifer S. ; SIDDIQI, Kaleem ; RYMAR, Vladimir V. ; SADIKOT, Abbas F. ; PIKE, G. B.: Flow-based fiber tracking with diffusion tensor and q-ball data: Validation and comparison to principal diffusion direction techniques. In: *NeuroImage* (2005)
- [DAFD06] DESCOTEAUX, Maxime ; ANGELINO, Elaine ; FITZGIBBONS, Shaun ; DERICHE, Rachid: Apparent Diffusion Coefficients from High Angular Resolution Diffusion Images: Estimation and Applications. In: *Magnetic Resonance in Medicine* 56 (2006)
- [DAFD07] DESCOTEAUX, Maxime ; ANGELINO, Elaine ; FITZGIBBONS, Shaun ; DERICHE, Rachid: Regularized, Fast, and Robust Analytical Q-Ball Imaging. In: *Magnetic Resonance in Medicine* 58 (2007)
- [DDKA09] DESCOTEAUX, Maxime ; DERICHE, Rachid ; KNÖSCHE, Thomas R. ; ANWANDER, Alfred: Deterministic and Probabilistic Tractography based on Complex Fibre Orientation Distributions. In: *IEEE Transactions on Medical Imaging* 28 (2009)

BIBLIOGRAPHY

- [Des07] DESCOTEAUX, Maxime: *High Angular Resolution Diffusion MRI: from Local Estimation to Segmentation and Tractography*, University of Nice-Sophia Antipolis, France, Diss., 2007
- [Ein05] EINSTEIN, Albert: Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen. In: *Annalen der Physik* (1905)
- [Fin07] FINK, Schneider ; FINK, Gereon R. (Hrsg.): *Funktionelle MRT in Psychiatrie und Neurologie*. Springer Medizin Verlag, springer.com, 2007
- [FPM09] FILLARD, Pierre ; POUPON, Cyril ; MANGIN, Jean-Francois: A Novel Global Tractography Algorithm based on an Adaptive Spin Glass Model. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2009* (2009)
- [Fra02] FRANK, Lawrence R.: Characterization of Anisotropy in High Angular Resolution Diffusion-Weighted MRI. In: *Magnetic Resonance in Medicine* 47 (2002)
- [Goh09] GOH, Alvina: Deterministic Tractography Using Orientation Distribution Functions Estimated With Probability Density Constraints And Spatial Regularity. In: *LNAO*, (2009)
- [Hah50] HAHN, Erwin: Spin Echoes. In: *Physical Review* (1950), 11
- [HMH⁺06] HESS, Christopher P. ; MUKHERJEE, Pratik ; HAN, Eric T. ; XU, Duan ; VIGNERON, Daniel B.: Q-Ball Reconstruction of Multimodal Fiber Orientations Using The Spherical Harmonic Basis. In: *Magnetic Resonance in Medicine* 56 (2006)
- [HZZJ10] H. Z. ZHANG, S. A. C. T. Martin McGinnity M. T. Martin McGinnity ; JING, M.: A Novel Criterion for Characterizing Diffusion Anisotropy in HARDI Data Based on the MDL Technique. In: *ICMB*, 2010
- [JA03] JANSONS, Kalvis M. ; ALEXANDER, Daniel C.: Persistent angular structure: new insights from diffusion magnetic resonance imaging data. In: *Inverse Problems* 19 (2003)
- [JBB09] JOHANSEN-BERG, Heidi ; BEHRENS, Timothy E. ; JOHANSEN-BERG, Heidi (Hrsg.) ; BEHRENS, Timothy E. (Hrsg.): *Diffusion MRI From Quantitative Measurement to in vivo Neuroanatomy*. Elsevier, 2009

BIBLIOGRAPHY

- [JLTS09] JEURISSEN, Ben ; LEEMANS, Alexander ; TOURNIER, Jacques-Donald ; SIJBERS, Jan: Fiber Tracking on the 'Fiber Cup Phantom' using Constrained Spherical Deconvolution. In: *LNAO* (2009)
- [Jon03] JONES, Derek K.: Determining and Visualizing Uncertainty in Estimates of Fiber Orientation From Diffusion Tensor MRI. In: *Magnetic Resonance in Medicine* 49 (2003)
- [Jon04] JONES, Derek K.: The Effect of Gradient Sampling Schemes on Measures Derived From Diffusion Tensor MRI: A Monte Carlo Study. In: *Magnetic Resonance in Medicine* 51 (2004)
- [Jon08] JONES, Derek K.: Tractography Gone Wild: Probabilistic Fibre Tracking Using the Wild Bootstrap With Diffusion Tensor MRI. In: *IEEE transactions on medical imaging* 27 (2008)
- [Kin04] KINDLMANN, Gordon: Superquadric Tensor Glyphs. In: *Proceeding of The Joint Eurographics - IEEE TCVG Symposium on Visualization* (2004)
- [LA03] LAZAR, Mariana ; ALEXANDER, Andrew L.: An error analysis of white matter tractography methods: synthetic diffusion tensor field simulations. In: *NeuroImage* 20 (2003)
- [LA05] LAZAR, Mariana ; ALEXANDER, Andrew L.: Bootstrap white matter tractography (BOOT-TRAC). In: *NeuroImage* 24 (2005)
- [LWT⁺03] LAZAR, Mariana ; WEINSTEIN, David M. ; TSURUDA, Jay S. ; HASAN, Khader M. ; ARFANAKIS, Konstantinos ; MEYERAND, M. E. ; BADIE, Benham ; ROWLEY, Howard A. ; HAUGHTON, Victor ; FIELD, Aaron ; ALEXANDER, Andrew L.: White Matter Tractography Using Diffusion Tensor Deflection. In: *Human Brain Mapping* 18 (2003)
- [MA95] MORNEBURG, Heinz ; ALEXANDRESCU, Mirca ; MORNEBURG, Heinz (Hrsg.): *Bildgebende Systeme für die medizinische Diagnostik*. Publicis MCD Verlag, Erlangen, 1995
- [MBLB97] MATTIELLO, James ; BASSER, Peter J. ; LE BIHAN, Denis: The b Matrix in Diffusion Tensor Echo-Planar Imaging. In: *Magnetic Resonance in Medicine* 37 (1997)
- [MCCZ99] MORI, Susumu ; CRAIN, Barbara J. ; CHACKO, V.P. ; ZIJL, Peter C. M.: Three-Dimensional Tracking of Axonal Projections in the Brain by Magnetic Resonance Imaging. In: *magnetic resonance imaging* 45 (1999)

BIBLIOGRAPHY

- [MSR09] MALCOLM, James G. ; SHENTON, Martha E. ; RATHI, Yogesh: Filtered Tractography: Validation on a Physical Phantom. In: *LNAO* (2009)
- [MZ02] MORI, Susumu ; ZIJL, Peter C. M.: Fiber tracking: principles and strategies - a technical review. In: *NMR In Biomedicine* 15 (2002)
- [OSV⁺06] OEZARSLAN, Evren ; SHEPHERD, Timothy M. ; VEMURI, Baba C. ; BLACKBAND, Stephen J. ; MARECI, Thomas H.: Resolution of complex tissue microarchitecture using the diffusion orientation transform (DOT). In: *NeuroImage* 31 (2006)
- [PA05] PARKER, Geoffrey J. M. ; ALEXANDER, Daniel C.: Probabilistic anatomical connectivity derived from the microscopic persistent angular structure of cerebral tissue. In: *Philosophical Transactions: Biological Sciences* 360 (2005)
- [PB03] PAJEVIC, Sinisa ; BASSER, Peter J.: Parametric and non-parametric statistical analysis of DT-MRI data. In: *Journal of Magnetic Resonance* 161 (2003)
- [PP99] PAJEVIC, S. ; PIERPAOLI, C.: Color Schemes to Represent the Orientation of Anisotropic Tissues From Diffusion Tensor Data: Application to White Matter Fiber Tract Mapping in the Human Brain. In: *Magnetic Resonance in Medicine* 42 (1999)
- [PPAM06] POUPON, C. ; POUPON, F. ; ALLIROL, L. ; MANGIN, J.-F.: A database dedicated to anatomo-functional study of human brain connectivity. In: *12th HBM Neuroimage*. Florence, Italie, 2006
- [PPC⁺05] PERRIN, M. ; POUPON, C. ; COINTEPAS, Y. ; RIEUL, B. ; GOLESTANI, N. ; PALLIER, C. ; RIVIÈRE, D. ; CONSTANTINESCO, A. ; LE BIHAN, D. ; MANGIN, J.-F.: Fiber tracking in q-ball fields using regularized particle trajectories. In: *Information Processing in Medical Imaging* (2005)
- [PRP⁺08] PRCKOVSKA, V. ; ROEBROECK, A.F. ; PULLENS, W.L.P.M. ; VILANOVA, A. ; HAAR ROMENY, B.M. t.: Optimal acquisition schemes in High Angular Resolution Diffusion Weighted Imaging. In: *Proceedings of MICCAI* (2008)
- [PVP⁺09] PRCKOVSKA, V. ; VILANOVA, A. ; POUPON, C. ; HAAR ROMENY, B.M. t. ; DESCOTEAUX, M.: Fast classification scheme for HARDI data simplification. In: *ICT Innovations 2009* (2009)

BIBLIOGRAPHY

- [PWKB02] PARKER, Geoffrey J. M. ; WHEELER-KINGSHOTT, Claudia A. M. ; BARKER, Gareth J.: Estimating Distributed Anatomical Connectivity Using Fast Marching Methods and Diffusion Tensor Imaging. In: *IEEE transactions on medical imaging* 21 (2002)
- [RMK09] REISERT, Marco ; MADER, Irina ; KISELEV, Valerij: Global Reconstruction of Neuronal Fibres. In: *Proceedings of MICCAI 2009*, (2009)
- [Rob00] ROBB, Richard A. ; WILEY, John (Hrsg.): *Biomedical Imaging, Visualization, and Analysis*. Wiley-Liss, 2000
- [Sak09] SAKAIE, Ken: Fast Persistent Angular Structure Based Streamline Tractography. In: *LNAO* (2009)
- [SDBS⁺08] SHERBONDY, Anthony J. ; DOUGHERTY, Robert F. ; BEN-SHACHAR, Michal ; NAPEL, Sandy ; WANDELL, Brian A.: ConTrack: Finding the most likely pathways between brain regions using diffusion tractography. In: *Journal of Vision* 8 (2008)
- [SKZ⁺01] STIELTJES, Bram ; KAUFMANN, Walter E. ; ZIJL, Peter C. M. ; FREDERICKSEN, Kim ; PEARLSON, Godfrey D. ; SOLAIYAPPAN, Meiyappan ; MORI, Susumu: Diffusion Tensor Imaging and Axonal Tracking in the Human Brainstem. In: *NeuroImage* 14 (2001)
- [ST65] STEJSKAL, E. O. ; TANNER, J. E.: Spin Diffusion Measurements: Spin Echoes in the Presence of a Time-Dependent Field Gradient. In: *The Journal of Chemical Physics* 42 (1965), 1, Nr. 1
- [TCC07] TOURNIER, J-Donald ; CALAMANTE, Fernando ; CONNELLY, Alan: Robust determination of the fibre orientation distribution in diffusion MRI: Non-negativity constrained super-resolved spherical deconvolution. In: *NeuroImage* 35 (2007)
- [TCGC04] TOURNIER, J.-Donald ; CALAMANTE, Fernando ; GADIAN, David G. ; CONNELLY, Alan: Direct estimation of the fiber orientation density function from diffusion-weighted MRI data using spherical deconvolution. In: *NeuroImage* 23 (2004)
- [TRW⁺02] TUCH, David S. ; REESE, Timothy G. ; WIEGELL, Mette R. ; MAKRIS, Nikos ; BELLIVEAU, John W. ; WEDEEN, Van J.: High Angular Resolution Diffusion Imaging Reveals Intravoxel White Matter Fiber Heterogeneity. In: *Magnetic Resonance in Medicine* 48 (2002)
- [Tuc04] TUCH, David S.: Q-Ball Imaging. In: *Magnetic Resonance in Medicine* 52 (2004)

BIBLIOGRAPHY

- [TYC⁺08] TOURNIER, J.-Donald ; YEH, Chun-Hung ; CALAMANTE, Fernando ; CHO, Kuan-Hung ; CONNELLY, Alan ; LIN, Ching-Po: Resolving crossing fibres using constrained spherical deconvolution: Validation using diffusion-weighted imaging phantom data. In: *NeuroImage* 42 (2008)
- [VZKL06] VILANOVA, A. ; ZHANG, S. ; KINDLMANN, G. ; LAIDLAW, D.: An Introduction to Visualization of Diffusion Tensor Imaging and Its Applications. In: *Visualization and Processing of Tensor Fields* (2006)
- [WKL99] WEINSTEIN, David ; KINDLMANN, Gordon ; LUNDBERG, Eric: sensorlines: Advection-Diffusion based Propagation through Diffusion Tensor Fields. In: *10th IEEE Visualization 1999* (1999)
- [WPG⁺97] WESTIN, C.F. ; PELED, S. ; GUDBJARTSSON, H. ; KIKINIS, R. ; JOLESZ, F. A.: Geometrical Diffusion Measures for MRI from Tensor Basis Analysis. In: *Proceedings of the 5th annual meeting of ISMRM* (1997)
- [WRT⁺00] WEDEEN, V.J. ; REESE, T.G. ; TUCH, D.S. ; WEIGEL, M.R. ; DOU, J.-G. ; WEISKOFF, R.M. ; CHESSLER, D.: Mapping fiber orientation spectra in cerebral white matter with Fourier-transform diffusion MRI. In: *Proc. Intl. Sot. Mag. Reson. Med.* 8 (2000)
- [WWS⁺08] WEDEEN, V.J. ; WANG, R.P. ; SCHMAHMANN, J.D. ; BENNER, T. ; TSENG, W.Y.I. ; DAI, G. ; PANDYA, D.N. ; HAGMANN, P. ; DÁRCEUIL, H. ; CRESPIGNY, A.J. de: Diffusion spectrum magnetic resonance imaging (DSI) tractography of crossing fibers. In: *NeuroImage* 41 (2008)
- [ZB02] ZHUKOV, Leonid ; BARR, Alan H.: Oriented Tensor Reconstruction: Tracing Neural Pathways from Diffusion Tensor MRI. In: *Visualization, 2002. VIS 2002. IEEE* (2002)
- [ZHGG09] ZHANG, Fan ; HANCOCK, Edwin R. ; GOODLETT, Casey ; GERIG, Guido: Probabilistic White Matter Fiber Tracking using Particle Filtering and von Mises-Fisher Sampling. In: *Medical Image Analysis* 13 (2009)