



Harvard Medical School / University of Heidelberg / Heilbronn University
Medical Informatics

Thesis

Inspection and Editing Tool for Diffusion Weighted MRI

submitted by Kerstin Kessel
August 1st 2008

Advisors

Ron Kikinis, M.D.

(Director of the Surgical Planning Laboratory, Brigham and Women's Hospital, Boston)

Steve Pieper, Ph.D.

(Surgical Planning Laboratory, Brigham and Women's Hospital, Boston)

Prof. Dr.-Ing. Hartmut Dickhaus

(Department of Medical Informatics, University of Heidelberg)

Acknowledgements

I would like to thank Prof. Dr.-Ing. Hartmut Dickhaus for introducing me to the Surgical Planning Laboratory (SPL) and offering the possibility to write this thesis abroad in Boston, USA.

Thank you Ron Kikinis and Steve Pieper for being great advisors. Your honest advice and encouragement brought out the best in me. I thank you Ron for the ability to work on the project at the SPL, Brigham and Women's Hospital, Harvard Medical School, Boston. Special thanks to you Steve for your guidance, ideas and suggestions on how to proceed. Thank you Sylvain Bouix, Raul San-Jose Estepar and Martin Styner for your help and giving me all the background information to motivate my project.

I gratefully acknowledge the Lions Club Heilbronn / Franken for their scholarship and financial support.

Many thanks to all my friends from at home and the US for their support and encouragement. Specifically, thank you Andreas Freudling for always explaining and being willing to help getting settled at the beginning. Thank you Ben, for being a great friend especially the last four months when we were the only students at the SPL. Thank you Katharina Quintus and Anja Kroch for encouraging me all the time and having an open door for just talking. Thank you Harrison Lee for hosting me in your apartment in the famous Donna Summer house. At last, I thank my best friend Nina for being there for me all the time.

Special thanks go to you Mom and Dad, for believing in me and encouraging me to always reach for the best. Thank you for the unconditional support, without this stay abroad would not have been possible. Thank you my sisters and little brother, especially Carmen for visiting me and reminding me to do my best.

Abstract

DICOM is the generally used standard for diffusion weighted magnetic resonance imaging (DW-MRI), but for the acquisition parameters it is defined imprecisely. First, some parameters are not standardized at all like the gradient measurement frame, diffusion tensors and moreover, vendors use their own methods for encoding. These facts require a strict quality control of the data. To support this, we present the implementation and use of an open-source Diffusion Editor module for researchers to process diffusion weighted image (DWI) and diffusion tensor image (DTI) data. As a tool to easily adjust and add missing acquisition values, it also provides the opportunity to confirm the geometric accuracy visually, using tensor estimation, tensor glyphs and fiducial seeding tractography. We demonstrate applications to measure the qualitatively impact of wrong acquisition parameters by analyzing real tensor images. In summary, our Diffusion Editor provides a handy tool to improve and optimize the workflow of DWI editing and thereby facilitates the quality control.

Contents

List of Abbreviations	V
List of Figures	VI
List of Listings	VII
1. Introduction	1
2. Background and State of the Art	4
2.1. Diffusion Weighted Imaging	4
2.2. Diffusion Tensor Imaging	7
2.3. Visualization of Diffusion Tensors	9
2.3.1. Tensor Glyphs	10
2.3.2. Tractography	10
2.4. Diffusion Storage and Representation	11
2.4.1. DICOM File Format	11
2.4.2. NRRD File Format	13
3. Diffusion Editor	15
3.1. Concept and Implementation	15
3.1.1. Parsing Parameters	16
3.1.2. Tensor Estimation Module	17
3.1.3. Tractography Module	19
3.1.4. Tensor Glyphs	19
3.1.5. Undo/Redo Mechanism	19
3.1.6. User Interface	20
3.2. Functionality	22
3.2.1. Editing Measurement Frame	22
3.2.2. Editing Gradients	23

3.2.3. Exploring	24
3.3. Application Examples	25
3.3.1. Tensor Rotation	25
3.3.2. Tensor Flipping	27
3.3.3. Gradient Editing	28
3.4. Tutorial Videos	29
4. Discussion	30
4.1. Conclusion	30
4.2. Outlook	31
Bibliography	32
A. Code Extracts	i
A.1. ParseGradientsBvalueToArray	i
A.2. ParseGradientsBvalueToString	iii
B. Content of the enclosed CD-ROM	v

List of Abbreviations

ADC	Apparent Diffusion Coefficient
CLM	Command Line Module
CT	Computed Tomography
DICOM	Digital Imaging and Communications in Medicine
DTI	Diffusion Tensor Imaging
DWI	Diffusion Weighted Imaging
DW-MRI	Diffusion Weighted Magnetic Resonance Imaging
FA	Fractional Anisotropy
GUI	Graphical User Interface
HIV	Human Immunodeficiency Virus
LPS	Left-Posterior-Superior
MRI	Magnetic Resonance Imaging
NA-MIC	National Alliance for Medical Imaging Computing
NRRD	Nearly Raw Raster Data
PACS	Picture Archiving and Communication Systems
RAS	Right-Anterior-Superior
RF	RadioFrequency
ROI	Region Of Interest
SNR	Signal-to-Noise Ratio
SP	Seeding Point
TE	Time Echo
VTK	Visualization ToolKit

List of Figures

1.1. Diffusion Editor in the 3D Slicer environment	2
2.1. Spin echo	5
2.2. Water anisotropy	6
2.3. Visualization methods tensor glyphs and tractography	9
2.4. Schematic tractography	11
2.5. Relationship between the coordinate frames of diffusion gradients, DWI/DTI image orientation and world space	13
3.1. Flow chart: measurement frame processing	16
3.2. Flow chart: gradient and b-value processing	17
3.3. Undo/redo and restore mechanism	20
3.4. Diffusion Editor interface	21
3.5. Measurement frame panel	22
3.6. Gradients panel	23
3.7. Testing panel	24
3.8. Tractography with 3 degrees rotation of the measurement frame	26
3.9. Tensor flipping from LPS to RAS	27
3.10. Tractography with rounded gradients	28

List of Listings

2.1. Example NRRD file header	14
A.1. ParseGradientsBvalueToArray	i
A.2. ParseGradientsBvalueToString	iii

1

Introduction

When Lauterbur [21] and Mansfield [24] introduced the modern magnetic resonance imaging (MRI) to the diagnostic medicine in 1973, nobody realized what will be possible with this technology someday. Today MRI holds an important clinical role due to the fact that is not comparable with alternative methods like X-Ray and computed tomography (CT). Primary advantages lie in its high image resolution and contrast, and that patients are not exposed to radiation. Furthermore, it provides a way for in vivo non-invasively observation of superficial as well as deep organs [14]. Many applications for MRI exist and increasingly extend its diagnostic capabilities. So does the diffusion weighted magnetic resonance imaging (DW-MRI) - a technique introduced in the 1980s, which measures diffusion of water molecules in tissue. Diffusion weighted imaging (DWI) and its sub-techniques (e.g. diffusion tensor imaging (DTI), tractography) are most commonly used in neuroradiology. For example, DWI has been well established in brain research especially neuropsychiatric disorders like schizophrenia [19; 31], Alzheimer disease [18], tumor research [23] and stroke [8].

To represent DWI data sets the DICOM¹ standard is used. This is done by MRI scanners generating image files with a DICOM header containing the acquisition parameters. However, calling it standard may not be the correct word in this case. DICOM is not well defined when it comes to these acquisition parameters of DWI. First, some parameters are not standardized at all like the gradient measurement frame and diffusion tensors. Moreover, vendors and researchers are not careful to

¹Digital Imaging and Communications in Medicine

encode parameters in a standard way and prefer their own unique methods. Even for a single manufacturer the encoding depends on the used MRI scanner software version and can consequently be quite different.

Regarding to the scanner software, there are further issues. We observed, if a custom DWI acquisition is used, it is possible that the values in the DICOM header are not correct. There is a separate text file created with the diffusion gradient directions used during the scan, whereas the values in the DICOM header are usually extracted from the protocol definition of the scanner software, not from the text file. This leads to a possible mismatch and increases the error-proneness.

These facts require a strict quality control of the data. To support this, the goal of this thesis is to develop a Diffusion Editor module (figure 1.1) for researchers to process diffusion weighted and diffusion tensor image data and give a tool to easily adjust and add missing acquisition parameters. An additional focus lies on the integration of existing modules for diffusion tensor estimation, tensor glyphs and fiducial seeding tractography, to visually confirm the geometric correctness.

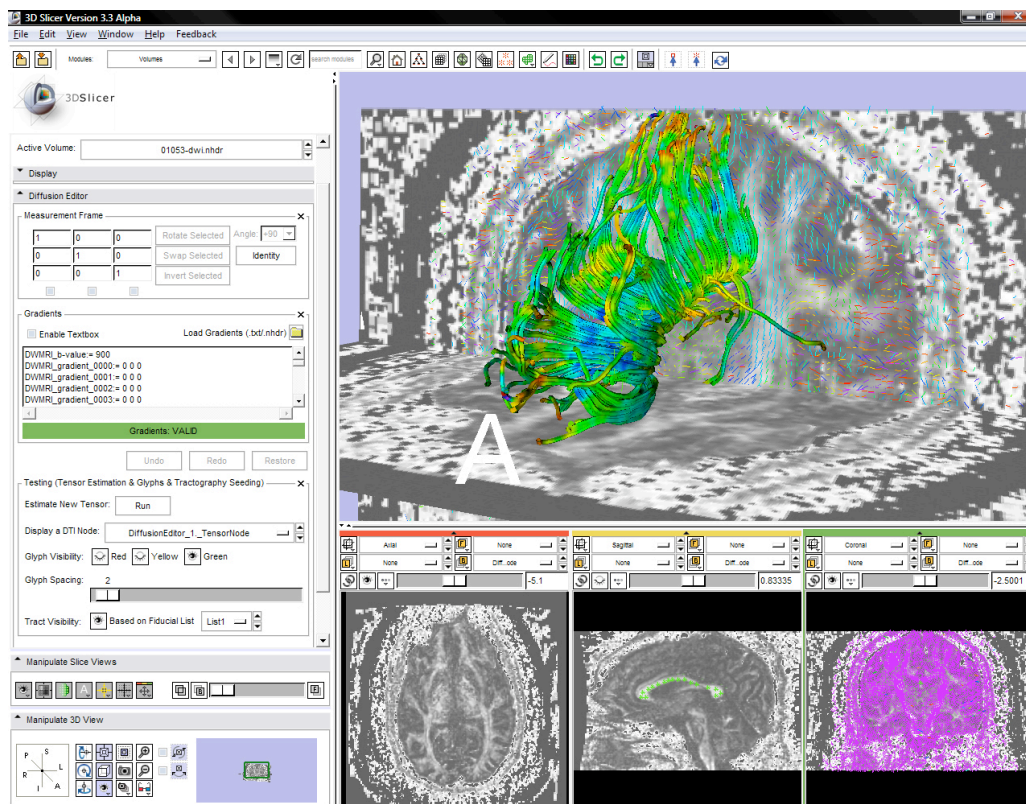


Figure 1.1.: Diffusion Editor in the 3D Slicer environment displaying a MRI brain scan showing fiducial seeding tractography along the corpus callosum and tensor glyphs in the coronal view.

We implement the module within the open source framework Visualization ToolKit (VTK) [4]. It is a 3D computer graphics, image processing, and visualization software which is currently used in medical applications. Furthermore we incorporate the Diffusion Editor into the 3D-Slicer environment [7]. As an open source and multi-platform software for neuroscience and image guided therapy, 3D Slicer provides a comprehensive tool for DWI and DTI processing. Therefore the Diffusion Editor will be of great avail for the research community.

In this thesis we describe the motivation and implementation of the Diffusion Editor followed by a demonstration and analysis of its use. First we give an overview on principles of diffusion weighted and diffusion tensor imaging, as well as the visualization methods tensor glyphs and tractography. We also outline the state of the art on diffusion data storage and representation. In this context the NRRD² file format is introduced. The next chapter describes in detail the concept and implementation as well as the functionality. We show the Diffusion Editor as a tool to measure the impact of wrong acquisition parameters qualitatively and demonstrate its use by analyzing tensor images. We conclude by discussing the improvement of the overall workflow of DWI editing and present an outlook on further development.

²Nearly Raw Raster Data

2

Background and State of the Art

In this chapter, we give a review on diffusion weighted (DWI) and diffusion tensor imaging (DTI) and explain the visualization methods tensor glyphs and tractography. Further, the current technique on how diffusion data is stored and the issues this procedure implicates, are examined. We will prove that the DICOM standard is insufficient to store the correct and complete acquisition parameters, which are necessary for diffusion tensor estimation and further DTI analysis. This fact requires a tool for subsequent editing.

2.1. Diffusion Weighted Imaging

Introduced in 1986 by Le Bihan et al. [13] diffusion weighted imaging (DWI) has become an important modality in clinical neuroimaging. It is working similar to anatomical MRI, but additionally captures diffusion caused by the Brownian motion of water molecules during the imaging process.

The conventional T2-weighted spin-echo imaging sequence uses two radiofrequency (RF) excitation pulses at 90° and 180° . The 180° refocusing RF is applied to rise the spin echo, which is detected at TE^3 (figure 2.1, (a)).

The DWI technique applies additional magnetic field gradient pulses around the 180° refocusing pulse (figure 2.1, (b)). The resulting effect is a different spin precession of the water molecules depending on their location. After the gradient field is turned

³Time Echo

off the spins relax to the original magnetic field.

The self diffusion of water molecules causes displacement during the time of two gradient pulses (Δ). This results in a destructive interference of the spin phase, which means spins are not completely refocused and in turn create a signal loss [19]. The amount of signal loss provides a quantitative measure of diffusion in the gradient direction. Eventually, that means one single DWI image detects only diffusion which occurs along the direction of the applied gradient.

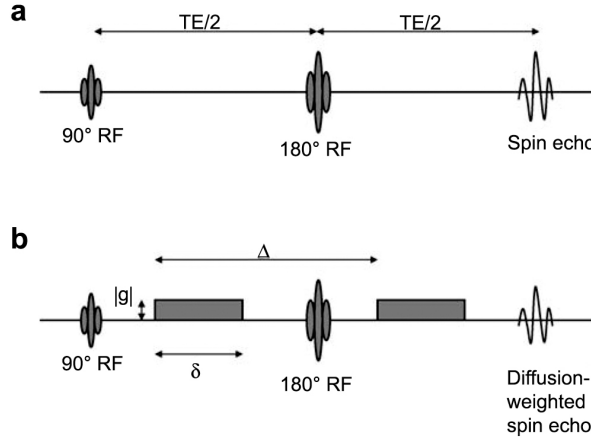


Figure 2.1.: (a) Spin echo formed by applying 90° ($t=0$) and 180° ($t=TE/2$) pulses; echo is detected at $t=TE$. (b) Two magnetic field gradients are added (rectangles) transforming the spin echo into a diffusion weighted spin echo. (from [28, p. 741])

Stejskal and Tanner [30, 1965] defined the following equation to calculate diffusion

$$S = S_0 e^{-bD} \quad (2.1)$$

where S is the signal with the gradient application, S_0 is the signal intensity without diffusion weighting, b is the diffusion weighted factor and D is a single scalar. The b -value is defined by Le Bihan et al. [13, 1986] as

$$b = \gamma^2 \delta^2 \left(\Delta - \frac{\delta}{3} \right) |g|^2 \quad (2.2)$$

where γ is a constant called gyromagnetic ratio (42 MHz/Tesla), δ is the gradient pulse duration, Δ is the separation of both and $|g|$ is the strength (amplitude).

In tissues, like brain gray matter, where the measured diffusion is equally and independent of the orientation of the tissue (isotropic), the diffusion characteristics are

described by a single scalar apparent diffusion coefficient (ADC). Diffusion of water molecules in tissues, like white matter and cardiac muscle, is not equally, in fact it is a three dimensional process [10; 12]. This effect is called *anisotropic diffusion* and can be measured by applying multiple field gradients. At least three measurements (in three different directions) are needed to calculate the mean diffusion for each voxel (figure 2.2, (b)).

Later Basser et al. [11, 1994] developed a multi-dimensional approach for estimation of diffusion data, called diffusion tensor imaging (DTI). A single scalar is no longer enough to describe this complex model, that is why each voxel is now described by a 3x3 matrix, the *diffusion tensor*, which fully characterizes the molecular mobility along each direction and correlation between these directions. This tensor can be visualized as an ellipsoid (figure 2.2, middle row) with its eigenvectors defining the orientation of the anisotropic diffusion, and its eigenvalues (λ_1 , λ_2 and λ_3) representing the magnitude of diffusion in this direction.

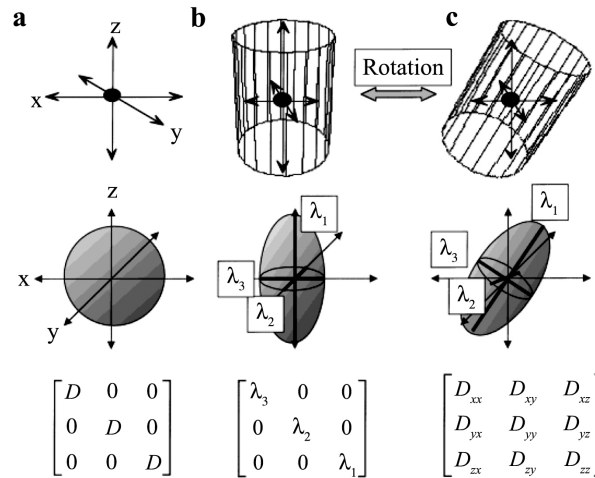


Figure 2.2.: Relationship between anisotropic diffusion (upper row), diffusion ellipsoids (middle row) and diffusion tensor (bottom row). In an isotropic (a) environment, water diffusion is symmetrical in all directions. The diffusion ellipsoid is spherical and can be described by a single diffusion constant \mathcal{D} . In an anisotropic medium, water diffusion is different along each axis. The diffusion ellipsoid of an cylindrical environment (b) has three principal axis (eigenvalues of \mathcal{D}) λ_1 , λ_2 and λ_3 . By applying an additional rotation to a system (c) a 3x3 tensor is needed to give a full characterization of the diffusion. (from [25, p. 107])

To quantitatively measure the degree of diffusion anisotropy several indices are used. The most popular ones are the scalars for trace, mean diffusion and fractional anisotropy (FA) and calculated as follows:

$$\text{trace}(\mathcal{D}) = \lambda_1 + \lambda_2 + \lambda_3 \quad (2.3)$$

$$\bar{\lambda} = \frac{\lambda_1 + \lambda_2 + \lambda_3}{3} = \frac{\text{trace}(\mathcal{D})}{3} \quad (2.4)$$

$$FA = \sqrt{\frac{3}{2}} \sqrt{\frac{(\lambda_1 - \bar{\lambda})^2 + (\lambda_2 - \bar{\lambda})^2 + (\lambda_3 - \bar{\lambda})^2}{(\lambda_1)^2 + (\lambda_2)^2 + (\lambda_3)^2}} \quad (2.5)$$

FA measures the difference between the largest eigenvalue and the two smaller eigenvalues. Applied on a DTI it results in a scalar map, where pixel range from 0 (isotropic $\lambda_1 = \lambda_2 = \lambda_3$, black pixel) to 1 (fully anisotropic $\lambda_2 = 0, \lambda_3 = 0$, white pixel) [14].

2.2. Diffusion Tensor Imaging

In DTI each voxel of the image data is represented by a tensor. This tensor describes the local water diffusion in the form of an ellipsoid, with its longest axis pointing towards the direction of the main diffusion.

The mathematical construct of a diffusion tensor is a 3x3 symmetric positive definite matrix. Since it is a symmetric matrix, only six unique elements are required to fully characterize the tensor

$$\mathcal{D} = \begin{pmatrix} \mathcal{D}_{xx} & \mathcal{D}_{xy} & \mathcal{D}_{xz} \\ \mathcal{D}_{yx} & \mathcal{D}_{yy} & \mathcal{D}_{yz} \\ \mathcal{D}_{zx} & \mathcal{D}_{zy} & \mathcal{D}_{zz} \end{pmatrix} \quad (2.6)$$

where $\mathcal{D}_{ij} = \mathcal{D}_{ji}$, $i, j = x, y, z$ [9; 14].

Refining Eq. 2.1 for tensor estimation we get

$$S_i = S_0 e^{-bg_i^T \mathcal{D} g_i} \quad (2.7)$$

where S_i is the signal intensity with the gradient application g_i , S_0 is the baseline, representing the signal intensity without diffusion weighting ($g_0 = 0$) and \mathcal{D} is the diffusion tensor [17; 32].

As we know six values characterize the tensor \mathcal{D} , so we need at least six measurements S_i , $i = 1..6$ and the baseline image S_0 to calculate the tensor. That means for the acquisition, seven separate images have to be made for every single slice of the image data. Each image showing diffusion in the applied gradient direction. (e.g. for a brain scan divided into 85 layers, $85 * 6 = 510$ separate images have to be acquired)

Putting these information in Eq. 2.7, results in the equation system

$$\begin{aligned}
 \ln(S_1) &= \ln(S_0) - bg_1^T \mathcal{D}g_1 \\
 \ln(S_2) &= \ln(S_0) - bg_1^T \mathcal{D}g_2 \\
 \ln(S_3) &= \ln(S_0) - bg_1^T \mathcal{D}g_3 \\
 \ln(S_4) &= \ln(S_0) - bg_1^T \mathcal{D}g_4 \\
 \ln(S_5) &= \ln(S_0) - bg_1^T \mathcal{D}g_5 \\
 \ln(S_6) &= \ln(S_0) - bg_1^T \mathcal{D}g_6
 \end{aligned} \tag{2.8}$$

The diffusion tensor \mathcal{D} can now be calculated by solving the system for each voxel in the image data [32].

However, generally more images than the minimum of seven (six gradients and one baseline) are acquired to improve the low signal-to-noise ratio (SNR) of DWI images [17]. There are two ways to do so. First, register many gradients in different directions. Or secondly, repeat already applied gradients and average the images. Yet the study from Goodlett et al. [17, 2007] documents that the second approach should be preferred, as repeated gradients show a statistical bias in DTI measurements.

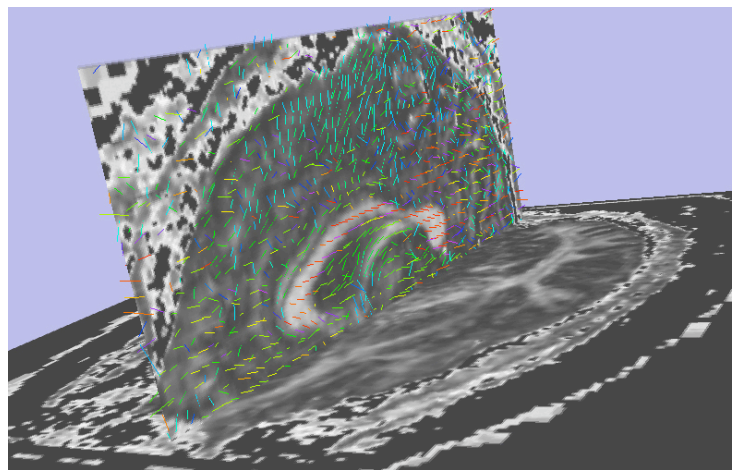
Consequently, the calculation of \mathcal{D} may quickly become more complex. Therefore the faster method of linear regression is preferred. It can be expressed by minimizing the linear least-squares problem [27]

$$\min_D \sum_k \log \left(\frac{S_k}{S_0} + bg_k^T \mathcal{D}g_k \right)^2 \tag{2.9}$$

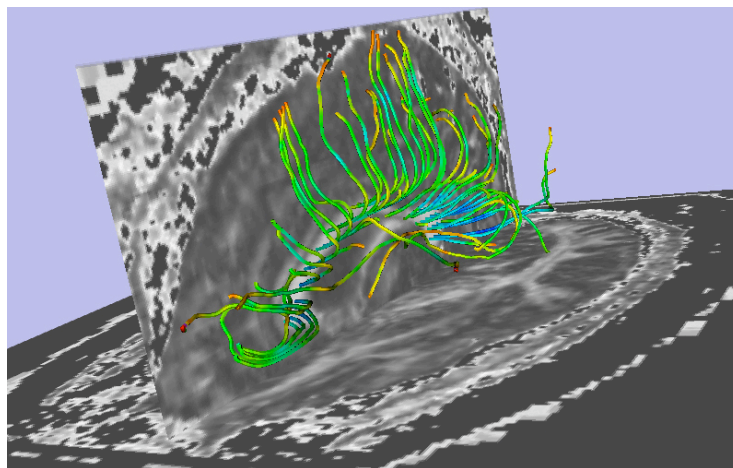
In this thesis we don't look at the tensor estimation in detail, but it should be mentioned that further algorithms for tensor estimation are used, such as the least-squares, weighted least-squares and non-linear methods.

2.3. Visualization of Diffusion Tensors

Diffusion tensors are used to characterize quantitatively the local structure and connectivity of the nerve tissue, such as white matter in the brain. It can be visualized by DWI because of the water diffusion along myelinated nerve fibers. A wide variety of techniques exists to accomplish that. Tensor glyphs and tractography (see figure 2.3) are the primary methods and are described in more detail in the following sections.



(a) tensor glyphs



(b) tractography

Figure 2.3.: Tensor glyphs and tractography generated from the same DTI data set. (a) shows line glyphs on the sagittal slice, colored by orientation. Red means diffusion left \leftrightarrow right, green anterior \leftrightarrow posterior and blue superior \leftrightarrow inferior. Looking at the red glyphs around the corpus callosum, we see that the tracts in (b) run exactly in left \leftrightarrow right direction in this ROI.

2.3.1. Tensor Glyphs

An intuitive way for tensor visualization are glyphs. Transform functions render from the tensor information geometric shapes, for example ellipsoids, lines or superquadrics⁴ [15]. These are displayed for each voxel in the image and provide diffusion information in a local region of interest (ROI). Attributes, like color, opacity, scale, and sampling rate, can easily be modified.

2.3.2. Tractography

Instead of visualizing diffusion in one single voxel, tractography provides information across multiple voxels showing the global connectivity of fiber bundles. It is assumed that the direction of the fibers is collinear with the direction of the major eigenvector of diffusion ellipsoids (see figure 2.2) [26].

Two major ways to build tracts exist. First, the streamline method; starting from a seeding point (also called fiducial) a line is propagated according to the pixel's direction of the main eigenvector, see figure 2.4. Second, the stochastic techniques, which includes for example the energetic property of next potential pixels or motion and noise of the data into the calculation of tracts [16].

Furthermore, several techniques for both approaches are well-established, such as fiducial and ROI seeding, or bootstrap and bayesian stochastic methods [22; 16].

Recently, many studies using tractography researched fiber damage in the brain with great results. Meanwhile, tractography established itself as an essential tool for stroke exploration [29] or early detection abnormalities of brain tissue [20]. Wu et al. [33] used DTI to study how HIV⁵ infection affects diffusion within the brain and among other things leads to dementia and motor speed losses. But these are just a few examples showing the possibility of tractography in terms of medical diagnosis and treatment in neuronal diseases. The continuing revolution of new methods and algorithms in DT-MRI and tractography develop them to standard clinical procedures.

⁴mathematical 3-dimensional objects for geometric modeling

⁵Human Immunodeficiency Virus

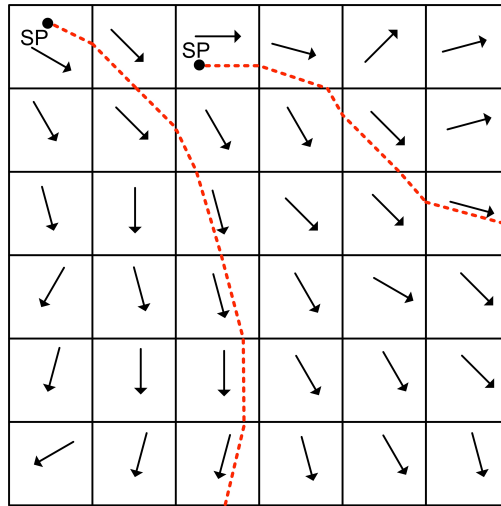


Figure 2.4.: This schematic diagram shows the streamline approach. Each pixel represents a tensor with its major eigenvector shown as an arrow. Starting at a seeding point (SP) the path follows the direction of the arrow until the next pixel is entered. Then the direction is calculated again by adjusting to the arrow (eigenvector) of the new pixel. This tracking is repeated until the anisotropy of the next pixel is less than a predefined threshold. Note that in reality this is a 3-dimensional process. (reproduced from [28, p. 747])

2.4. Diffusion Storage and Representation

In this section, we outline the DICOM framework in terms of diffusion imaging and characterize the occurring issues. As an alternative the NRRD file format is introduced, which is mainly used in the 3D-Slicer environment to represent diffusion data.

2.4.1. DICOM File Format

The DICOM framework is the standard to use when it comes to medical imaging. It defines a network communication protocol as well as a file format. With that, and the possibility to integrate workstations, servers and other network hardware into a picture archiving and communication systems (PACS) of radiological institutions, it is easy to exchange information and operate with the data.

In the last DICOM release the following tags for diffusion weighted imaging (DWI) are defined. [2, p.27]

(0018,9087) Diffusion b-value	FD 1
(0018,9089) Diffusion Gradient Orientation	FD 3

This deficient specification raises several issues. First, the DICOM tags, which are defined (b-value and gradients) are not used by vendors and researchers, probably because the standard was released after they had created their own non-standard implementations. It occurs, the DICOM header does not match the real acquisition information, due to rounding effects. (We assume that historical reasons causes this special effect, when MRI software developers decided to round values to make a display nicer.) It also has been found that important gradient information is not stored at all or written in hidden tags of the DICOM header.

Not knowing the vendors protocol of saving the acquisition parameters leads to extensive problems in DWI / DTI processing for researchers. And if we think one step further, it can have even worse effects in surgery planning.

The second serious problem is the fact, that no DICOM tags for DTI are defined. That means estimated tensors can not be saved in the DICOM header of images. Furthermore, there is no standard how to record the used measurement frame of diffusion gradients properly. DICOM defines that the gradients are specified in image coordinates IJK ⁶, hence the measurement frame is a constant and therefore not explicit in the header.

For the calculation of tensors it is important that diffusion gradients and image orientation are mapped to the same coordinate frame. Figure 2.5 shows the relation between the image coordinates, diffusion gradient coordinates and the mapping to the world space or anatomical coordinate frame in right-anterior-superior (RAS). The image coordinate transformation from $IJK \rightarrow RAS$ needs to know the space directions (pixel size and thickness) and space origin (position of the first pixel in the first image). Both parameters are available from DICOM. To map the diffusion gradients from $XYZ \rightarrow RAS$, we also need to know the measurement frame. But this information is not stored in DICOM.

As long as DICOM is vendor-specific and does not support DWI-specific acquisition parameters, an alternative way has to be found. The NRRD file format from Gordon Kindlmann [6] provides such a possibility to represent diffusion data. It is currently limited to a small group of researches associated with NA-MIC⁷ and consequently used in 3D-Slicer, but this advanced research community is likely to influence future adoption of the format.

⁶I,J,K are the labels of the grid, similar to the X,Y,Z labels of the gradient coordinate axis

⁷The National Alliance for Medical Imaging Computing (NA-MIC) USA is a multi-institutional, interdisciplinary team of computer scientists, software engineers, and medical investigators who develop computational tools for the analysis and visualization of medical image data. [5]

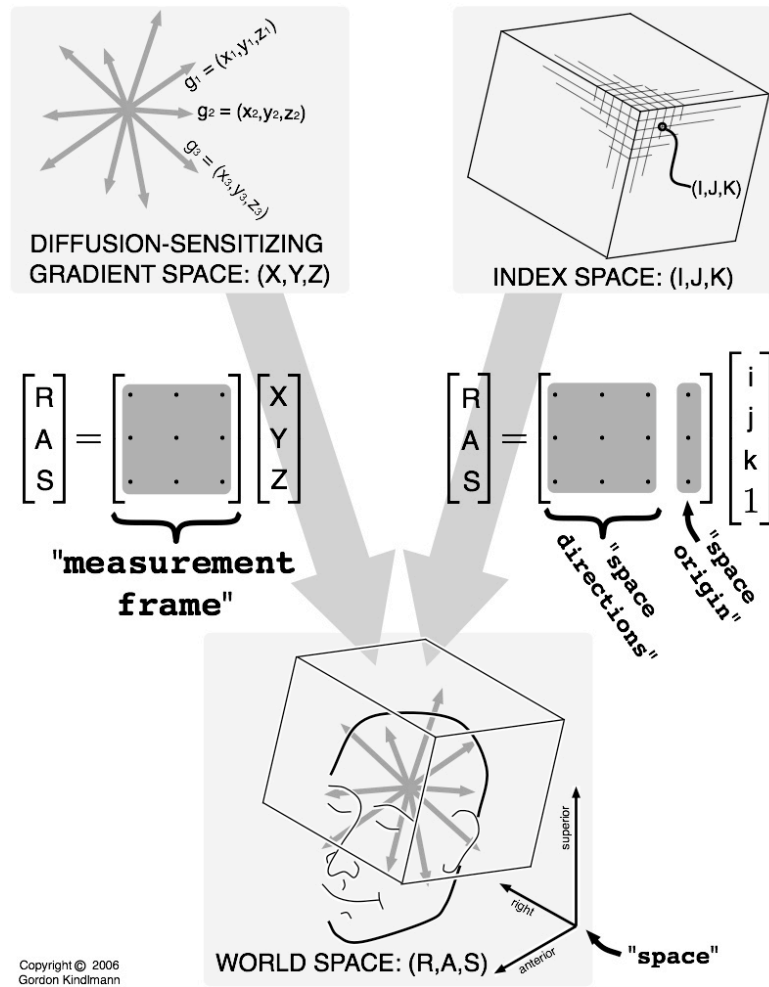


Figure 2.5.: Relationship between the coordinate frames of diffusion gradients, DWI/DTI image orientation and world space (also called patient space) in *RAS*. The first transformation matrix needs the gradient measurement frame to calculate $XYZ \rightarrow RAS$. Furthermore, space direction and space origin have to be known for the second matrix to transform the image from $IJK \rightarrow RAS$. (Subject to the copyright of Gordon Kindlmann)

2.4.2. NRRD File Format

The NRRD file format can be seen as the solution for the problem of describing DWI data sets. The important point to make is that it is uniquely able to store diffusion tensors and gradient measurement frames. As a file format it can also be used for non-diffusion data.

The flexible NRRD format includes a single header file and image files, which are available as a packed archive or single files. Listing 2.1 shows an example header containing all acquisition parameters. We want to take a closer look at the important fields concerning the problems from the previous section. According to line 7, the

data set uses RAS coordinates. Line 10 and 16 show the necessary parameters *space direction* and *space origin* to transform the image orientation to RAS, see Eq. 2.10. Also the *measurement frame* is available in line 18 for the transformation of the gradients, see Eq. 2.11. Line 21-27 defines these gradients (in this case six gradients and one baseline) with the b-value in line 20.

$$\begin{pmatrix} R \\ A \\ S \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (2.10)$$

$$\begin{pmatrix} R \\ A \\ S \end{pmatrix} = \begin{pmatrix} -0.859375 & 0 & 0 & 117.5 \\ 0 & 0 & -0.859375 & -93 \\ 0 & 5 & 0 & 119 \end{pmatrix} \begin{pmatrix} I \\ J \\ K \\ 1 \end{pmatrix} \quad (2.11)$$

```

1  NRRD0005
2  # Complete NRRD file format specification at:
3  # http://teem.sourceforge.net/nrrd/format.html
4  content: caseD20
5  type: short
6  dimension: 4
7  space: right-anterior-superior
8  sizes: 256 256 30 7
9  thicknesses: NaN NaN 4 NaN
10 space directions: (-0.859375,0,0) (0,0,-0.859375) (0,5,0) none
11 centerings: cell cell cell none
12 kinds: space space space list
13 endian: big
14 encoding: gzip
15 space units: "mm" "mm" "mm"
16 space origin: (117.5,-93,119)
17 data file: caseD20_DWI.raw.gz
18 measurement frame: (0,0,1) (1,0,0) (0,1,0)
19 modality:=DWMRI
20 DWMRI_b-value:=1000
21 DWMRI_gradient_0000:= 0 0 0
22 DWMRI_gradient_0001:= 0.7071 0.7071 0
23 DWMRI_gradient_0002:= 0 0.7071 0.7071
24 DWMRI_gradient_0003:= 0.7071 0 0.7071
25 DWMRI_gradient_0004:= 0 0.7071 -0.7071
26 DWMRI_gradient_0005:= 0.7071 -0.7071 0
27 DWMRI_gradient_0006:=-0.7071 0 0.7071

```

Listing 2.1: Example NRRD file header

Diffusion Editor

This chapter describes the concept and implementation of the Diffusion Editor. At this point we explain parsing diffusion parameters, incorporating the modules for tensor estimation and tractography as well as the tensor glyph view. We present the user interface and describe the functionality of the Diffusion Editor in general, and by demonstrating application examples.

3.1. Concept and Implementation

We implement the Diffusion Editor within 3D-Slicer, which uses the open source framework Visualization ToolKit (VTK). It is a cross-platform image processing and visualization software, successfully used in medical context.

3D-Slicer has a module based architecture. Thereby each module provides specific features for image analysis. The Diffusion Editor itself is not an independent module. For now, we want to incorporate it as an additional part in the existing *Volumes* module of Slicer. Originally, the *Volumes* module loads and saves volume data and adjusts display parameters. Hence, the Diffusion Editor is enabled if the active volume is a DWI or DTI data set (disabled for any non-diffusion data).

For the implementation, the requested functions can be divided into the following:

- Procedure to make the diffusion data usefully available for editing.
- Incorporation of the display option *tensor glyphs*, and the modules *tensor estimation* and *fiducial seeding tractography* for exploring.

3.1.1. Parsing Parameters

Internally in Slicer, each object is represented by a MRML node with specific attributes according to its data type. In our case the diffusion data is stored as a `vtkMRMLDiffusionWeightedVolumeNode`, which has the attributes for the measurement frame, gradients and b-values.

The measurement frame is stored as a `vtkMatrix4x4`. Each entry is read out and displayed on the graphical user interface (GUI). Before storing we need to check the determinant of the measurement frame, it has to be 1. Figure 3.1 shows a flow chart for this process. It is important to save changes immediately back to the node. This way the *new* parameters are useable for any following operations and it relieves the user of doing it manually. But in turn, this requires an undo/redo mechanism. We describe this technique in section 3.1.5.

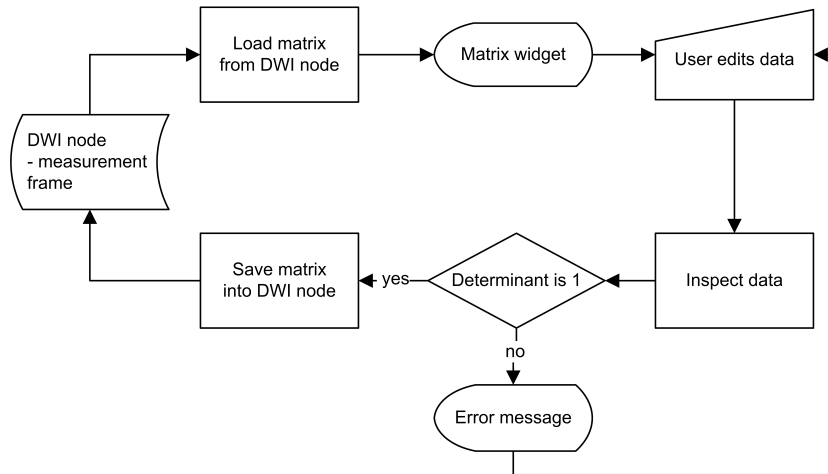


Figure 3.1.: Flow chart diagram demonstrating the measurement frame processing.

Gradients and b-values are stored as `vtkDoubleArray`. A single b-value is stored for every gradient because it is possible to use multiple b-values during a DWI acquisition. Figure 3.2 provides a flow chart demonstrating key steps in the pipeline of gradient and b-value processing. Multiple b-values result in scaling gradients by normalizing them to $\sqrt{\frac{b_{val}}{b_{val,max}}}$, where b_{val} is the actual b-value and $b_{val,max}$ is the maximal b-value all gradients are normalized to. This scaling is part of converting DICOM data to NRRD header, in which normalized gradients have to be stored. However, within the `vtkMRMLDiffusionWeightedVolumeNode` gradients are stored with its according b-value. To display a single $b_{val,max}$ we need to parse the stored b_{val_i} by solving the following equation:

$$b_{val_{max}} = \frac{b_{val_i} \cdot (|g_{max}| - |g_{min}|)}{|g_i|} = \frac{b_{val_i} \cdot range(|g|)}{|g_i|} \quad (3.1)$$

where $b_{val_{max}}$ is the single b-value for all gradients, b_{val_i} is the b-value to gradient g_i , $|g_i|$ is the normalized gradient and $range(|g|)$ is the difference between the maximal normalized gradient $|g_{max}|$ and the minimal normalized gradient $|g_{min}|$.

Additionally, to store changes back to the node, we must invert the previous calculation and parse $b_{val_{max}}$ to every single b_{val_i} :

$$b_{val_i} = b_{val_{max}} \cdot \frac{|g_i|}{range(|g|)} \quad (3.2)$$

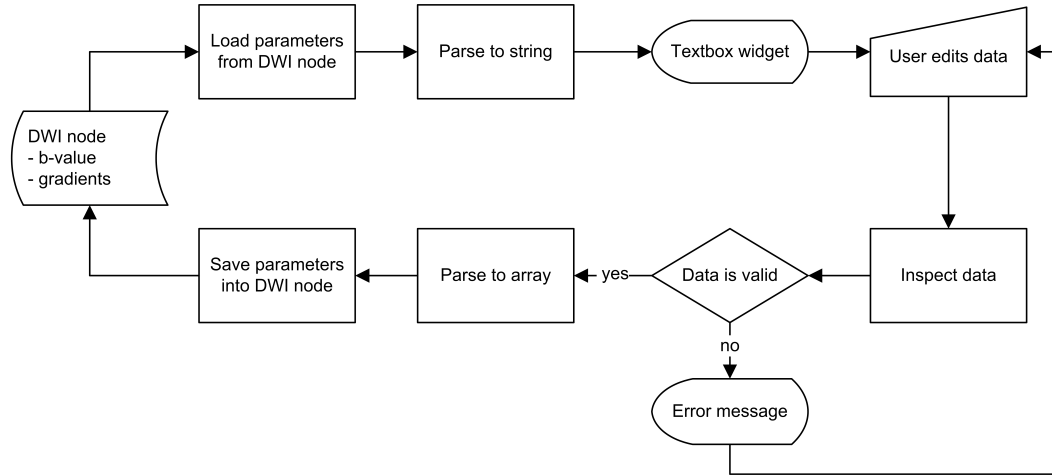


Figure 3.2.: Flow chart diagram demonstrating the pipeline of gradient and b-value processing.

(See also code extracts in sections A.1 and A.2)

3.1.2. Tensor Estimation Module

The tensor estimation module of 3D-Slicer is implemented as a command line module (CLM). This has the advantage to process a large number of data sets via batch mode. However, Slicer provides also an interactive GUI for CLMs, which are generated automatically. We describe below how the tensor estimation CLM is called within the code of the Diffusion Editor.

In the following we speak of a tensor *node* (`vtkMRMLDiffusionTensorVolumeNode`), Slicer’s internal representation of a tensor volume.

First, we create a CLM and set its name. It is important to use the actual name of the CLM used in the 3D-Slicer environment.

```
vtkMRMLCommandLineModuleNode *tensorCLM = vtkMRMLCommandLineModuleNode::
    SafeDownCast(this->MRMLScene->CreateNodeByClass("vtkMRMLCommandLineModuleNode
    "));
tensorCLM->SetModuleDescription("Diffusion Tensor Estimation");
```

The required input parameters must be set before the CLM can be executed. The module provides the linear-least-squares, non-linear-least-squares and weighted-least-squares method for the estimation. But the Diffusion Editor uses the linear-least-squares method by default, as it is the fastest and most robust. For the processing estimation the default threshold is 0.5, and the options for island removing and threshold mask applying are disabled.

```
tensorCLM->SetParameterAsString("estimationMethod", "Least Squares");
tensorCLM->SetParameterAsDouble("otsuOmegaThreshold",0.5);
tensorCLM->SetParameterAsBool("removeIslands", 0);
tensorCLM->SetParameterAsBool("applyMask", 0);
tensorCLM->SetParameterAsString("inputVolume", this->ActiveNode->GetID());
```

The output nodes are created and added to the current Slicer scene, which holds all currently existing nodes. For the tensor node the operations are:

```
vtkMRMLDiffusionTensorVolumeNode *tensorNodeCLM =
    vtkMRMLDiffusionTensorVolumeNode::SafeDownCast(this->MRMLScene->
    CreateNodeByClass("vtkMRMLDiffusionTensorVolumeNode"));
tensorNodeCLM->SetName("GradientEditor_TensorNode");
tensorNodeCLM->SetScene(this->GetMRMLScene());
this->MRMLScene->AddNode(tensorNodeCLM);
```

We create an additional `baselineNode` and `maskNode` like the `tensorNode` from above and set the outputs.

```
tensorCLM->SetParameterAsString("outputTensor", tensorNodeCLM->GetID());
tensorCLM->SetParameterAsString("outputBaseline", baselineNodeCLM->GetID());
tensorCLM->SetParameterAsString("thresholdMask", maskNodeCLM->GetID());
```

Finally, the CLM can be executed in the background. We invoke the CLM by using the CLM GUI's logic to provide progress feedback on the overall Slicer status bar.

```
vtkCommandLineModuleGUI *moduleGUI = vtkCommandLineModuleGUI::SafeDownCast(this->
    Application->GetModuleGUIByName("Diffusion Tensor Estimation"));
moduleGUI->Enter();
moduleGUI->SetCommandLineModuleNode(tensorCLM);
```



```

moduleGUI->GetLogic()->SetCommandLineModuleNode(tensorCLM);
moduleGUI->GetLogic()->Apply(tensorCLM);

```

3.1.3. Tractography Module

For the Diffusion Editor we determine to use fiducial seeding tractography, described in section 2.3.2. The *Tractography Fiducial Seeding* module of Slicer is an interactive module. It can be invoked by calling `CreateTracts()` in the logic. The module takes as input a tensor node, computed from the previous tensor estimation, and a fiducial list, containing all seeding points to compute a set of tracts. These tracts are stored in an output fiber node. To get process feedback for the status bar, we use the GUI's logic like in the tensor estimation,

```

vtkSlicerTractographyFiducialSeedingGUI *moduleGUI =
    vtkSlicerTractographyFiducialSeedingGUI::SafeDownCast(
        this->Application->GetModuleGUIByName("FiducialSeeding"));
moduleGUI->Enter();
moduleGUI->SetVolumeSelector(this->TensorNode);
moduleGUI->SetFiducialSelector(fiducialListNode);
moduleGUI->SetOutFiberSelector(this->FiberNode);
moduleGUI->CreateTracts();

```

3.1.4. Tensor Glyphs

Tensor glyphs are a display option and visualized for the active volume. Every tensor node has an attribute for glyph visibility. It can be set accordingly for each plane (axial, sagittal, coronal). Then the glyphs are rendered and displayed.

```

glyphDisplayNodes = this->TensorNode->GetSliceGlyphDisplayNodes();
//glyphs on for plane i (1=axial, 2=sagittal, 3=coronal)
glyphDisplayNodes[i]->SetVisibility(1);

```

3.1.5. Undo/Redo Mechanism

The undo/redo mechanism of the Diffusion Editor provides the opportunity to go back to a previous state of the active `vtkMRMLDiffusionWeightedVolumeNode`. As each change of the measurement frame or gradients is stored at once, they are available in following operations, e.g. tensor estimation.

Before a change is applied to the node, the active node is copied and saved in a stack. We differ between size and position of the stack, where size is the actual number of all nodes in the stack and position points to the active node.

When undo is called, the current state of the node is saved into the stack. From this position, the active node is set to the previous node and the stack position is decremented. Undo can be repeated until the stack position is 1. That means the first saved node in the stack is active. Once undo is executed, redo is enabled.

When redo is called, it reverses the last performed undo action. Based on the stack position, the active node is set to the next node. Redo can be repeated until the active node is the last saved node.

That way, every saved node in the stack can be selected as the active node. However, if a new node is saved, and the stack position is not equal to the stack size, all following nodes are no longer reachable and deleted.

Additionally to undo/redo, restore can be called. It sets the active node to the original node and the stack is cleared. Figure 3.3 shows an illustration.

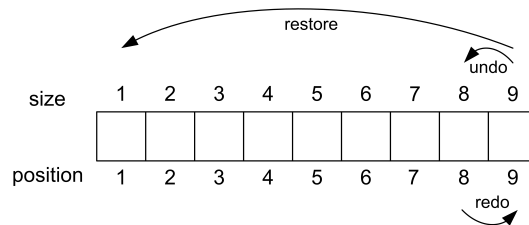


Figure 3.3.: Schematic description of the undo/redo and restore mechanism. Undo: Position is decremented, the previous node is active. Redo: Position is incremented, the next node is active. Restore: The first node is active and the stack is cleared.

3.1.6. User Interface

An additional task is to build a clear and simple interface. 3D-Slicer uses KWWidgets [3], a cross-platform and open-license GUI Toolkit, for its interface structure.

Figure 3.4 shows the interface of the Diffusion Editor, which can be divided into three main parts:

- a) Measurement Frame Panel: Provides a display for the measurement frame. Operations for rotating, swapping and inverting the measurement frame (according to the selected columns) and resetting the matrix to identity are implemented. Additionally, manual editing can be performed.

- b) Gradients Panel: Provides a display for the gradients. Furthermore, it contains the functionality for the import of gradients and manual editing.
- c) Testing Panel: Contains all functionality in terms of tensor estimation and visualization. These are setting glyph visibility and glyph spacing as well as fiducial seeding tractography.

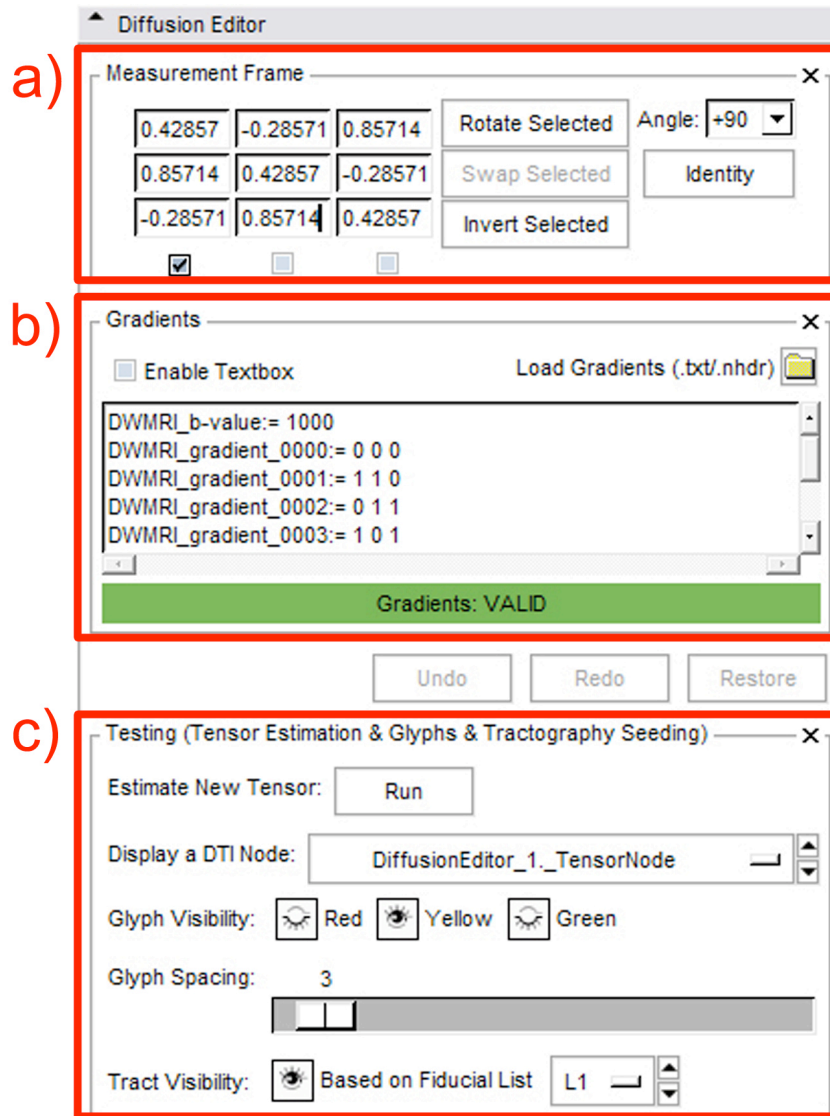


Figure 3.4.: User Interface of the Diffusion Editor. It is split into the editing panels for measurement frame (a) and gradients (b), as well as the testing panel (c) for tensor estimation and visualization.

We decide to put the control buttons above the testing panel, because undo/redo and restore effect only changes of gradients and measurement frame.

3.2. Functionality

In the following, we describe the functionality of the Diffusion Editor to edit the measurement frame and gradients, compute a tensor and investigate the results by tensor glyphs and fiducial seeding tractography.

As precondition we determine a loaded DWI volume.

3.2.1. Editing Measurement Frame

We assume the given measurement frame is not correct and we know the right one. There are several ways to change the matrix in the measurement frame panel (see figure 3.5):

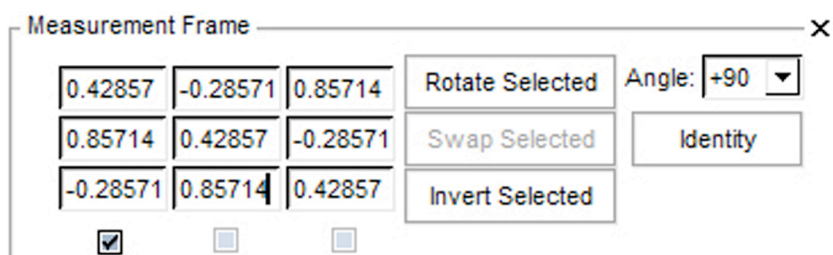


Figure 3.5.: Measurement frame panel. Options are: Manual editing, setting the matrix to identity and rotating, swaping, inverting selected axes.

Manual Editing

To set a correct value, we enter it into the field for the according element of the measurement frame. The determinant of the matrix has to be 1. This is automatically checked by the editor and if necessary, an error message is generated.

Identity

The measurement frame is set to the identity matrix. This is helpful to get to a clear state after too much editing.

The following operations are only enabled when the right number of axes/columns are selected (x-axis = column 1, y-axis = column 2, z-axis = column 3):

Rotate Selected

In this case, one column has to be selected, around which the measurement frame is rotated. Before that, we have to choose an angle from the given set of values or set it manually.

Swap Selected

With this operation, the values of two selected columns are exchanged.

Invert Selected

Basically, inverting a column means flipping an axis. The x-axis flips the volume around left-right, the y-axis around anterior-posterior and the z-axis around superior-inferior. For example, if the measurement frame is given in left-posterior-superior (LPS) and right-anterior-superior (RAS) is needed, we need to invert the x- and y-axis.

3.2.2. Editing Gradients

In the gradients panel (see figure 3.6) there are two alternatives available to edit gradients of DWIs:

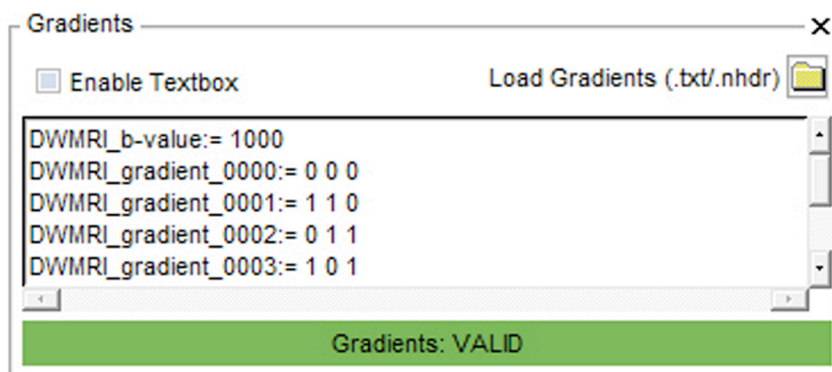


Figure 3.6.: Gradients panel. Options are: Manual editing and loading gradients from a given file.

Manual Editing

The checkbox *Enable Textbox* must be activated. Afterwards, we can enter gradient values or copy & paste gradients into the textbox. They are checked in realtime and the result is displayed below in the status bar; it is green for *Gradients: VALID* or red for *Gradients: INVALID*.

Load Gradients

To load gradients we choose a text or NRRD header file (.txt or .nhdr), in which the correct gradients are saved. They can be stored as single values and don't need to be provided in a specific format. As a matter of course, they are also automatically verified.

3.2.3. Exploring

If we have no tensor yet, we calculate a new tensor by running the *tensor estimation* in the testing pane (see figure 3.7). The DTI selector (*Display a DTI Node*) contains all available tensor nodes from the current scene. To enable the visualization options, one of these tensor nodes has to be selected.

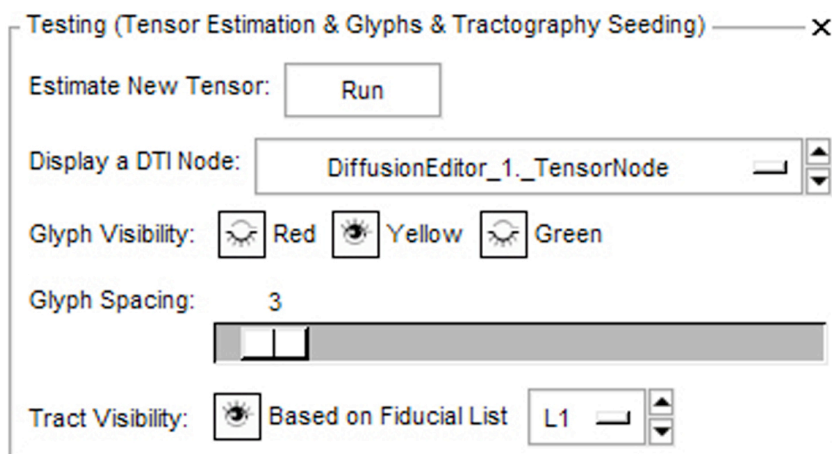


Figure 3.7.: Testing panel. Options are: Estimating a new tensor, activating glyphs and tracts view for a selected tensor volume.

Display a DTI Node

This selector activates a tensor volume for exploring. All following settings regarding glyphs and tracts are applied directly to this currently selected node. For example, we assume a fiducial list is selected, the tracts view is active and already displaying tracts for the current tensor volume. If the DTI selector is changed to another already computed tensor, tracts are automatically updated. This offers the possibility of an immediate comparison of several tensors.

Glyph Visibility

Activating the red, yellow or green visibility button displays tensor glyphs. Here, the colors represent the colored labels of the three 2D slice viewers, in which the glyphs are displayed accordingly. By default, red stands for the axial slice, yellow for the sagittal slice and green for the coronal slice. But other orientations can also be selected.

Glyph Spacing

This controller regulates the density of glyphs.

Tract Visibility

First, fiducials have to be set. This creates a new fiducial list automatically, which has to be selected in order to start the tractography. The visibility button executes the calculation of tracts and displays them afterwards.

3.3. Application Examples

In this section, we provide a set of application examples to demonstrate the use of the Diffusion Editor and visually illustrate the different functionalities, described in the previous section 3.2. The results of the transformations applied to the measurement frame and gradients are inspected and analyzed. These show the importance of correct diffusion header data.

For all examples we use a LPS brain scan with 144x144 pixels, 1.6667mm x 1.6667mm pixel size, 1.7mm thickness and 85 layers in 59 gradient directions by a b-value of 900 (01053-dwi-filt-Ed.raw.gz).

3.3.1. Tensor Rotation

Figure 3.8 illustrates a sequential rotation of the measurement frame by increments of 3 degrees. A tensor is estimated for every condition. To compare the results and illustrate the qualitative impact, which is minor at first, and then becomes obvious at extreme conditions, we successively select the tensors with the DTI selector. Each volume shows the same representative set of tracts and demonstrate the continuing change.

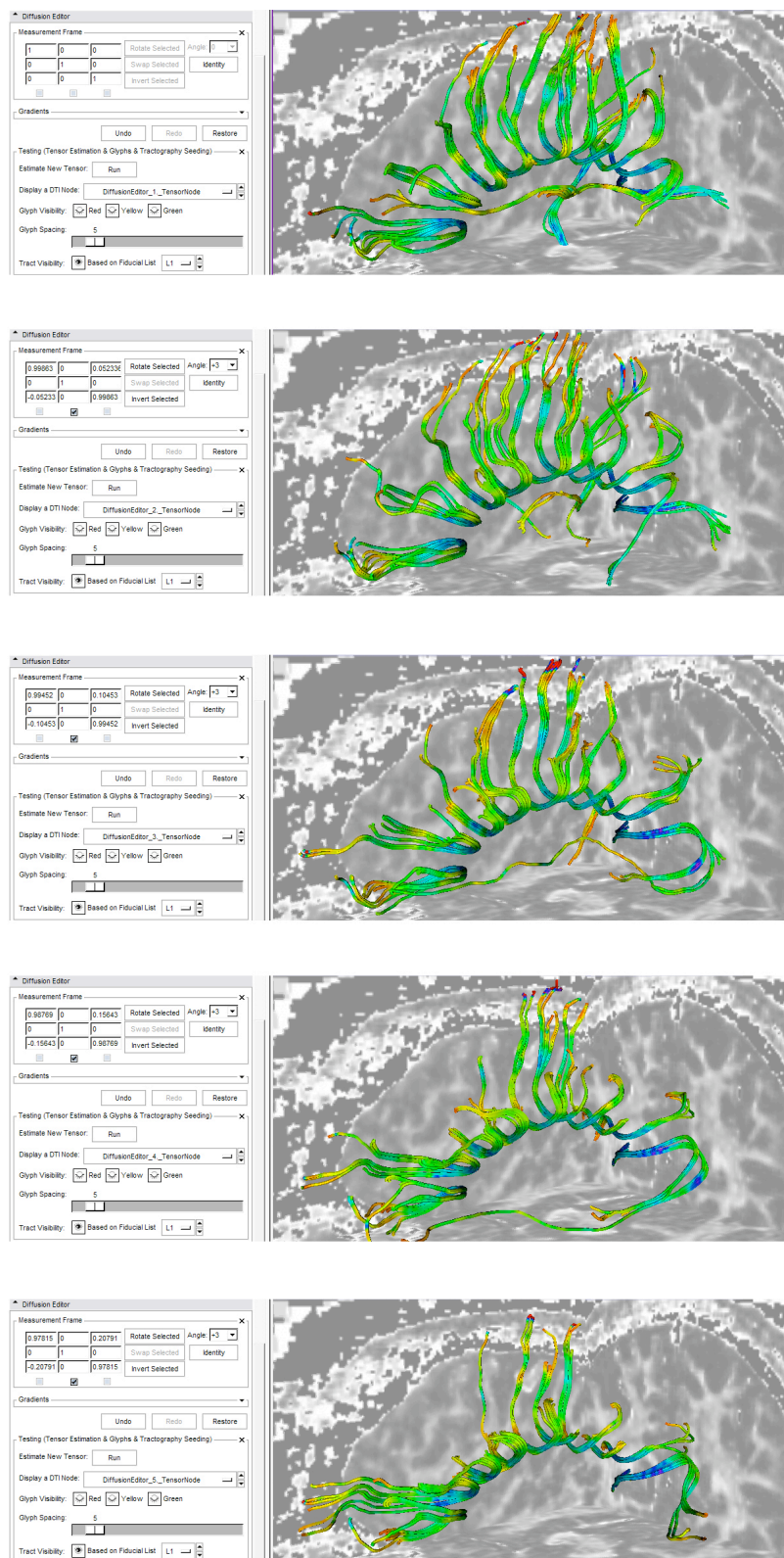
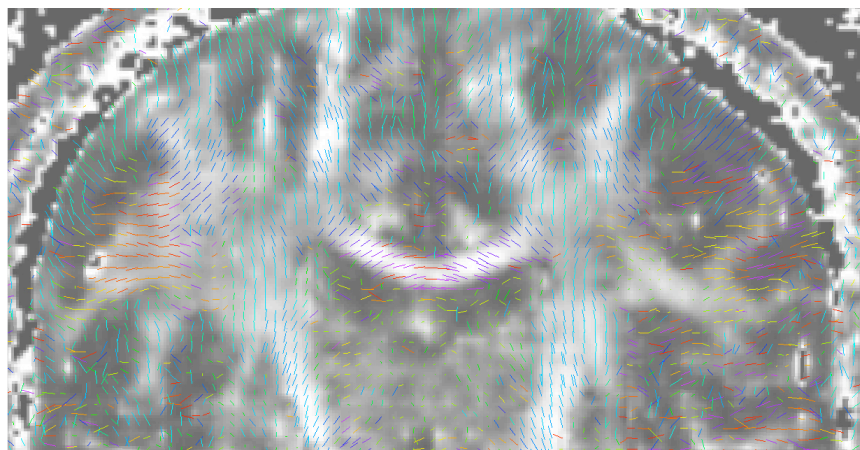


Figure 3.8.: Tractography on the same tensor image. Starting at the top, each image shows the impact of a further 3 degrees rotation of the measurement frame.

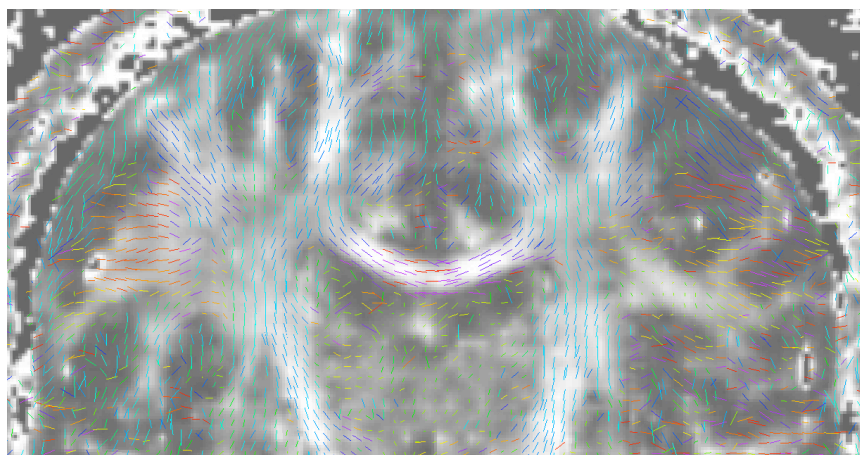
3.3.2. Tensor Flipping

If tractography does not show useful results, it might be because the acquisition is up side down. We can explore this by flipping (inverting) the measurement frame along the x-, y- or z-axis and estimate a new flipped tensor. By selecting one of the tensors in the DTI selector, we see immediate comparison.

Figure 3.9 illustrates a flip in x- and y-axis from RAS to LPS coordinates. In this case, the ROI of the corpus callosum is a good indication for inspection of the right geometric orientation of glyphs. These are obviously incorrect in RAS, but after flipping to LPS the glyphs are oriented in the right direction.



(a) Measurement frame in RAS

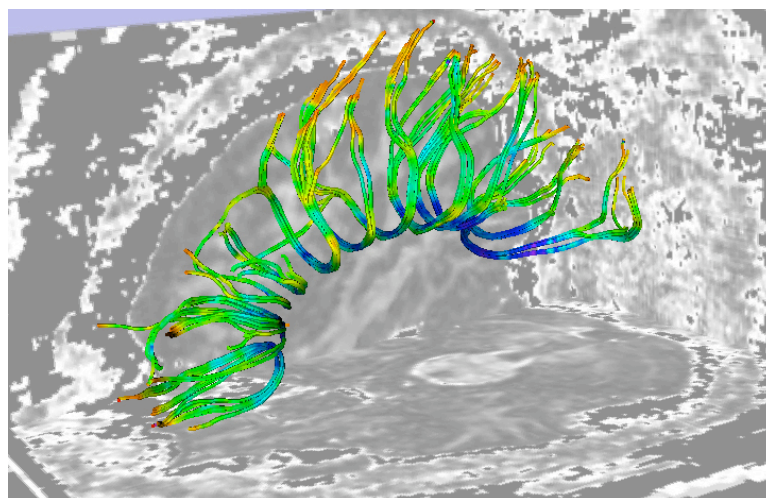


(b) Measurement frame in LPS

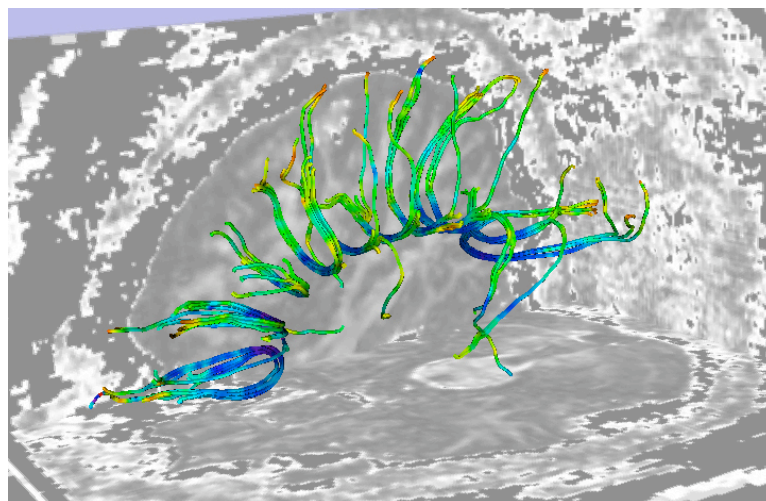
Figure 3.9.: Glyphs on the same DTI show the impact of flipping the measurement frame from RAS and LPS. Apparently, in figure a) the direction of the glyphs in the area of the corpus callosum are not correct. However, in figure b) the measurement frame is flipped and the glyphs have the right geometrical orientation.

3.3.3. Gradient Editing

We described the issue of rounded gradients in section 2.4.1. At this point, we examine the quantitative impact by showing tractography in the ROI around the corpus callosum. The gradients are rounded to the next integer, saved in a text file and loaded into the editor. We estimate a new tensor and compare the results by selecting a tensor volume with the DTI selector.



(a) Gradients original



(b) Gradient rounded

Figure 3.10.: Tractography on the same DTI show the impact of rounding gradients. In figure a) tracts starting in the area of the corpus callosum show a correct geometric orientation. Obviously, rounded gradients in figure b) have an impact on the seeding of tracts.

Certainly, the more decimal points you have after rounding the less impact is observable. As a qualitative measure the fractional anisotropy would give a significant evaluation.

3.4. Tutorial Videos

Additionally, to support the whole user guidance and explain the use and functionality, tutorial videos are available. They demonstrate a typical workflow and are based on each other. In the following, the tutorials are described in detail, starting with loading a DICOM data set, editing parameters, estimating tensors and finally exploring these parameters with tensor glyphs and tractography. (The video material as well as the example data sets are available on the enclosed CD-ROM and on the Diffusion Editor documentation wiki page [1])

Load DICOM

We start by importing a DICOM data set within Slicer. It has to be converted into NRRD (see section 2.4) so it can be loaded as a DWI in the *Volumes* module. The *DICOM to NRRD* module undertakes this task.

Edit Gradients

The gradients from the previously loaded DICOM have not been stored correctly in the header. We demonstrate changing the gradients manually and loading the correct gradients from an external text file (this file is also included in the example data set). Afterwards, to save the new gradients to file, the overall save option of 3D-Slicer is used.

Edit Measurement frame

The measurement frame from the DICOM data is also incorrect, but we know the used setting from the acquisition. To adjust the matrix we swap and invert the necessary axes. Like the gradients, we save the result to file.

Testing

To confirm the correctness of the parameters, we estimate a tensor and show glyphs on the volume. Furthermore, a set of fiducial points is added and tracts are displayed for verification. Secondly, we rotate the measurement frame by 90° degrees and run a new tensor estimation. We activate the visibility for tracts and glyphs on the sagittal slice. Switching between both tensors using the DTI selector provides the possibility to see and explore the impact of that change.

Discussion

4.1. Conclusion

In this thesis we pointed out the issues that currently exist for diffusion weighted image processing and its sub-techniques. We therefore presented an Diffusion Editor as a solution, with the purpose to improve the overall workflow.

We explained the concept and development within the 3D-Slicer environment using C++ and special frameworks, such as the Visualization ToolKit (VTK) and KWWidgets. The different functionalities and how to use the module are presented step by step. Moreover, we demonstrate the impact of wrong acquisition parameters on later calculations and visually illustrate the qualitatively analysis of results.

The advantages of the Diffusion Editor are apparent and confirmed by the feedback of frequent Slicer users. It is very useful to inspect and edit diffusion weighted and tensor data. It optimizes the editing process and accelerates the workflow, particularly to the former way of manually adjusting in the header data. During the use, ideas for further features emerged, which are discussed in the following section.

Tensor estimation, tensor glyphs and fiducial seeding tractography are the additional functions to directly explore the results of the previous editing and thereby check the correctness. They are an ideal extension to support the quality of acquisition parameters. However, in some cases the quantitative measures like fractional anisotropy should be considered.

4.2. Outlook

Further development on the Diffusion Editor is planned, as several users already requested additional features. Due to temporal reasons, they are not implemented completely and therefore not yet part of the editor.

We want to expand the gradients part for multiple b-values and add a table view to support this. For now, this is still an issue, as they can not be stored in DICOM or NRRD properly. The current realization is via normalization of gradients, as described in section 3.1.1. The feature should make adding and processing of multiple b-values easier for users.

At this stage, we start a complete new tensor estimation after changing the measurement frame. This re-estimation is very time consuming, particularly for large DWI volumes. But, both are 3x3 matrices, which permits to apply the transformation, we used on the measurement frame, on the tensor as well. The solution is a much faster way and can be explained as follows:

$$\begin{aligned}
 MFrame_{old} \cdot transform &= MFrame_{new} & (4.1) \\
 transform &= MFrame_{old}^{-1} \cdot MFrame_{new} \\
 Tensor_{old} \cdot transform &= Tensor_{new}
 \end{aligned}$$

where *transform* is the applied operation on the *old* measurement frame (*MFrame*) to get to the *new* one. This *transform* matrix is applied to the *old* tensor and results in the *new* tensor.

The necessary filter for this feature already exists and we already incorporated it into the Diffusion Editor. But first testing showed that there are still bugs in it. Hence, it is not yet part of the editor as further testing must be performed.

Saving gradients to the actively used node appeared to take a remarkable long time. First analyses showed that the saving-event triggers various events within 3D-Slicer, which in turn start other processes. Using the new *Eventbroker* in Slicer should reveal dependencies and help to improve the performance.

Bibliography

- [1] Diffusion Editor. Inspection and editing tool for diffusion weighted MRI., http://www.slicer.org/slicerWiki/index.php/Modules:Volumes:Diffusion_Editor-Documentation, accessed 30.07.2008.
- [2] Digital Imaging and Communications in Medicine (DICOM). Part 6: Data dictionary, http://medical.nema.org/dicom/2007/07_06pu.pdf, accessed 22.06.2008.
- [3] Kitware Inc. KWWidgets, <http://www.kwwidgets.org>, accessed 06.07.2008.
- [4] Kitware Inc. Visualization ToolKit (VTK), <http://www.vtk.org/>, accessed 22.06.2008.
- [5] National Alliance for Medical Imaging Computing (NA-MIC)., <http://www.na-mic.org/>, accessed 02.07.2008.
- [6] Nearly Raw Raster Data (NRRD). Image file format., <http://teem.sourceforge.net/nrrd/index.html>, accessed 02.07.2008.
- [7] Surgical planning laboratory (SPL). 3D-Slicer software environment for visualization and image analysis, www.slicer.org, accessed 22.06.2008.
- [8] G. W. Albers. Diffusion-weighted MRI for evaluation of acute stroke. *Neurology* 51(3 Suppl 3):S47–S49, Sep 1998.
- [9] R. Bammer, B. Acar, and M. E. Moseley. In vivo MR tractography using diffusion imaging. *European Journal of Radiology* 45(3):223–234, Mar 2003.
- [10] P. J. Basser and D. K. Jones. Diffusion-tensor MRI: theory, experimental design and data analysis - a technical review. *NMR in Biomedicine* 15(7-8):456–467, 2002.

-
- [11] P. J. Basser, J. Mattiello, and D. LeBihan. Estimation of the effective self-diffusion tensor from the NMR spin echo. *Journal of Magnetic Resonance B* 103(3):247–254, Mar 1994.
- [12] C. Beaulieu. The basis of anisotropic water diffusion in the nervous system - a technical review. *NMR in Biomedicine* 15(7-8):435–455, 2002.
- [13] D. L. Bihan, E. Breton, D. Lallemand, P. Grenier, E. Cabanis, and M. Laval-Jeantet. MR imaging of intravoxel incoherent motions: application to diffusion and perfusion in neurologic disorders. *Radiology* 161(2):401–407, Nov 1986.
- [14] D. L. Bihan, J. F. Mangin, C. Poupon, C. A. Clark, S. Pappata, N. Molko, and H. Chabriat. Diffusion tensor imaging: concepts and applications. *Journal of Magnetic Resonance Imaging* 13(4):534–546, Apr 2001.
- [15] D. B. Ennis, G. Kindlman, I. Rodriguez, P. A. Helm, and E. R. McVeigh. Visualization of tensor fields using superquadric glyphs. *Magnetic Resonance in Medicine* 53(1):169–176, Jan 2005.
- [16] O. Friman, G. Farneback, and C.-F. Westin. A Bayesian approach for stochastic white matter tractography. *IEEE Trans Med Imaging* 25(8):965–978, Aug 2006.
- [17] C. Goodlett, P. T. Fletcher, W. Lin, and G. Gerig. Quantification of measurement error in DTI: theoretical predictions and validation. *Medical Image Computing and Computer-Assisted Intervention MICCAI 2007* 10(Pt 1):10–17, 2007.
- [18] H. Hanyu, T. Asano, H. Sakurai, Y. Imon, T. Iwamoto, M. Takasaki, H. Shindo, and K. Abe. Diffusion-weighted and magnetization transfer imaging of the corpus callosum in Alzheimer’s disease. *Journal of the Neurological Sciences* 167(1):37–44, Aug 1999.
- [19] M. Kubicki, R. McCarley, C.-F. Westin, H.-J. Park, S. Maier, R. Kikinis, F. A. Jolesz, and M. E. Shenton. A review of diffusion tensor imaging studies in schizophrenia. *Journal of Psychiatric Research* 41(1-2):15–30, 2007.
- [20] C. Lai, S. Z. Zhang, H. M. Liu, Y. B. Zhou, Y. Y. Zhang, Q. W. Zhang, and G. C. Han. White matter tractography by diffusion tensor imaging plays an important role in prognosis estimation of acute lacunar infarctions. *British Journal of Radiology* 80(958):782–789, Oct 2007.
- [21] P. C. Lauterbur. Image formation by induced local interactions: Examples employing nuclear magnetic resonance. *Nature* 242(244):190–191, Jul 1973.

-
- [22] M. Lazar and A. L. Alexander. Bootstrap white matter tractography (BOOT-TRAC). *NeuroImage* 24(2):524–532, Jan 2005.
- [23] M. Maeda, A. Matsumine, H. Kato, K. Kusuzaki, S. E. Maier, A. Uchida, and K. Takeda. Soft-tissue tumors evaluated by line-scan diffusion-weighted imaging: influence of myxoid matrix on the apparent diffusion coefficient. *Journal of Magnetic Resonance Imaging* 25(6):1199–1204, Jun 2007.
- [24] P. Mansfield and P. K. Grannell. NMR 'diffraction' in solids? *Journal of Physics* C6:L422–L426, 1973.
- [25] S. Mori and P. B. Barker. Diffusion magnetic resonance imaging: its principle and applications. *The Anatomical Record* 257(3):102–109, Jun 1999.
- [26] S. Mori and P. C. M. van Zijl. Fiber tracking: principles and strategies - a technical review. *NMR in Biomedicine* 15(7-8):468–480, 2002.
- [27] M. Niethammer, R. S. J. Estepar, S. Bouix, M. Shenton, and C.-F. Westin. On diffusion tensor estimation. *Conf Proc IEEE Eng Med Biol Soc* Suppl:6707–6710, 2006.
- [28] T. P. L. Roberts and E. S. Schwartz. Principles and implementation of diffusion-weighted and diffusion tensor imaging. *Pediatric Radiology* 37(8):739–748, Aug 2007.
- [29] P. W. Schaefer, W. A. Copen, M. H. Lev, and R. G. Gonzalez. Diffusion-weighted imaging in acute stroke. *Magnetic Resonance Imaging Clinics of North America* 14(2):141–168, May 2006.
- [30] E. O. Stejskal and J. E. Tanner. Spin diffusion measurements: Spin echoes in the presence of a time-dependent field gradient. *Journal of Chemical Physics* 42:288–292, 1965.
- [31] E. V. Sullivan and A. Pfefferbaum. Diffusion tensor imaging in normal aging and neuropsychiatric disorders. *European Journal of Radiology* 45(3):244–255, Mar 2003.
- [32] C.-F. Westin, S. E. Maier, H. Mamata, A. Nabavi, F. A. Jolesz, and R. Kikinis. Processing and visualization for diffusion tensor MRI. *Medical Image Analysis* 6(2):93–108, Jun 2002.
- [33] Y. Wu, P. Storey, B. A. Cohen, L. G. Epstein, R. R. Edelman, and A. B. Ragin. Diffusion alterations in corpus callosum of patients with HIV. *American Journal of Neuroradiology* 27(3):656–660, Mar 2006.



Code Extracts

A.1. ParseGradientsBvalueToArray

```
int vtkSlicerDiffusionEditorLogic::ParseGradientsBvaluesToArray(const char *
    oldGradients, unsigned int numberOfGradients, vtkDoubleArray *newBValues,
    vtkDoubleArray *newGradients)
{
    if (oldGradients == NULL)
    {
        vtkErrorMacro(<< this->GetClassName() << ": oldGradients is NULL");
        return 0;
    }

    // read in current gradients
    std::stringstream grad;
    grad << oldGradients;

    // save all gradient values in a vector
    std::vector<double> vec;
    while(!grad.eof())
    {
        std::string dummy = "";
        double newValue = -1;
        grad >> dummy;
        // check if valid value
        if(StringToDouble(dummy, newValue)) vec.push_back(newValue);
        else

```

```

{
    std::string::size_type pos = dummy.find_first_of("=",0);
    std::string dummy2 = dummy.substr(pos+1);
    if(StringToDouble(dummy2, newValue)) vec.push_back(newValue);
}
}

// exit if not the necessary number of values are input
if(vec.size() != numberOfGradients*3+1)
{
    vtkWarningMacro("given values "<<vec.size()<< ",needed "<<numberOfGradients*3+1);
    return 0;
}

// set number of gradients to new arrays
newGradients->SetNumberOfComponents(3);
newGradients->SetNumberOfTuples(numberOfGradients);
newBValues->SetNumberOfTuples(numberOfGradients);

// create for bValue calculation
vtkDoubleArray *gradientNormalized = vtkDoubleArray::New();
double gradientRange[2];

// set gradients and compute gradientNormalized values for bValue calculation
for(unsigned int j = 1; j < vec.size(); j=j+3)
{
    for(unsigned int i=j; i<j+3;i++)
    {
        newGradients->SetValue(i-1,vec[i]);
    }
    gradientNormalized->InsertNextValue(sqrt(vec[j]*vec[j]+vec[j+1]*vec[j+1]+vec[j
        +2]*vec[j+2]));
}

// get range for bValue calculation
gradientNormalized->GetRange(gradientRange);

// compute bValues and set them
for (unsigned int i=0; i<numberOfGradients;i++)
{
    newBValues->SetValue(i,vec[0]*gradientNormalized->GetValue(i)/gradientRange[1]);
}

gradientNormalized->Delete();
return 1;
}

```

Listing A.1: ParseGradientsBvalueToArray

A.2. ParseGradientsBvalueToString

```

std::string vtkSlicerDiffusionEditorLogic::ParseGradientsBvaluesToString(
    vtkDoubleArray *BValues, vtkDoubleArray *Gradients)
{
    std::stringstream output;

    // create for bValue calculation
    double gradientRange[2];
    vtkDoubleArray *gradientNormalized = vtkDoubleArray::New();
    double g[3]; // single gradient

    // compute norm of each gradient
    for(int i=0; i<Gradients->GetNumberOfTuples();i++)
    {
        Gradients->GetTuple(i,g);
        gradientNormalized->InsertNextValue(sqrt(g[0]*g[0]+g[1]*g[1]+g[2]*g[2]));
    }

    // get gradientRange of norm array
    gradientNormalized->GetRange(gradientRange);

    // compute bValue
    double bValue = -1;
    for(int i = 0; i< BValues->GetSize(); i++)
    {
        double numerator = BValues->GetValue(i)*gradientRange[1];
        double denominator = gradientNormalized->GetValue(i);
        if(!numerator == 0 && !denominator == 0)
        {
            bValue = numerator/denominator;
            break;
        }
    }

    // read in new bValue
    output << "DWMRI_b-value:= " << bValue << endl;

    // read in new gradients
    // (this->Gradients->GetSize() is not always correct.)
    for(int i=0; i < Gradients->GetNumberOfTuples()*3; i=i+3)
    {
        output << "DWMRI_gradient_" << setfill('0') << setw(4) << i/3 << " := " << " ";
        for(int j=i; j<i+3; j++)
        {
            output << Gradients->GetValue(j) << " ";
        }
    }
}

```

```
    output << "\n";  
  }  
  
  gradientNormalized->Delete();  
  return output.str(); // string with bValue and gradients  
}
```

Listing A.2: ParseGradientsBvalueToString

B

Content of the enclosed CD-ROM

Directory	Contents
<code>./DiffusionEditor/Source</code>	.cxx and .h source files
<code>./DiffusionEditor/Data</code>	Example data
<code>./Tutorials</code>	Four tutorial videos: Load DICOM, Edit Measurement Frame, Edit Gradients and Testing
<code>./Tutorials/Data</code>	Necessary example data for the tutorials
<code>./Thesis</code>	PDF version of the thesis in A4 and letter format

Declaration

I have written this thesis independently, solely based on the literature and other sources listed in the bibliography and properly cited in the text. This document has not and will not be submitted to any other institution apart from the University of Heidelberg / University Heilbronn as a requirement for being awarded a diploma in Medical Informatics (“Medizinische Informatik”).

Heilbronn, August 1st 2008

(Kerstin Kessel, student ID: 160082)