

MASTER

Interactive fiber tracking and visualization

Holthuisen, R.F.J.

Award date:
2002

[Link to publication](#)

Disclaimer

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

TECHNISCHE UNIVERSITEIT EINDHOVEN

Department of Mathematics and Computer Science

MASTER'S THESIS

Interactive Fiber Tracking
And Visualization

by

R.F.J. Holthuisen

Supervisor: dr. A.C. Telea
Advisors: prof. dr. ir. J. J. van Wijk
ir. A. van Muiswinkel

Eindhoven, August 2002

Table of Contents

1. Introduction	5
1.1. Assignment	6
1.2. Overview	8
2. Diffusion Imaging	9
2.1. Diffusion.....	9
2.1.1. Introduction to Anisotropy	11
2.2. Data Handling.....	12
2.2.1. Scanning Techniques.....	12
2.2.2. Data Loading	12
2.2.3. Internal Data Storage	12
2.3. Interpolation	14
3. Fiber Tracking	15
3.1. Fiber Tracking	15
3.2. Numerical Integration of a Differential Equation	17
3.2.1. Euler's Method.....	17
3.2.2. Runge-Kutta Method	18
3.2.3. Comparing Euler's Method and Runge-Kutta	18
3.2.4. Non Cubic Voxels	19
3.3. Fiber Tracking Termination	20
3.4. Anisotropy as fiber quality indicator.....	21
4. Seed Points and Regions of Interest	23
4.1. Seed Points (Single ROI).....	23
4.2. Improved Seed Point Algorithm.....	25
4.3. Multiple ROIs	26
5. Visualization and User Interaction.....	27
5.1. The 3D Model.....	27
5.2. ROIs.....	28
5.2.1. Single ROIs.....	28
5.2.2. Multiple ROIs.....	28
5.3. Color Mappings	29
5.3.1. Planar Anisotropy	30
5.3.2. Tensor Shape	31
5.4. Fibers	33
5.4.1. Single Fibers (Hyperstreamlines).....	33
5.4.2. Real Time Fiber.....	34
5.4.3. Many Fibers (Quick Fiber Visualization)	35
6. Implementation.....	36
6.1. Overview	36
6.2. Performance.....	36
7. Conclusions & Recommendations	37
7.1. Conclusions	37
7.2. Recommendations	38
8. References.....	39
Appendix A. Original Fiber Tracking Program.....	40
A.1. Data Loading	41
A.1.1. Data Simplification Phase	41
A.2. Region of Interest Selection Phase	42
A.2.1. 2D Area Selection.....	42

A.2.2	Single Point Selection.....	43
A.3.	Fiber Tracking Phase.....	44
A.3.1	The Ideal Case	44
A.3.2	Non-Ideal Slice Case	45
A.3.3	Non-Ideal Slice Orientation Case.....	45
A.3.4	Normal Case	45
A.4.	Fiber Visualization Phase.....	46
Appendix B. Diffusion Storage Format.....		47
Appendix C. Dealing with crossing Fibers		51

Definition of terms and abbreviations

<i>Term</i>	<i>Description</i>
ADC	Apparent Diffusion Coefficient, Diffusion strength
Anisotropy	The shape of diffusion
B-value	The strength of the bipolar gradient pulse, which controls the sensitivity of the measured data to diffusion
Diffusion	The movement of molecules in a liquid due to thermal motion
Fiber	A thread or structure of tissue
fps	Frames Per Second
IDL	Interactive Data Language, a programming language
MR	Magnetic Resonance
MRI	Magnetic Resonance Imaging.
Pride	Philips Research International Development Environment
RAD	Rapid Application Development
ROI	Region of Interest
Tensor	A symmetrical 3x3 matrix of diffusion data
TU/e	Technical University of Eindhoven
Voxel	A 3D Volume in 3D space, defined by a grid

1. Introduction

Magnetic Resonance Imaging (MRI) allows to the *in vivo* study of human anatomy. This technology can generate images with high contrast of various tissues and organs. Because MRI can generate these images without damaging the patient, this technique is often used in diagnostic studies of the head, spine, joints and cardio-vascular systems.

Magnetic Resonance utilizes the magnetic moment of protons [1] to generate images of an object. Various scanning techniques are available to generate various types of images. One particular imaging technique is diffusion imaging (See chapter 2).

A 2D diffusion weighted image consists of pixels and represents a slice of the subject with a certain thickness. Using the thickness, a stack of images can be taken to represent a 3D volume of the subject consisting of "3D pixels" or voxels.

In fibers in the human brain, diffusion of water (See paragraph 2.1) is strong in the direction of the fibers, but weak perpendicular to the fibers. Fiber tracking uses this property to reconstruct fibers from diffusion measurements. The reconstructed fibers give a unique insight in the human brain, which allow for various useful applications.

Currently the diffusion scanning resolution is limited. This implies that the only 'fibers' that can be tracked will be relatively large structures of tissue. In reality these large structures are composed of many smaller fibers. Since the measured diffusion in a voxel is the average diffusion over the entire voxel, it is possible to track these large structures, as the average diffusion of fiber bundles will still be oriented.

The use of fiber tracking to track large structures can be useful in various areas. Applications can be:

- A research tool to obtain better knowledge of the structure of the brain.
- A research tool to help map the growth of the fibers in the brain, as a subject increases in age
- Fiber analysis as part of a diagnosis.
- Fiber analysis of a patient as a pre-surgical orientation to minimize collateral damage done by surgery

Currently fiber tracking is mainly used for the brain. However, other organs such as the spine or kidneys may also benefit from fiber tracking technology.

Figure 1 shows that constructed fibers are similar to fibers as found in post-mortem studies, which illustrates the clinical relevance of fiber tracking.



Figure 1 A comparison of fiber tracking results (right) with the anatomical layout (left) of the human brain

1.1. Assignment

Philips Medical Systems

Philips Medical Systems (PMS) is one of the world leading manufacturers of medical systems. The aim of PMS is *to provide clinical excellence without compromise* to the customer.

The student was an intern at Philips Medical Systems, and as such his goals and requirements are determined to a large extent by PMS.

The aim of this particular project, carried out in the MR imaging group, was to develop an Fiber Tracking Tool that can be used for clinical applications.

To achieve this aim, these global steps are taken:

1. Investigate the technical possibilities of fiber tracking and diffusion imaging
2. Create global acceptance for the capabilities and applications of fiber tracking.
By creating a Fiber Tracking tool, and distributing it to Philips' Clinical Research sites, clinical scientists will be enabled to investigate possible clinical applications.
3. Clinically prove the capabilities and applications of fiber tracking. This is one of the requirements of the Food and Drug Administration (FDA) before medical software packages are allowed to be released for clinical use.
4. Implement a FDA approved fiber tracking tool on the MRI scanner.

Assignment

My assignment contains the following items:

- Investigate the technical possibilities of fiber tracking and diffusion imaging.
- Implement a Fiber Tracking tool

Related Work

Diffusion imaging is a relatively new MRI imaging technique. Thus a great deal of research is currently focussing of diffusing imaging. The proceeding ISMRM of 2001 contains over 180 papers about diffusion. Relevant articles to this thesis are in particular [5] and [8].

Fiber tracking is an even newer modality, with the first major papers, such as [9] written around 1999.

An initial Fiber tracking tool was provided by others to serve as a start. This tool is summarized in Appendix A. Improvements of the algorithms are suggested in [4].

Current Problems

Although tools such as the tool in Appendix A exist, there are problems with the usability of these tools. They usually work for a few types of datasets, are unacceptably slow, require other tools that may not even run on the same machine, are console based or not even interactive, etc. In order to get clinical scientists to use diffusion and fiber tracking tools, and thus possibly discover new clinical applications, there needs to be a single tool that supports a wide variety of datasets, is quick, is interactive, is user friendly and is robust. The Fiber Tracking tool should also incorporate already established diffusion visualization capabilities, so a single tool can provide all the desired functionality.

Requirements

Summarizing the requirements for the tool to be developed:

- Many/all dataset types must be supported
By supporting a wide variety of different scans, clinical scientist are not restricted in the type of scans or hardware they want to use. This will allow for an as wide possible clinical use of the program. Even scans other than neurological scans should be supported.
- Interactivity is vital
Since Fiber Tracking programs are new, there is no predefined series of user input that will always lead to the wanted results. The user must be encouraged to experiment with the Fiber Tracking program. This is done by high interactivity.
- Algorithm speed is vital
Interactivity is improved if the program responds quickly
- Robustness is important
Fiber tracking can produce anatomically incorrect results. The program should be as robust as possible and wherever possibly indicate the reliability of the found results.
- User friendliness is important
The program is meant for clinical scientists who do not have a vast amount of computer knowledge
- One tool must provide all functionality in one integrated solution

1.2. Overview

Figure 2 illustrates the fiber tracking process. This thesis is structured similarly to the phases of the fiber tracking process:

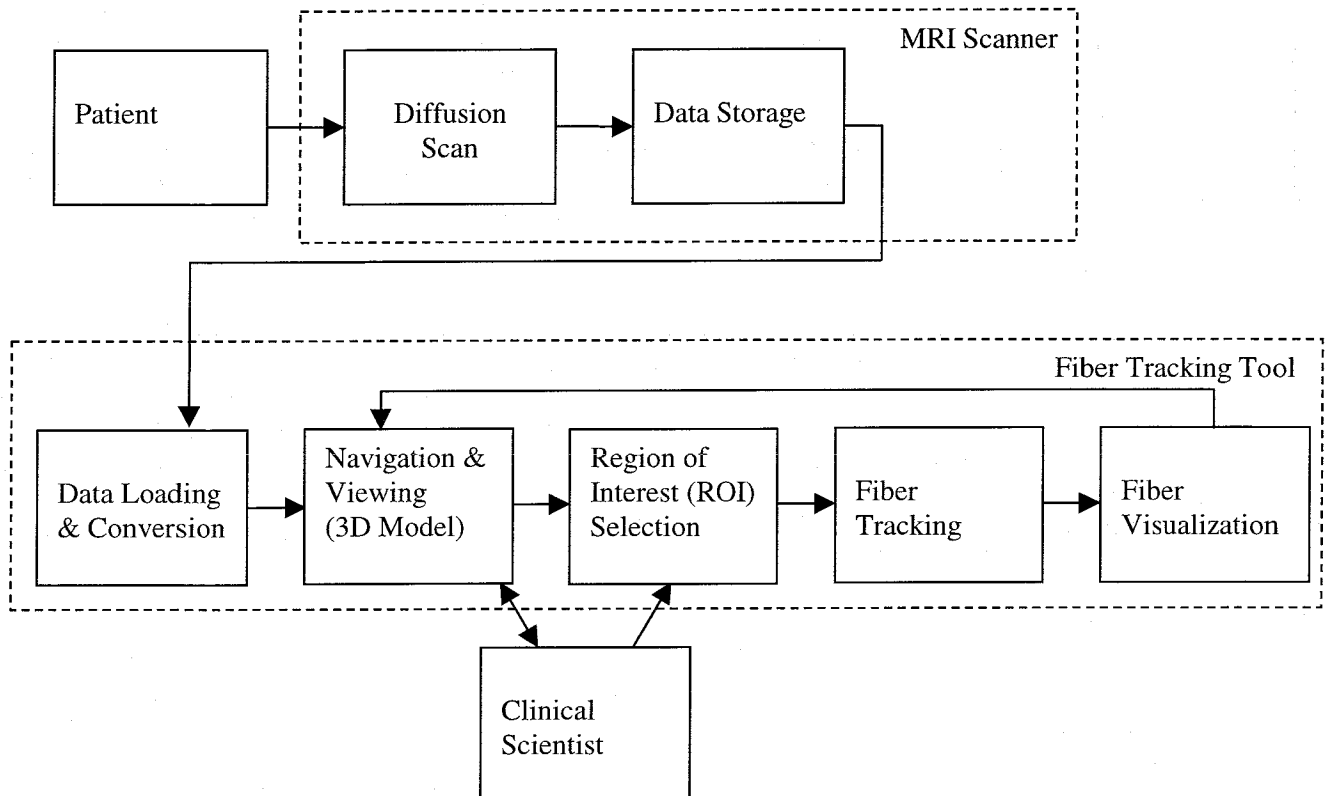


Figure 2 Overview of the Fiber Tracking Process

First a diffusion scan is made of a patient in a MRI Scanner. For more information, see paragraph 2.1.

This dataset is loaded by the Fiber Tracking tool. This can be done from the local hard disk or directly from the scanner via the Pride (Philips Research International Development Environment) interface. For more information, see paragraph 2.2.

If fibers are tracked in the entire dataset, too many fibers would be found, and the user will be unable to interpret the visualization. To prevent this from happening, the user must select a limited region of interest (ROI). In the fiber tracking phase, the fibers are calculated for the ROI with the selected algorithms. For more information, see paragraph 3.1.

A 3D model is available, where the user can view slices of data, and directly select a ROI in the 3D model. For more information, see paragraph 4.1.

In the fiber visualization phase, the 3D visualization of the fiber is constructed. It will be shown in the 3D model. For more information, see paragraph 5.4.

2. Diffusion Imaging

2.1. Diffusion

Diffusion is a fundamental property of liquids: The ability of molecules in liquid to move within it as a result of the Brownian motion. Diffusion can be shown by dropping a drop of ink in water. The ink will spread through the liquid as a result of the Brownian motion. Brownian motion results from the random thermal motion of molecules in a liquid.

Using a special MR technique one can obtain weighted diffusion data [2]. The measured diffusion data shows in each voxel of several mm^3 , on a statistical basis, the movement of water molecules in that area. Because molecules in a liquid travel more quickly when they are not obstructed by structures, this data gives a unique insight in the structure and geometrical organization of the scanned object.

By applying a magnetic pulse, the so-called bipolar gradient pulse, the measured data will be sensitive to diffusion. By varying the axis of the bipolar gradient pulse, diffusion can be measured in different directions.

Using both data where the bipolar gradient pulse has been applied and reference data measured without the bipolar gradient pulse, the Apparent Diffusion Coefficient (ADC) can be calculated. Figure 3 shows the measured data with and without the bipolar gradient pulse.

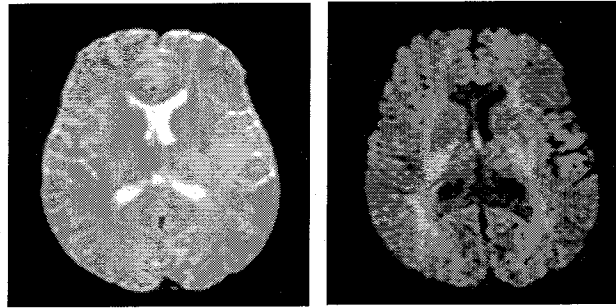


Figure 3 The image on the left show the measured data when no bipolar gradient pulse has been applied. The image on the right shows the measured data when a strong bipolar gradient pulse has been applied.

The ADC is defined as

$$ADC_{\underline{g}} = \frac{\text{Log}\left(\frac{DWI_{\underline{g}}}{DWI_{B0}}\right)}{-B_{value}} \quad (2.1)$$

Where \underline{g} is the vector describing the axis of the bipolar gradient pulse, and where the scalar $DWI_{\underline{g}}$ is the measured signal when the bipolar gradient pulse is applied, and the scalar DWI_{B0} is the measured signal when no bipolar gradient pulse is applied. B_{value} is the strength of the bipolar gradient pulse.

Diffusion in a voxel can be described by a symmetrical second order tensor, representable as a three-by-three real valued symmetric matrix

$$D = \begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{pmatrix} \quad (2.2)$$

An example diffusion measurement of a slice of the human brain is shown in Figure 4. The various pictures represent the tensor elements as described above.

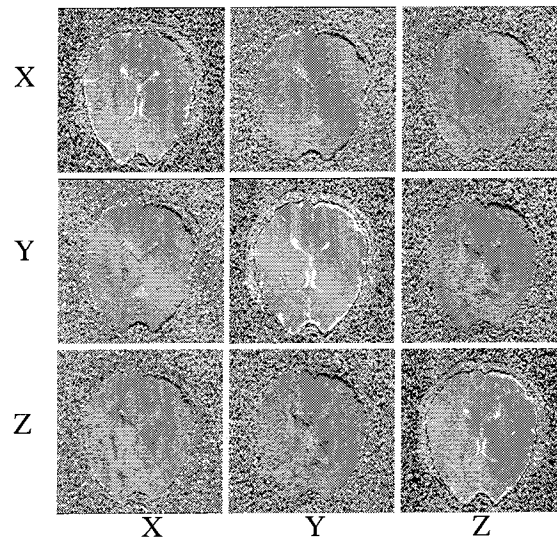


Figure 4 Matrix of images: Each image in the matrix represents an element of the diffusion matrix.

All real-valued three-by-three symmetric matrices have three real eigenvalues and corresponding eigenvectors [3]. These eigenvalues and eigenvectors can be represented as an ellipsoid describing the diffusion tensor. The ellipsoid's axes are in the direction of the eigenvectors (v_1 , v_2 and v_3), and are scaled according to the eigenvalues (λ_1 , λ_2 , λ_3). The eigenvectors and eigenvalues are sorted on the eigenvalue: $\lambda_1 \geq \lambda_2 \geq \lambda_3$. These eigenvectors are called the major eigenvector, the medium eigenvector and the minor eigenvector or primary eigenvector, secondary eigenvector and tertiary eigenvector.

Because diffusion is stronger in the direction of a fiber, and weaker orthogonal to this direction, the major eigenvector should point in the direction of the fiber, while the medium and minor eigenvectors describe diffusion in the directions orthogonal to the fiber. Figure 5 shows an ellipsoid representing a tensor and the eigenvectors scaled according to the eigenvalues of the tensor.

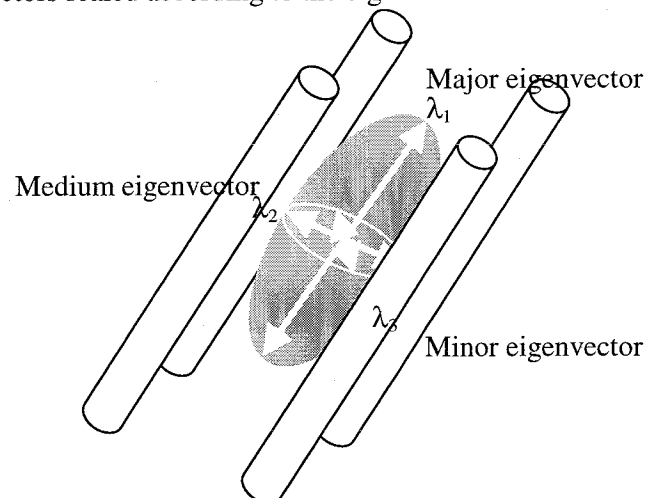


Figure 5 Eigenvectors and Eigenvalues give an intuitive representation of the diffusion tensor

At least six diffusion directions need to be measured to construct the diffusion tensor. In case the shape of diffusion in a voxel is complex and non-ellipsoidal, more than six measurements of different directions are required to properly represent the shape of the diffusion in this voxel. Although a tensor is an inadequate model for complex diffusion shapes, in the current tool it is still the model used for scans with more than six directions. This is done because all algorithms assume that eigenvalues and eigenvectors exist. The additional information that multiple directions provide is used to calculate a tensor that is less influenced by noise.

$N \geq 6$ measurements in different diffusion directions are performed on the scanner. From the measured data, the tensor is derived by

$$\underline{d} = (H^T H)^{-1} H^T \underline{S}$$

where $\underline{d} = (D_{xx}, D_{yy}, D_{zz}, D_{xy}, D_{xz}, D_{yz})^T$;

$$H = \begin{pmatrix} h_{g_1} \\ \vdots \\ h_{g_N} \end{pmatrix}; \tag{2.3}$$

$$\underline{h}_g = (g_x^2, g_y^2, g_z^2, 2g_x g_y, 2g_x g_z, 2g_y g_z);$$

$$\underline{S} = \begin{pmatrix} ADC_{g_1} \\ \vdots \\ ADC_{g_N} \end{pmatrix}$$

2.1.1. Introduction to Anisotropy

The term anisotropy describes the shape of diffusion in a voxel, as illustrated in Figure 6. Anisotropy is high when diffusion varies greatly as a function of direction. Anisotropy is low when diffusion is isomorphic or varies little.

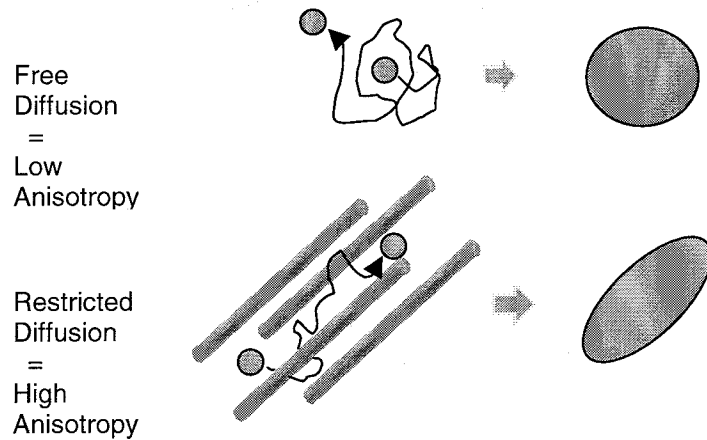


Figure 6 Diffusion is anisotropic in fiber bundles

Anisotropy is an important property of diffusion in MRI. Anisotropy images give important information to a clinical scientist. New imaging methods using anisotropy are discussed in paragraph 5.3.2 and 5.3.1.

Anisotropy is also used during fiber tracking as discussed in paragraph 3.4.

2.2. Data Handling

The Fiber Tracking Tool supports loading datasets in several formats.

2.2.1. Scanning Techniques

In order to calculate a tensor, at least six diffusion directions need to be measured. There are four types of scanning techniques on a Philips MRI scanner to scan diffusion over six directions:

- Default scanning technique (Non Overplus)
- Improved scanning technique (Overplus)
- A custom predefined scanning technique
- A custom manually entered scanning technique

Two other types of scanning techniques have the possibility to scan more than six directions. See Appendix B for more information.

2.2.2. Data Loading

A dataset that is imported consists of:

- DWI_{B_0} images
- DWI_g images for each diffusion direction g measured
- Parameters, such as pixel size, slice thickness, B_{value} and slice orientation

In order to convert the images to tensors, the user must manually enter several other parameters, since these are not available via the Pride interface. See Appendix B for more information.

2.2.3. Internal Data Storage

The patient coordinate system is the coordinate system that is relevant to the clinical scientist. Thus visualizations must be shown in the patient coordinate system. It is however inefficient to transform the data fully into the patient coordinate system, since this possibly requires a rotation other than 90° , 180° or 270° . The voxel cube will not be axis-aligned after such a rotation, leading to increased memory usage, as illustrated in Figure 7. Therefore the tensors are stored in the APAT coordinate system. The APAT coordinate system is the same as the patient coordinate system, except for a rotation T_{ang} . T_{ang} is always a rotation, so the 3D Model that visualizes the slices, ROIs and fibers is rotated according to this rotation, in order to present the data in the patient coordinate system to the user. Also when color mappings are calculated, T_{ang} rotates the tensors used before calculating the color, since color mappings must also be in the patient coordinate system.

Voxel positions are defined in the memory coordinate system defined in (3.5). See paragraph 3.2.4 for more information.

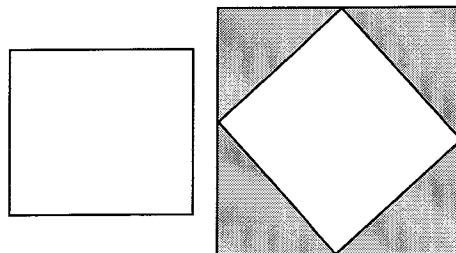


Figure 7 The left image show a 2D dataset that is axis aligned. The right image shows the same 2D dataset that is not axis aligned. The grey area represents wasted computer memory.

Implementation

To improve memory usage, only six measured signals are stored in memory. If there are more than six directions measured, these are reduced to six measurements by first calculating the tensor, and then reconstructing virtual measured signals using measurement directions from the Muthupallai scheme [10].

Apart from the measured signals the DWI_{B0} images are also stored in memory, since these are required for proper interpolation (See paragraph 2.3).

The user can load an optional anatomical dataset, which is used for visualization. Provided that this anatomical dataset has the same resolution as the diffusion dataset, the memory requirements can be calculated:

Resolution	Memory required
128x128x20	9.375 Mb
128x128x40	18.75 Mb
256x256x40	75 Mb

Table 1 Memory requirements for storing typical datasets

2.3. *Interpolation*

The fiber tracking algorithm benefits from being able to determine the tensor at any arbitrary position in the dataset, hence interpolation of the discrete data is required. Normally the scanner interpolates the dataset before it is exported or saved, but there are several problems with this approach:

- Only slices are interpolated, there is no interpolation between slices
- Interpolating before data loading will increase both disk space and memory usage significantly
- The tensor for an arbitrary position still needs to be calculated.

These problems can be overcome by not interpolating on the scanner, but by interpolating in the Fiber Tracking tool.

Using linear interpolation directly on the measured data corresponds with the physical reality. This is an algorithm that is already in use on the MRI scanner for some time. Interpolation of ADCs, tensors or eigenvectors would yield incorrect results. This is the reason why not the tensors are stored in memory, but the measured data instead.

Calculation of the interpolated eigenvalues and eigenvectors is done by:

1. Trilinear interpolation is performed on the measured data
2. The tensor is calculated from the measured data, see equation (2.3).
3. The eigenvectors and eigenvalues of the tensor are calculated

A consequence of this approach is that the eigenvectors and eigenvalues can not be calculated during the loading of the dataset, but on the fly, which will reduce the speed of the program.

Note that using smoothing (bilinear filtering) on an anisotropy image is incorrect, and should be avoided if two neighboring tensors differ greatly. This is due to the same reasons as why interpolation of the tensors or eigenvectors is incorrect.

3. Fiber Tracking

3.1. Fiber Tracking

A common technique to visualize tensor fields is to use an icon that represents the eigenvectors. The main problem of this technique is that the display will be too crowded if many icons are used, and hence it will be very hard for the viewer to get structural information of the scanned subject.

Only a subset of the diffusion data can be shown, in order to prevent overcrowding of the display. This can be done via fiber tracking. Fiber tracking is the process of constructing a visual representation of fibers out of tensor data. The fibers show insight in the major diffusion direction. Constructed fibers are similar to the actual fibers in the human brain, as is shown by Figure 1. The analogy between anatomical and constructed fibers makes fiber tracking a particular interesting technique for clinical scientists.

Fiber tracking initiates from a seed point that may be placed by the user. By tracking the major eigenvectors, the fibers passing through the region of interest can be found. These fibers provide better insight in the anisotropic structures of the object and can be used to provide this information to the user.

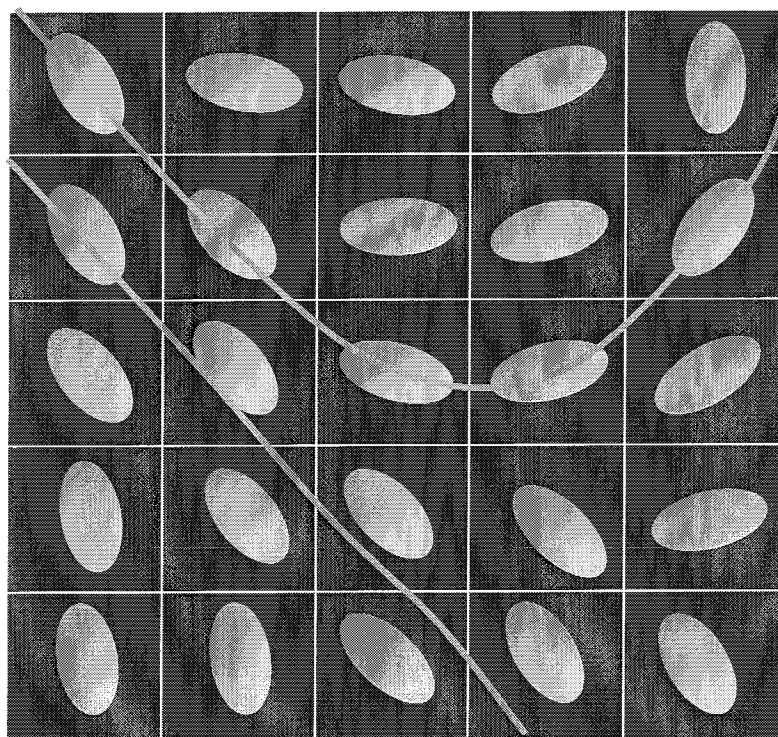


Figure 8 Fiber tracking tracks the major eigenvectors in voxels

Since interpolation can be used to calculate the tensor at arbitrary positions, fiber tracking can be implemented by numerically integrating a differential equation.

If V is a vector field of major eigenvectors of a tensor field D

$$V = \{v(p) \mid Dv(p) = \lambda_1 v(p)\}$$

$$\text{then } \underline{p}(t) = \underline{p}(0) + \int_0^t v(\underline{p}(t)) dt \quad (3.1)$$

defines the fiber in one direction over a distance given by the parameter t , provided that V has been compensated for 'flipped vectors': because vectors are symmetrical, they may be inverted in the direction you want to track the fiber in, as illustrated in Figure 9.

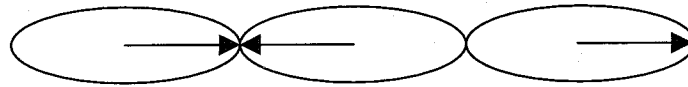


Figure 9 Major eigenvectors may be the opposite of the direction expected

In order to cope with this problem the new vector should be compared with the previous vector used in the numerical integration. If the angle between these vectors is greater than 90° , then the new vector must be inverted.

Normally a fiber is tracked from a seed point. This tracking occurs in two directions: The direction of the major eigenvector and the opposite direction of the major eigenvector. There are two main factors that influence the length of the fiber and the equivalence between the real and the reconstructed fiber:

- The algorithm used to numerically integrate the differential equation.
See paragraph 3.2 for more information
- The termination criteria that determine when to stop fiber tracking
See paragraph 3.3 for more information

3.2. Numerical Integration of a Differential Equation

A single fiber is tracked from an initial point by numeric integration of the differential equation (§3.1), [7], [8]. If p_0 is the initial point, then the relationship between two points p_n and p_{n+1} on the integration path can be summarized by the equation

$$p_{n+1} = p_n + s r_{n+1} \quad (3.2)$$

where s is the step size. r_n is defined differently whether Runge-Kutta is used or Euler's method is used to numerically integrate the differential equation.

3.2.1. Euler's Method

Using Euler's method, r_n is defined as

$$r_{n+1} = \text{sign}(V(p_n) \bullet r_n) V(p_n) \text{ for } n \geq 0$$

$$r_0 = V(p_0) \text{ or } -V(p_0) \text{ In order to track both sides of the starting point} \quad (3.3)$$

$$\text{where } \text{sign}(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ -1, & \text{if } x < 0 \end{cases}$$

$V(x)$ is the eigenvector in point x , adjusted to the voxel size (See §3.2.4). The next point is defined by basically translating the current point along its eigenvector, or the opposite direction of its eigenvector in case the angle between the previous eigenvector and the current eigenvector is greater than 90° . Although Euler's method may appear to be an intuitive method, it is an inaccurate method as is shown in the next figure:

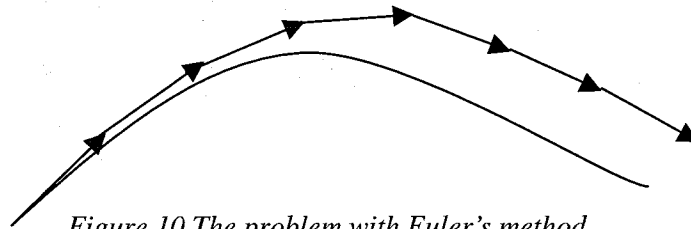


Figure 10 The problem with Euler's method

Even though the direction of the arrow is the direction of the line in its corresponding point, there is a big error in the result of Euler's method. The step's error is $O(s^2)$, thus by decreasing the step size, the accuracy can be increased.

A method that is more accurate for larger step sizes is the Runge-Kutta method.

3.2.2. Runge-Kutta Method

Using the Runge-Kutta method, r_n is defined as

$$r_{n+1} = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \text{ for } n \geq 0$$

$r_0 = V(p_0)$ or $-V(p_0)$ In order to track both sides of the starting point

where $k_1 = \text{sign}(V(p_n) \bullet r_n)V(p_n)$;

$$k_2 = \text{sign}\left(V\left(p_n + \frac{sk_1}{2}\right) \bullet r_n\right)V\left(p_n + \frac{sk_1}{2}\right); \quad (3.4)$$

$$k_3 = \text{sign}\left(V\left(p_n + \frac{sk_2}{2}\right) \bullet r_n\right)V\left(p_n + \frac{sk_2}{2}\right);$$

$$k_4 = \text{sign}(V(p_n + sk_3) \bullet r_n)V(p_n + sk_3);$$

$$\text{sign}(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ -1, & \text{if } x < 0 \end{cases}$$

The used Runge-Kutta method is the classical 4th order Runge-Kutta method, with a small change made in order to make all vectors have an angle of less than 90°, since diffusion is symmetrical.

The Runge-Kutta method requires the calculation of four times as many eigenvectors, and is thus about four times as computationally expensive as Euler's method for the same step size. The step's error for the Runge-Kutta method is $O(s^5)$,

3.2.3. Comparing Euler's Method and Runge-Kutta

When comparing Euler's method and Runge-Kutta, the following test was used:

Calculate the 'real' fiber path using a very small step size. Then by changing the step size gradually, find the step sizes at which Euler's method and the Runge-Kutta method start deviating visibly from the 'real' fiber path.

The result depends to quite a high degree on the dataset used. For a 128x128x60 dataset, there was very little difference between the two methods. For Euler the maximum step size was about 0.3 voxel, while for Runge-Kutta the maximum step size was about 1.2 voxel. Apparently the fibers are too short and too straight, for Runge-Kutta to provide a significant benefit.

Further effort in improving computational efficiency while remaining acceptable accuracy is still possible. Adaptive step sizes can be used: Generally step sizes can be bigger when the angle between the vectors of two corresponding points is small. Also the accuracy is more critical close to the starting point than near the end of the fiber. Fiber length might also be a criterion to adjust the step size to.

Numerical Recipes [7] contains more information about the general comparison between Euler's method and Runge-Kutta, and possibly faster algorithms.

3.2.4. Non Cubic Voxels

Eigenvectors are defined in the APAT coordinate system. Voxel positions in computer memory however use the memory coordinate system defined as

$$\begin{aligned}x_{memory} &= x_{APAT} \frac{x_{scan\ resolution}}{x_{physical\ size}} \\y_{memory} &= y_{APAT} \frac{y_{scan\ resolution}}{y_{physical\ size}} \\z_{memory} &= z_{APAT} \frac{z_{scan\ resolution}}{z_{physical\ size}}\end{aligned}\tag{3.5}$$

During fiber tracking the memory coordinate system is used for efficiency reasons. As a result the eigenvectors need to be converted to the memory coordinate system during fiber tracking. When converting the eigenvectors, it has to be taken into that the step size is not affected. The effective step size is defined as

$$step\ size_{effective} = step\ size_{defined\ by\ the\ user} \min\left(\frac{x_{physical\ size}}{x_{scan\ resolution}}, \frac{y_{physical\ size}}{y_{scan\ resolution}}, \frac{z_{physical\ size}}{z_{scan\ resolution}}\right)\tag{3.6}$$

in the physical coordinate system in the Fiber Tracking tool.

3.3. *Fiber Tracking Termination*

Fiber tracking is not always possible. Only certain tissue types allow fiber tracking. In the brain this typically is white matter: tissue mainly consisting of the axons of neurons. If one would track fibers in tissue types that are not suitable, the result would be erratic and irrelevant. If the user starts fiber tracking in an unsuitable area, or suitable tissue progresses into unsuitable tissue, the fiber tracking process should terminate. Suitable tissues have certain characteristics that distinguish them from other tissue types.

A) Suitable tissues are anisotropic

Because diffusion is directed in areas where you want to track fibers, the relevant tensors must be anisotropic. Fiber tracking can only be performed when a minimal anisotropy is present. E.g. in isotropic data a fiber would show a "random walk" behavior. Therefore a minimal anisotropy is a good threshold to continue the fiber tracking process. The calculation of anisotropy is discussed in paragraph 3.4.

B) Correct fibers will only have small angles between two neighboring voxels

White matter in the human brain is relatively straight. If the fiber tracking results in a highly angulated result, it is incorrect. Fiber tracking should terminate when that variation in the angle is below a set threshold.

This threshold can be the angle between two vectors. If the angle is greater than 90° , then the angle minus 90° is the actual angle between the two voxels, to compensate that eigenvectors are bi-directional.

In literature the absolute inner product of two eigenvectors is often used. Since the magnitude of eigenvectors is always 1, this is the same as $|\cos(\text{angle})|$.

C) Anatomical mask

An anatomical image can be used as a mask. If the intensity of the anatomical image is below a certain threshold, fiber tracking is terminated.

D) Suitable tissues have typical eigenvalues

Tissue types have typical ranges for the major eigenvalue. This property can be used to terminate the fiber tracking algorithm, when the algorithm has left the tissue type that the user is interested in.

In the Fiber Tracking tool, the A, B and C characteristic can be set to control the fiber tracking algorithm.

3.4. Anisotropy as fiber quality indicator

Fiber tracking can only be performed when a minimal anisotropy is present. E.g. in isotropic data a fiber would show a "random walk" behavior. Therefore a minimal anisotropy is a good threshold to continue the fiber tracking process. In particular fractional anisotropy is used in literature as a minimum threshold.

Normal thresholds for fractional anisotropy have values between 0.1 and 0.6.

$$\text{Fractional Anisotropy} = \frac{\sqrt{3 \sum_{i=1,2,3} (\lambda_i - \lambda_{avg})^2}}{\sqrt{2 \sum_{i=1,2,3} \lambda_i^2}} \quad (3.7)$$

$$\text{where } \lambda_{avg} = \frac{\sum_{i=1,2,3} \lambda_i}{3}$$

However, consider this case:

A tensor has these eigenvalues: $\lambda_1 = 2, \lambda_2 = 2, \lambda_3 = 0.1$. A tensor like this is called a planar tensor. It has high diffusion in the plane described by the eigenvectors v_1 and v_2 , but little diffusion in the direction perpendicular to this plane.

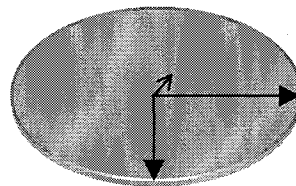


Figure 11 An ellipsoid representing a planar tensor

The eigenvectors v_1 and v_2 can rotate around v_3 , while still describing the same, or a very similar tensor.

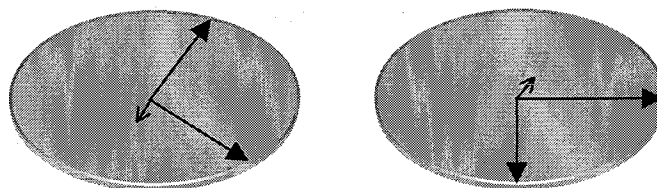


Figure 12 Two similar planar tensors

Reliable fiber tracking is not possible with such high variations in the primary eigenvector, so the quality indicator should give a low value for planar tensors. However, the fractional anisotropy of the above-mentioned tensor is ~0.68. Apparently fractional anisotropy is an unsuitable quality indicator for fiber tracking.

A much better quality indicator is principal or primary anisotropy defined as

$$\text{Principal Anisotropy} = \left(\frac{4(\lambda_1 - \lambda_{avg})^2}{\lambda_1^2 + \lambda_2^2} \right)^{0.4} = \left(1 - \frac{2\lambda_1\lambda_2}{\lambda_1^2 + \lambda_2^2} \right)^{0.4} \quad (3.8)$$

$$\text{where } \lambda_{avg} = \frac{\lambda_1 + \lambda_2}{2}$$

The power of 0.4 is used to provide comfortable contrast for typical neurological images.

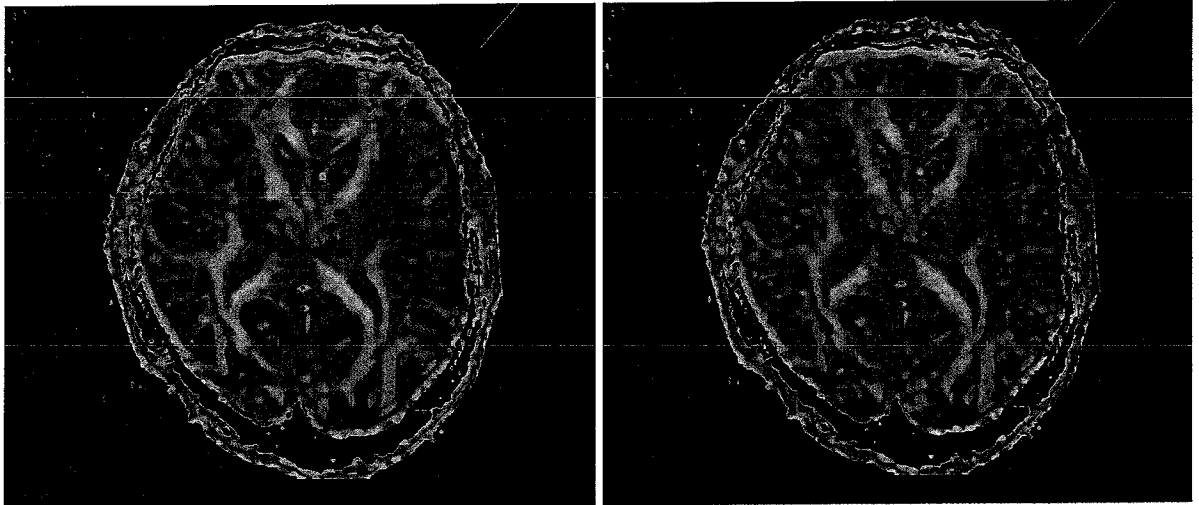


Figure 13 Overview: On the left fractional anisotropy, on the right principal anisotropy

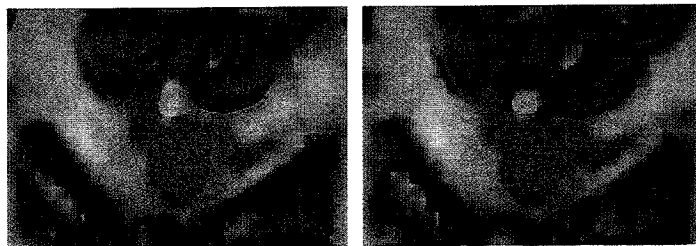


Figure 14 Detail: On the left fractional anisotropy, on the right principal anisotropy

The images above show the difference between fractional and principal anisotropy. With principal anisotropy different structures are more distinct with a clearer separating low anisotropy area in between. Also structures where fiber tracking is not possible are filtered out better using principal anisotropy instead of fractional anisotropy, while areas where fiber tracking is possible remain equally bright compared to fractional anisotropy.

4. Seed Points and Regions of Interest

4.1. Seed Points (Single ROI)

The Fiber Tracking tool allows the user to define a region of interest (ROI) where he wants to start tracking. A simple and quick approach would be that for every voxel in the ROI a fiber is tracked. However, this has limitations.

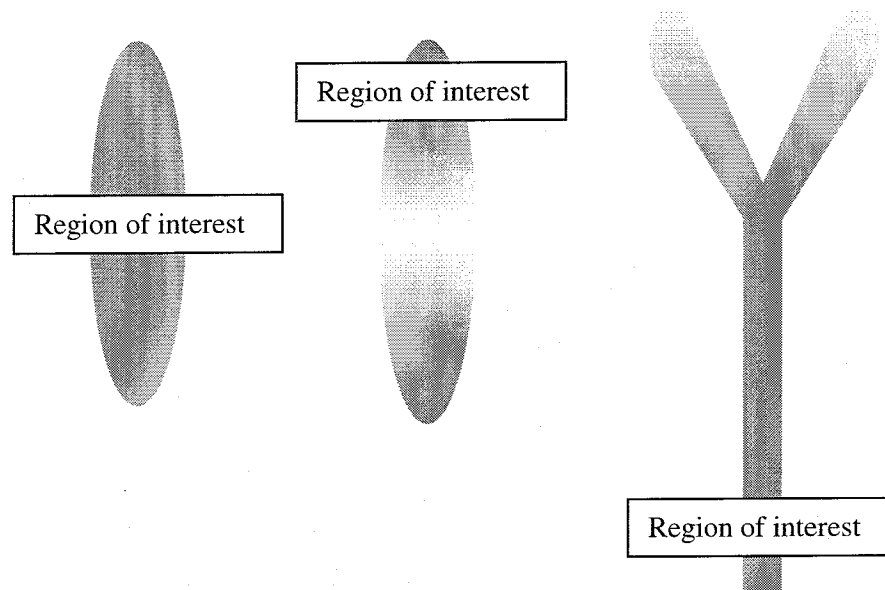


Figure 15 Seed points are vital for the final fiber tracking result

The image above shows the problem with this approach. In the left case the region of interest is set at the widest part of the fiber, and fiber tracking yields good results. In the middle case the situation is not ideal but acceptable. The region of interest is set at a thin part of the fiber, and the fiber found will be less dense when the fiber gets wider. The right case shows another problem. A fiber splits and the region of interest is set quite some distance before the split. Because of the long distance, approximation and measurement errors, there is no guarantee that the fiber tracts are distributed evenly and all splits are coped with effectively.

In summary:

- The density of the seed points in the region of interest should depend on whether the fiber diverges
- Measurement and approximation error cause problems with splits after distances

In [4] it is described that these problems can be dealt with by altering the seed points. Instead of tracking from points from the ROI, seed points are placed evenly in the entire dataset. All seed points are tracked, and fibers will only be visualized if the resulting fiber passes through the ROI. Test trials with this approach in the Fiber Tracking tool showed a great improvement in the results.

- The number of fibers found dramatically increased
- The number of fibers found in close proximity (thickness of a fiber bundle) is a good indicator of anatomical correctness.
- The algorithm is too slow for practical use (Between 2½ minutes and 10 hours, depending on the voxel size)

Not just the fiber tracking time was unacceptably slow, but the large increase in the amount of fibers found also generates performance problems during generation and display of the 3D model.

To cope with these problems, the following measures were taken:

- A new seed point algorithm was developed to reduce the calculation time of the fiber tracking phase. (See paragraph 4.2)
- Multiple ROIs are introduced to reduce the amount of fibers found and improve control over the fiber tracking result. (See paragraph 5.2.2)
- A new quick visualization algorithm was implemented that has results in a smaller polygon count and is very quick, but only suitable for thin fibers. (See paragraph 5.4.3)

4.2. Improved Seed Point Algorithm

The algorithm described in paragraph 4.1 (robust algorithm) that tracks fibers for evenly placed seed points in the entire dataset is too slow for practical use, therefore an improved seed point algorithm (quick algorithm) was developed:

Mark every voxel that it has not been tracked.

Prepare a stack of voxels that have to be tracked.

Prepare an empty stack of voxels that are the starting points of valid fibers.

Using bit-masks, mark for each voxel which ROIs it belongs to.

For each voxel in the ROI, add this voxel and all surrounding voxels to the stack of voxels that have to be tracked.

Track the voxels in the stack:

Remove the voxel from the stack of voxels that have to be tracked.

Check whether the voxel has already been tracked. If it has not been tracked:

If the fiber found for this voxel is valid (Passes through the ROI in case of single ROI fiber tracking, or passes through all the ROIs in case of multiple ROI fiber tracking):

Add all surrounding voxels to the stack of voxels to be tracked.

Add the voxel to the stack of voxels that are the starting points of valid fibers.

Mark that the voxel has been tracked.

Although the robust algorithm will find more fibers than the quick algorithm, it is too slow. Generally the quick algorithm will find all the major fiber pathways that the robust algorithm finds, and is therefore the recommended algorithm.

The figures below show the differences between the quick and the robust algorithms.

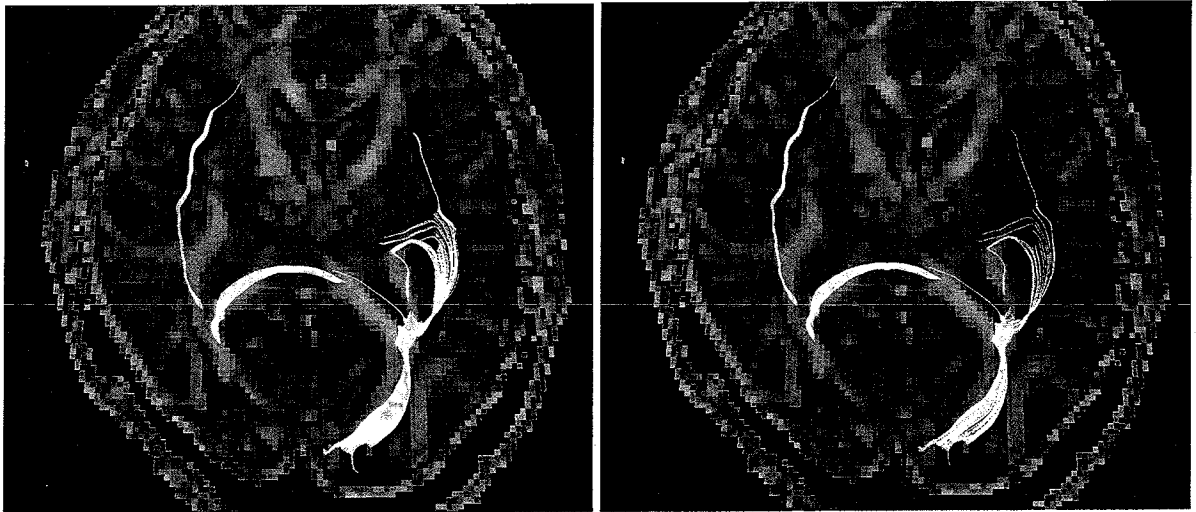


Figure 16 Results of the Improved Seed Point Algorithm

In the figures above all fibers passing through a single voxel in the corpus callosum are shown. Small voxels are used. In the left image the resulting fibers of the robust algorithm are shown. In the right image the resulting fibers of the quick algorithm are shown. Calculation time of the robust algorithm: over 10 hours. Calculation time of the quick algorithm: 2 minutes.

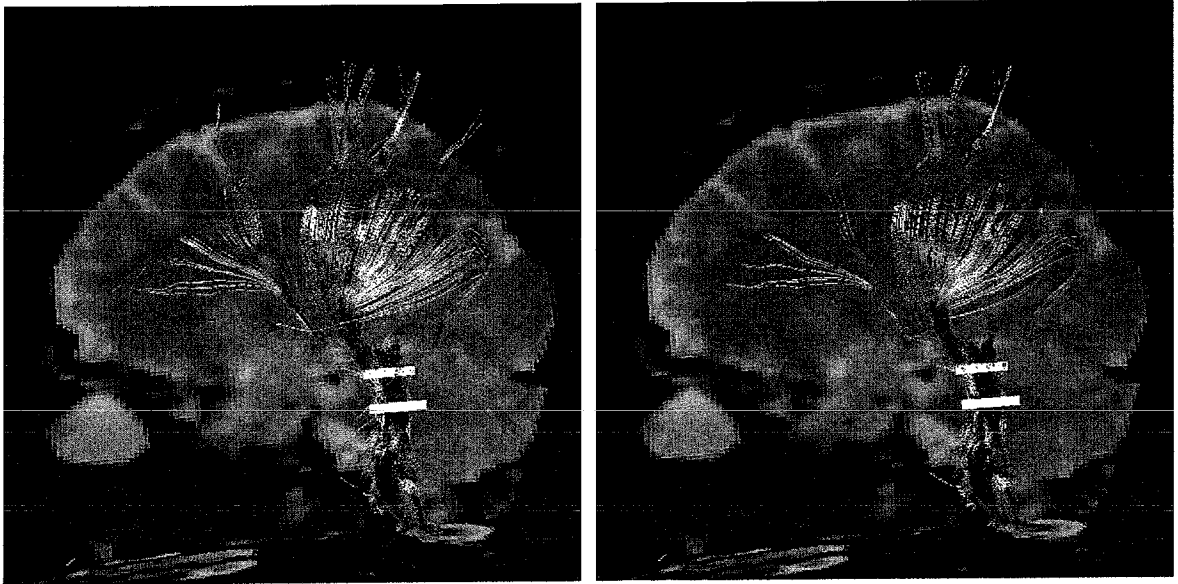


Figure 17 More Results of the Improved Seed Point Algorithm

In the figures above all fibers passing through both the two yellow ROIs is shown. Large voxels are used. In the left image are the resulting fibers of the robust algorithm are shown. In the right image are the resulting fibers of the quick algorithm are shown. Calculation time of the robust algorithm: 2½ minutes. Calculation time of the quick algorithm: 3 seconds.

4.3. Multiple ROIs

The new seed point algorithm from paragraph 4.1 finds a very large amount of fibers passing through a single ROI. It is desirable that the user has more control over the generation of fibers for two reasons:

- Filtering out the data that the user wants to see
Often the user knows more than one landmark of the structure he wishes to track. By defining multiple ROIs undesirable branches can be filtered out.
- Limiting the amount of fibers found to increase performance

Implementation

For performance reasons a maximum of eight different ROIs are allowed. (A bit mask of one byte can be used for marking the ROIs as described in 4.2)

Both the robust and the quick algorithm are available in the Fiber Tracking tool.

Interactivity

For more information on how the user can define ROIs, see paragraph 5.2.2.

5. Visualization and User Interaction

The Fiber Tracking tool uses one integrated screen for both visualization and user interaction, shown in Figure 18. A single integrated screen increases ease of use and interactivity of the program.

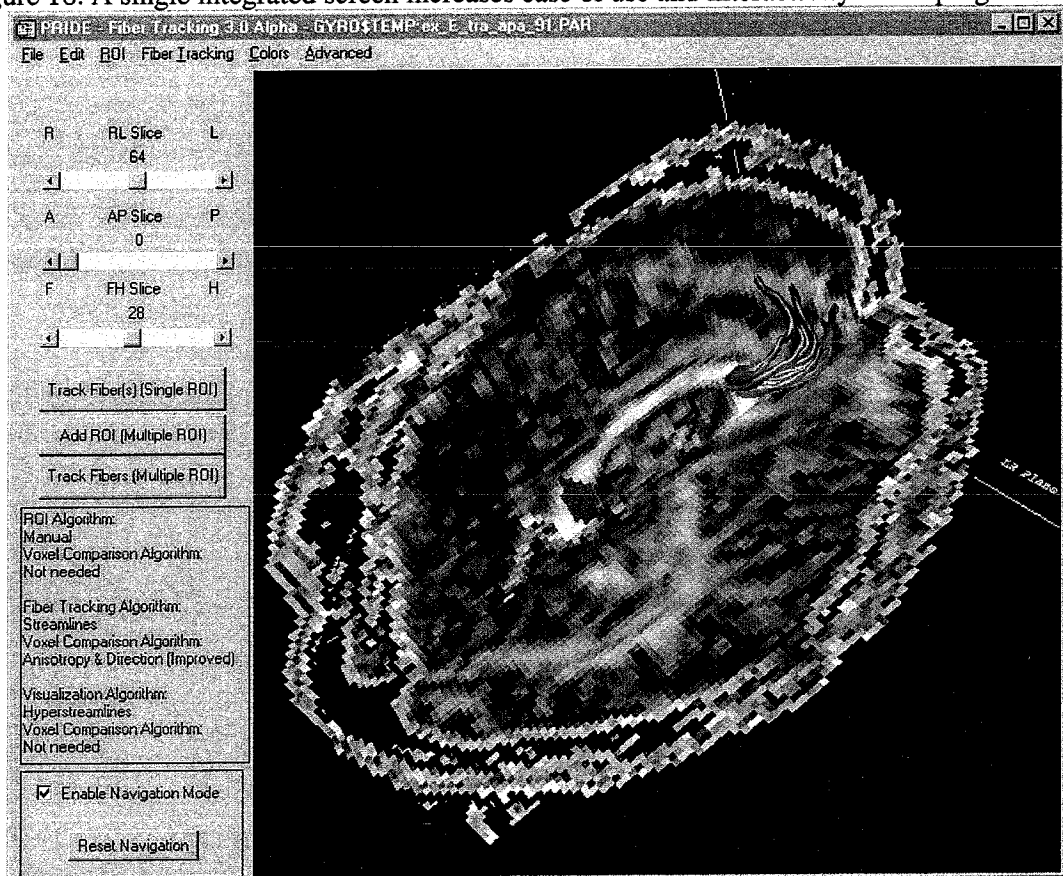


Figure 18 The Main Screen of the Fiber Tracking tool

The main controls are shown in the left. The right shows the 3D model, with slices of the dataset, a ROI (purple) and some fibers.

5.1. The 3D Model

The 3D model is used to visualize all graphical objects (slices, ROIs and fibers). Two constant components of the 3D model are the slices and the axis. Which slices are shown is controlled by the main controls. All available color mapping schemes can be used to visualize the slices. For more information about the available color mapping algorithms, see paragraph 5.3.

Interactivity

The 3D model is used for a great deal of interaction as well: ROI drawing, Real Time fiber visualization or parameter adjustment using pop-up menus when the user right-clicks on a fiber or ROI. The user can rotate the 3D model with the left mouse button. The keyboard is used to zoom and pan the 3D model.

Via the menu bar various actions are available such as:

- Loading and Saving of datasets. Loading of fibers. Saving of images of the 3D model.
- Setting the algorithms and their parameters
- Default algorithm setting for new fibers
- Basic navigation
- Fiber and ROI removal

5.2. ROIs

The 3D model is used to draw ROIs. There are four different ROI types in the Fiber Tracking tool:

- **Single Point**
The user can simply right click on a slice in the 3D model to define a single voxel ROI.
- **Automatic (2D)**
The user can right click on a slice in the 3D model to define a 2D ROI. The user defines a seed point by a right click on the desired location of a visualized slice. A flood-fill type algorithm in the current slice is then used to calculate the 2D ROI. To compare whether voxels should or should not be included in the ROI, the same algorithm is used as for stopping fiber tracking. (See paragraph 3.3)
- **Automatic (3D)**
This is essentially the same as the 2D variant of the automatic algorithm, except the flood-fill type algorithm fill is in 3D and is not restricted to the selected slice.
- **Manual**
The user can draw polygons or other 2D primitives on the image of the 3D model. For each point in the manually drawn ROI its 3D coordinates in the 3D model are calculated to define the ROI.

5.2.1. Single ROIs

When the user wants to quickly track some fibers, he uses the single ROI mode. For each voxel in the ROI a fiber is tracked and visualized. The ROI is visualized by highlighting the appropriate parts of the slice.

5.2.2. Multiple ROIs

Multiple ROIs uses the seed point algorithms of paragraph 4.2. Both the quick and the robust algorithm can be used for tracking.

First the user must create one or more ROIs. Then the user must explicitly start fiber tracking, when the desired ROIs are created.

Visualization

When the user has defined multiple ROIs, 3D axis-aligned boxes are used to visualize these.

Interactivity

To improve the interactivity of the Fiber Tracking tool, the user is enabled to:

- Change the color mapping of the ROI visualization
- Delete the ROI
- Merge ROIs.

Since fibers are required to pass through all ROIs (AND relationship), the merge option can be used as an OR relationship to allow great flexibility in the defining of multiple ROIs.

5.3. *Color Mappings*

One of the main requirements of the Fiber Tracking program is to investigate the clinical possibilities of fiber tracking and diffusion data. Apart from investigating the capabilities of fiber tracking, the program also provides some new color mapping algorithms to provide a human-viewable interpretation of the diffusion data. We define a color mapping algorithm as an algorithm that:

- Calculates relevant scalar values from the measured data.
- Transforms the relevant scalar values into a color or a gray value.

Clinically known color mapping algorithms visualize the following:

- Raw images: diffusion intensity of the different measured directions directly.
- Isotropic images: the intensity of diffusion that is present in all directions.
- Fractional anisotropy images: the shape of diffusion.
- Relative anisotropy images: A combination of both shape and intensity.
- Anatomical images: Images of another scan can be useful since these can be of higher quality than diffusion scans, which show anatomical landmarks better.

These gray-scale images can be color coded by using the primary eigenvectors. A typical color coding is red if the vector is directed left/right, green if the vector is directed anterior/posterior and blue if the vector is directed feet/head. By using the same color scheme for direction regardless of the scanning technique, typical landmarks are provided which the user easily identifies.

A color mapping algorithm already introduced in 3.4 is principal anisotropy. This technique is very suitable for fiber tracking, since principal anisotropy corresponds with the quality of fibers.

5.3.1. Planar Anisotropy

Another new color mapping algorithm is planar anisotropy. Planar anisotropy corresponds with the planar shape of a diffusion tensor. Figure 11 show a tensor with high planar anisotropy. The third eigenvector can be used to color code the color mapping.

$$\begin{aligned} \text{Planar Anisotropy} &= \left(1 - \left(\frac{4(\lambda_1 - \lambda_{avg1})^2}{\lambda_1^2 + \lambda_2^2} \right)^{0.4} \right) \left(\frac{4(\lambda_2 - \lambda_{avg2})^2}{\lambda_2^2 + \lambda_3^2} \right)^{0.4} \\ &= \left(1 - \left(1 - \frac{2\lambda_1\lambda_2}{\lambda_1^2 + \lambda_2^2} \right)^{0.4} \right) \left(1 - \frac{2\lambda_2\lambda_3}{\lambda_2^2 + \lambda_3^2} \right)^{0.4} \end{aligned} \quad (5.1)$$

$$\text{where } \lambda_{avg1} = \frac{\lambda_1 + \lambda_2}{2};$$

$$\lambda_{avg2} = \frac{\lambda_2 + \lambda_3}{2}$$

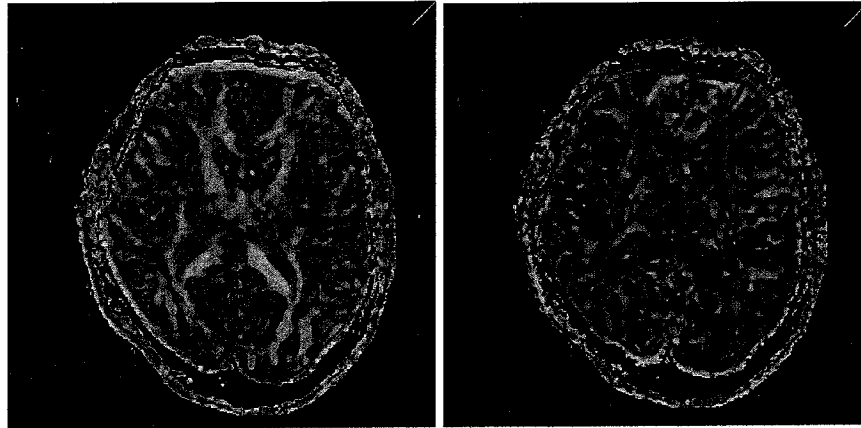
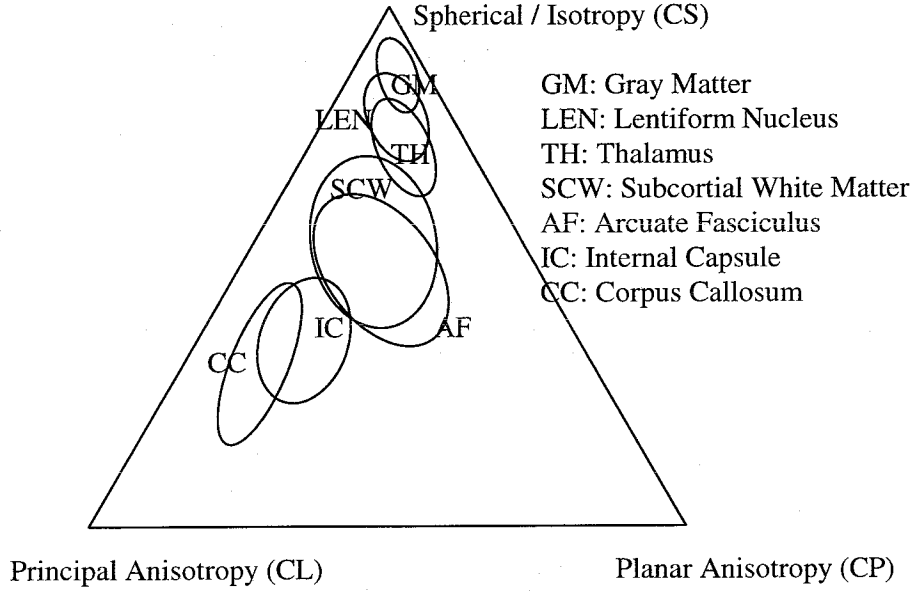


Figure 19 Principal Anisotropy (left) and Planar Anisotropy (right)

Planar anisotropy (Figure 19) shows information not visible with known color mapping algorithms. However, there is no known clinical purpose for this technique yet.

5.3.2. Tensor Shape

In Figure 20 it is shown that various tissue types have characteristic tensor shapes [5].



*Figure 20 The Tensor Shape of various Tissues
The circles show typical ranges for CS/CP/CL for certain tissues.*

Ideally, information about all the possible shapes is given in one image, to provide the maximum amount of information to the viewer of the image at a glance. The "Tensor Shape" color mapping algorithm directly shows tensor shape by using the formula

$$c_{red} = \lambda_1 CL^{0.5}$$

$$c_{green} = \lambda_1 CP$$

$$c_{blue} = \lambda_1 CS^{2.5}$$

where

$c_{red}, c_{green}, c_{blue}$ is the amount of red, green and blue of the color that is displayed.

$$CL = \frac{\lambda_1 - \lambda_2}{3\lambda_{avg}}, \text{ Where the maximum and minimum values allowed are 1 and 0.}$$

$$CP = \frac{2(\lambda_2 - \lambda_3)}{3\lambda_{avg}}, \text{ Where the maximum and minimum values allowed are 1 and 0.}$$

$$CS = \frac{\lambda_3}{\lambda_{avg}}, \text{ Where the maximum and minimum values allowed are 1 and 0.}$$

$$\lambda_{avg} = \frac{\lambda_1 + \lambda_2 + \lambda_3}{3} \tag{5.2}$$

The powers of 0.5 and 2.5 are used to provide comfortable contrast between the color components for typical neurological images.

CS/CP/CL are multiplied with λ_1 to provide additional information about the diffusion strength.

Examples of this color mapping are shown in Figure 21.

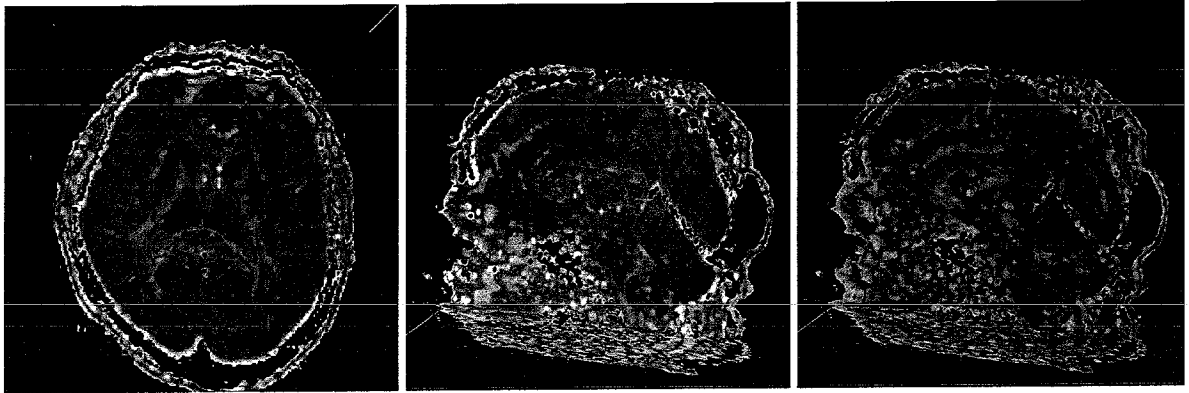


Figure 21 Tensor Shape Color Mapping

The left and middle image show the tensor shape color mapping. The right image shows a color-coded fractional anisotropy color mapping.

This color mapping algorithm provides a very good segmentation of the various tissue types. In particular the middle image shows information that is not visible on the color-coded fractional anisotropy color mapping on the right.

5.4. Fibers

Fiber visualization is one of the key requirements of the Fiber Tracking tool. There is no established method for fiber visualization. Evaluation of an early version of the Fiber Tracking tool showed that there are three different situations requiring different visualization methods:

- A single fiber or a few fibers are visualized
Performance is no issue, and there is enough display space to perform a high detail visualization
- A real-time fiber is visualized
Although enough display space is available, performance is an issue here
- A great number of fibers are visualized
Display space is limited, so high detail visualizations are pointless. Performance is critical.

More information about these methods is given in the next paragraphs.

5.4.1. Single Fibers (*Hyperstreamlines*)

A typical visualization for tensors is hyperstreamlines [6]. A hyperstreamline is a geometric primitive (such as an ellipsoid or a cross) that sweeps along one of the eigenvector fields while stretching in the transverse plane under the combined action of the two orthogonal eigenvector fields. Color usually represents the amplitude of the longitudinal eigenvalue. Because the tube-like nature of fibers corresponds to some extent with hyperstreamlines, it is logical to use this technique for fiber visualization.

The figure below shows a hyperstreamline representing a fiber.

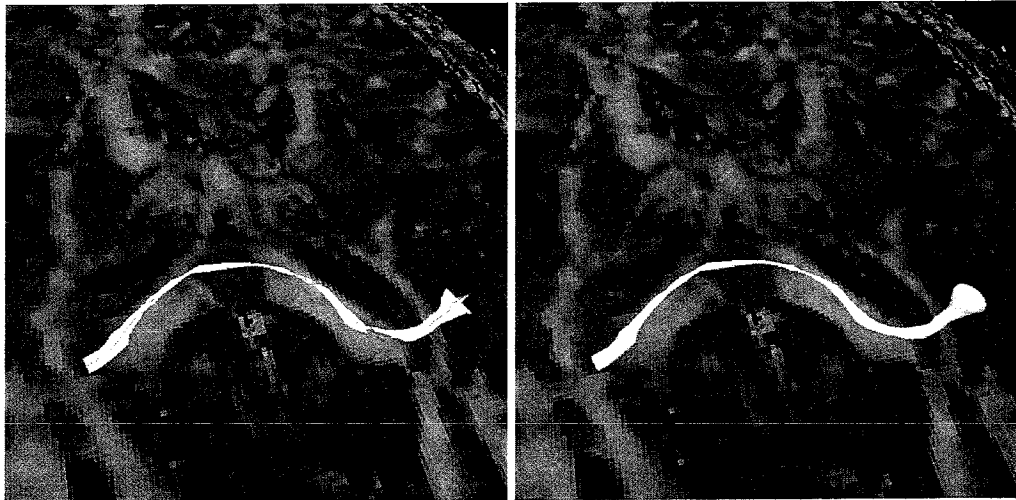


Figure 22 Two Hyperstreamline Visualization

The image on the left shows a hyperstreamline in helix form. The image on the right shows the same hyperstreamline in ellipsoid form. Yellow is used as a constant color to ensure a high contrast with the background slices.

There are several problems with this representation:

- The visualization is thin when the anisotropy is high. This means that fibers that are likely to be anatomically correct are thin, while fibers that are less likely to be anatomically correct are thick.
- The medium and minor eigenvectors can rotate erratically around the major eigenvector, making the cross representation needlessly distracting and ugly.
- There is no limit to the width of the tube, since the medium and minor eigenvalues are not limited to strict ranges.

To correct these problems, an alternative representation can be used. Instead of letting the eigenvalues of the medium and minor eigenvalues determine the width of the ellipsoid, anisotropy is used. This results in a circular appearance as shown in the next figure. It is debatable whether the new variant can be considered a 'hyperstreamline', but for ease of reading we will use this name for the rest of this document for this new variant as well.

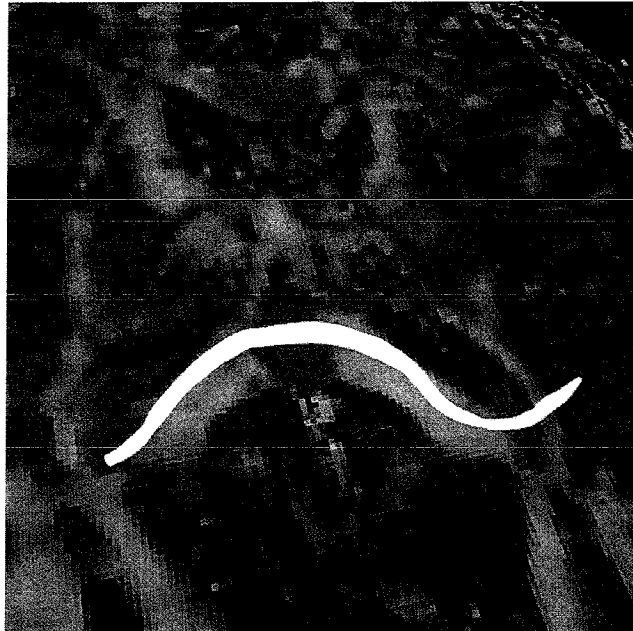


Figure 23 Improved hyperstreamlines

This new representation is more stable and better suitable for the visualization of fibers. All issues mentioned earlier are resolved, resulting in the default used visualization technique in the Fiber Tracking tool.

Interactivity

To improve the interactivity of the Fiber Tracking tool, the user is enabled to:

- Change the color mapping algorithm
- Delete the fiber
- Change the visualization technique (e.g. hyperstreamlines traditional, hyperstreamlines anisotropy, quick, etc.)
- Change the fiber tracking algorithm
- Save the fiber

by simply right-clicking on the desired fiber in the 3D model.

5.4.2. Real Time Fiber

The real time fiber is a fiber that the user can track and see instantly by moving the mouse pointer over an area of interest in the 3D model. There is plenty of screen space to display the real time fiber, but performance is an issue. In order to maximize performance, a simple polyline representation is used.

The color mapping algorithm can be set to any custom color map. Right clicking on the real time fiber will remove the fiber.

5.4.3. *Many Fibers (Quick Fiber Visualization)*

When using the multiple ROI tracking method, a great number of fibers will be found. In order to visualize that many fibers these criteria are used:

- Fibers are thin, so providing detailed information is not possible
- Construction time (Creating the polygons) and rendering time (Displaying the polygons) are crucial.

The swept cross structure is a 3D structure with minimal complexity that has a guaranteed 3D fill. A polyline did not offer shading and was not used. A swept line structure is half as complex as the swept cross structure, but can disappear under the wrong camera angle, and thus could not be used.



Figure 24 A cross and a line polygon structure

As with the hyperstreamlines, the right mouse button can be used to change the color mapping, merge fibers, delete fibers, change the used algorithms, etc. via a popup menu.

6. Implementation

6.1. Overview

The Fiber Tracking tool is designed to run on a PC which is according to Pride specifications: P III 500 with 128 MB Memory and a 3-button mouse.

A 3D accelerator is not required, which influenced the 3D visualization design decisions, since performance and interactivity are very important.

The initial version was written in IDL 5.4, which is an interpreted RAD tool. Since IDL is rather slow, performance critical parts were written in C++ as a DLL that is called from IDL.

<i>Code Size</i>	Number of Lines	kb
C++	2800	80
IDL	11000	386

Table 2 Code Size of the Fiber Tracking tool

For the 3D visualization IDL's Object Graphics Library is used. The library is based on OpenGL and offers a subset of the OpenGL interface. The 3D object graphics library is relatively new in the IDL package and there are some bugs that limit the possibilities. In particular, transparency is not properly supported in IDL 5.4.

6.2. Performance

IDL supports both a hardware accelerated and a software implementation of OpenGL. Since a 3D accelerator is not required, the visualization features need to be limited:

- When rotating, zooming and panning the 3D model a wireframe is used to increase performance.
- Bilinear filtering is optional
- Maximized use of simple polygon structures

When no complex fibers are visualized, performance is acceptable (± 10 fps) on the minimal Pride requirements. When a 3D accelerator is present, visualization is fluent.

Other areas that have major impact on performance and interactivity are:

- Improved Seed Point Algorithm (See paragraph 4.2)
- Quick generation of fibers (See paragraph 5.4.3)
- Real time fiber (See paragraph 5.4.2)
- Interpolation (See paragraph 2.3)
- Fiber Tracking (See paragraph 3.1)

7. Conclusions & Recommendations

7.1. Conclusions

The Fiber Tracking tool allows a relative quick and intuitive visualization of fibers, in particular in the human brain. It also serves as a general diffusion visualization tool, even for areas where fibers can not be tracked.

The Fiber Tracking tool has been sent to a dozen clinical sites to experiment with the tool. The most interesting results are shown in Figure 25, which shows that the tool is useful for clinical scientists.

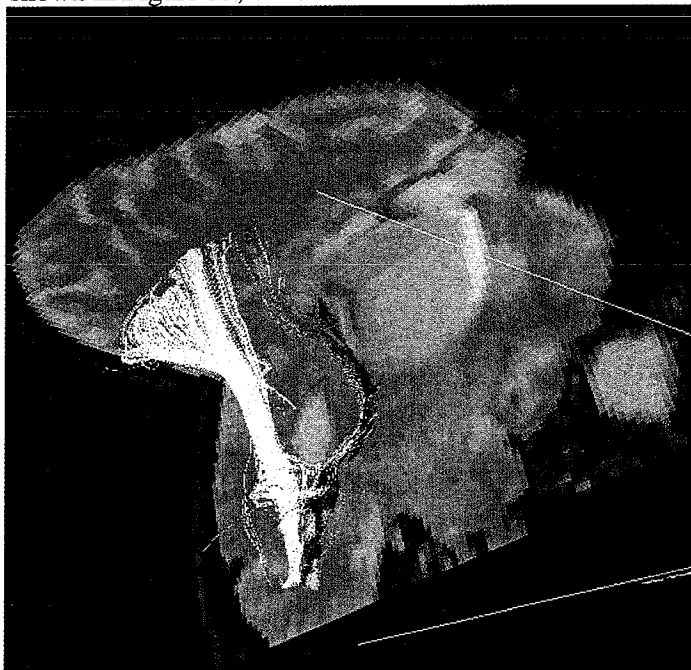


Figure 25 Clinical Application of Fiber Tracking

The left/right Cortico-Spinal-Tract (CST) is shown above. It can be seen that the left-CST is being pushed away from the tumor, which is vital information for the neurosurgeon. Dr. Kei Yamada and Dr. Osamu Kizu from the Kyoto Furitsu Hospital in Japan created this case.

Dr. Kei Yamada: "It is fun to turn around the images in 3-D space. This is almost like playing a game. It is already helping the med school students to understand the anatomy."

The Fiber Tracking tool supports a wide variety of datasets, is quick, is interactive, is user friendly, is robust and supports a fair number of well-established diffusion visualization techniques. However, there are still some areas where the Fiber Tracking tool can be improved. These are discussed in the next paragraph.

7.2. *Recommendations*

- Since the Fiber Tracking tool is a Pride tool, it is only available to clinical sites with Pride. To increase the availability of the Fiber Tracking tool, it should be implemented on the MRI Scanner.
- IDL is not a suitable language for a complex performance critical tool such as the Fiber Tracking tool. A implementation in C++ with OpenGL or Direct3D would be a better choice.
- The improved seed point algorithm returns fibers that are very complex and cause a slow rendering time. A quick polygon simplification algorithm should be used if possible to improve the display performance.
- Improve the fiber tracking speed by using a better algorithm for numerical integration. See paragraph 3.2.3 for more information.
- The Fiber Tracking tool does not support adjusting of the intensity of color maps. This limits the use of anatomical datasets and other color mapping algorithms such as tensor shape and relative anisotropy. The Fiber Tracking tool should support leveling of the intensity of color maps individually, for each color map that requires it. This should result in a good visualization even if different color mappings are displayed simultaneously.
- The Fiber Tracking tool only supports other techniques to a limited extent through imported 'anatomical' datasets. The Fiber Tracking tool should support more formats to import, for instance fMRI data.
- Normally when a diffusion scan is made, only six diffusion directions are measured. However, it is not unlikely that scans of more than six directions will be used more frequently in the future. Although the Fiber Tracking tool supports these scans, it uses the tensor model, so any additional information provided by these scans will not be available. By using a better model, the Fiber Tracking tool should be able to cope more effectively with crossing fibers with this type of scans.
- With the tensor model, it is also possible to deal better with crossing fibers. In Appendix C some potential improvements to the fiber tracking algorithm are discussed.
- Various color mapping that use different colors yield undesirable results when gray-scaling such an image. These color mapping can possibly be improved, so gray-scaled images still yield useful results.

8. References

1. C-N Chen, D I Hoult. Biomedical Magnetic Resonance Technology. Medical Science Series. ISBN 0-85274-118-9
2. Merboldt KD, Hanicke W, Frahm J. Self-diffusion NMR imaging using stimulated echoes. J Magn Reson 1985;64:p. 479-486
3. Gilbert Strang. Linear Algebra and Its Applications. Academic Press, Inc. Orlando 1976:Chapter 5.5
4. Susumu Mori et al. Fiber-Tracking of Human Brainstem: Validity, Reproducibility, and Tract Properties. Proc. Intl. Soc. Mag. Reson. Med 9 (2001)
5. Andrew L. Alexander, Khader Hasan, Gordon Kindlmann, Dennis L. Parker, Jay S. Tsuruda. A Geometric Analysis of Diffusion Tensor Measurements of the Human Brain. Magnetic Resonance in Medicine 44;p. 283-291
6. Thierry Delmarcelle, Lambertus Hesselink. Visualizing Second Order Tensor Fields with Hyperstreamlines. IEEE Computer Graphics and Applications July 1993; Vol. 13, No 4: p. 25-33
7. Numerical Recipes in C. http://www.ulib.org/webRoot/Books/Numerical_Recipes/bookc.html. Chapter 16.
8. C.R. Tench et al. White Matter Mapping Using Diffusion Tensor MRI. Magnetic Resonance in Medicine 47;p. 967-972
9. Mori S, Crain BJ, Chacko VP, van Zijl PCM. Three dimensional tracking of axonal projections in the brain by magnetic resonance imaging. Ann Neurol 1999;45:265-269.
10. Stefan Skara et al. Condition Number as a Measure of Noise Performance of Diffusion Tensor Data Acquisition Schemes with MRI. Journal of Magnetic Resonance (2000) 147, p.340-352.

Appendix A. Original Fiber Tracking Program

A Fiber Tracking program by Susumo Mori was provided as a starting point for this thesis. This appendix will provide a brief summary of this program.

The used fiber tracking algorithm is based on the FACT-Method [9].

The program is divided in several phases:

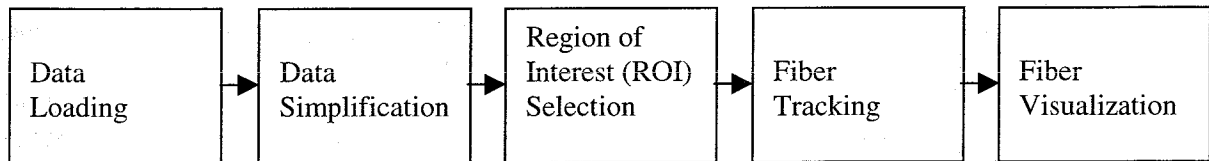


Figure 26 Overview of Susumo Mori's Fiber Tracking Program

First the data is loaded. This is done via the Pride Libraries from either the local harddisk or from the Pride database.

In the data simplification phase, the Tensor field is reduced to a vector field. This simplification allows for easier tracking and visualization in later phases.

If the Fiber Tracking algorithm would be applied to the entire dataset, the resulting fiber will be so large and complex, that the user is unable to interpret the visualization. To prevent this from happening, the algorithm is only applied to a limited region of interest. The selection of the region of interest is done in the region of interest selection phase. The user is presented with a 2D slice of the vector field, from which the user can select a region of interest. This can be done in two different ways. The user can enter a 2D polygon with the mouse. In the second method the user selects a single point, which is then expanded recursively by comparing the vector data of the neighboring pixels.

In the fiber tracking phase, the program will track the fiber for each voxel in the region of interest.

In the fiber visualization phase, the found fiber is visualized.

All phases will be described in more detail in the following sections.

A.1. Data Loading

In the first phase the data is loaded. This is done via the Pride Libraries from either the local harddisk or from the Pride database.

A.1.1 Data Simplification Phase

While the data is being loaded, the program will simplify the data. The tensor field is reduced to a vector field. This simplification allows for easier tracking and visualization in later phases. The major eigenvector of a tensor determines the direction of the corresponding vector, while the fractional anisotropy of the tensor determines the length of the corresponding vector.

This simplification is done with the following algorithm:

1. Calculate the eigenvectors and eigenvalues of the tensor field
2. For each major eigenvector \mathbf{ev} of the current voxel \mathbf{cv} , the fractional anisotropy is calculated with the following formula:

$$Anisotropy(cv) = \begin{cases} \sqrt{\frac{3 * \sum_{i=1,2,3} (\lambda_i - \lambda_{avg})^2}{2 * \sum_{i=1,2,3} \lambda_i^2}}, & (Trace(cv) \geq 0.10) \\ 0.10, & (Trace(cv) < 0.10) \end{cases}$$

$$Where Trace(cv) = \sqrt{\frac{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}{3}}$$

$$Where \lambda_{avg} = \frac{\sum_{i=1,2,3} \lambda_i}{3}$$

In case the trace of the tensor of this voxel is too small, the eigenvector is considered to be noise, and the anisotropy will be set to 0.10.

3. Discard all tensor data: The medium and minor eigenvectors and all eigenvalues.
4. The vector field is defined by the vector \mathbf{ev} which represents the direction of the vector in the vector field, and the anisotropy which represents the length of the vector.

The algorithm will discard the tensor field from this point and use the calculated vector field for the rest of the calculations. There are various issues with this approach:

- Medium and minor eigenvectors are discarded. The loss of data reduces the number of available algorithms to compare voxel similarities, and thus reducing the maximum effectiveness of fiber tracking.

A.2. Region of Interest Selection Phase

If the Fiber Tracking algorithm would be applied to the entire dataset, the resulting set of fibers will be so large and complex, that the user is unable to interpret the visualization. To prevent this from happening, the algorithm is only applied to a limited region of interest. The selection of the region of interest is done in the region of interest selection phase.

In the region of interest selection phase, the user is presented with a 2D slice of the vector field, from which the user can select a region of interest. This slice is oriented on the XY, YZ, or XZ plane. The slice is represented in the following manner: Each pixel represents its corresponding vector. The vector size controls the luminance of the pixel, while the orientation of the vector controls the color of the pixel. The amount of direction of the major eigenvalue of the pixel in the X, Y, and Z directions controls the level of red, green and blue correspondingly.

Region of interest selection can be done in two different ways. The user can enter a 2D polygon with the mouse. In the second method the user selects a single point, which is then expanded recursively by comparing the vector data of the neighboring pixels. Both alternatives will be discussed in the following sections.

A.2.1 2D Area Selection

The user selects a 2D area by clicking the corner points of a convex polygon. This is a very simple and intuitive technique for selecting a region of interest. The main drawback of this technique is that it is impossible or difficult to precisely select the edges of the fiber the users wants to track, and that no automatic algorithm is used to improved the results near these problem areas.

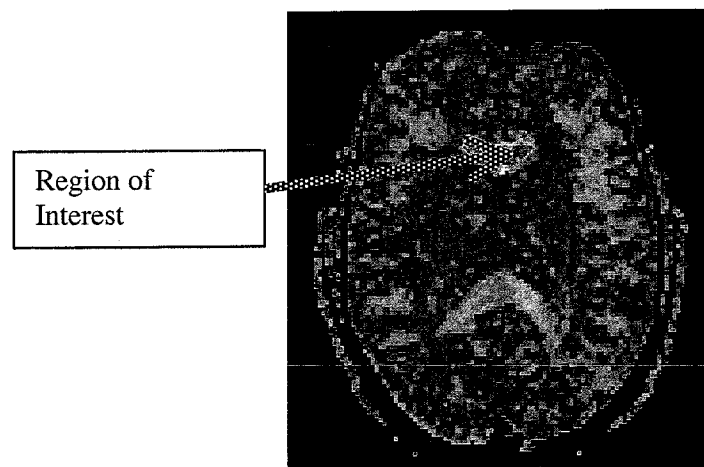


Figure 27 The user is shown a slice. He enters a region of interest by entering a polygon with the mouse

A.2.2 *Single Point Selection*

In this method the user selects a single point, which is then expanded recursively by comparing the vector data of the neighboring pixels. This expansion is done in the 3D vector space of the current slice and the two neighboring slices if available.

1. In the 3D vector space of the current slice and the two neighboring slices (if available), the region of interest is recursively expanded if for the adjacent point **ap** and the current point **cp** the following expression evaluates to True:

$$Anisotropy(ap) > ani_{threshold} \wedge \sum_{i=0}^2 (|\mathbf{ev}(ap) \bullet \mathbf{ev}(cp)|) > vec_{threshold}$$

$$Where [a, b, c] \bullet [d, e, f] = [a d, b e, c f]$$

This set of points is used as a start for the Fiber Tracking phase.

A.3. Fiber Tracking Phase

In the fiber tracking phase, the program will track the fiber for each voxel in the region of interest. The algorithm used does not track increases in the thickness of the fiber, nor does it track possible splits in the fiber.

For each voxel in the region of interest, the following fiber-walking algorithm is used for each voxel in the ROI:

1. Walk in the direction of its major eigenvector until a new voxel is reached.
2. A new voxel nv is reached from the current voxel cv , if the following expression is True, continue fiber tracking until a new voxel is reached. If the expression evaluates to False, stop fiber tracking.

$$Anisotropy(nv) \geq whitematter_{threshold} \wedge \sum_{i=0}^2 (|\mathbf{ev}(nv) \bullet \mathbf{ev}(cv)|) \geq energy_{threshold}$$

$$Where [a, b, c] \bullet [d, e, f] = [a d, b e, c f]$$

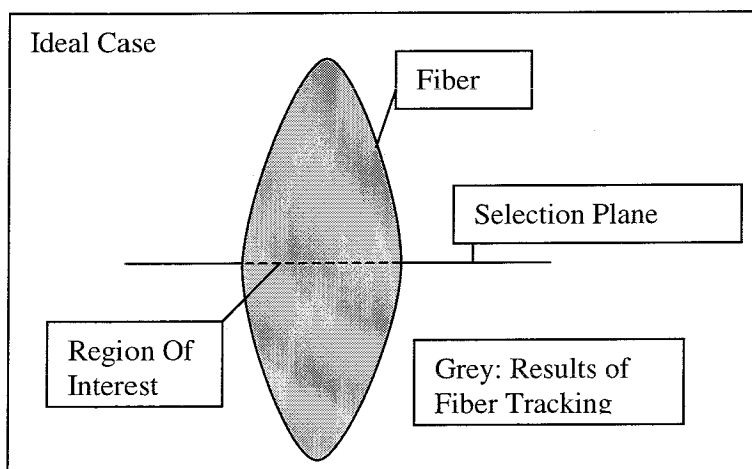
3. Track in the opposite direction of the starting point in step 1. Because eigenvectors are symmetrical, fiber tracking needs to be performed in both directions.

Issues with this approach:

- As explained in the data simplification phase, better criteria can be used to compare voxel similarity.
- The fiber-tracking algorithm is rather limited. It only tracks the voxels in the ROI in the direction of the major eigenvector. This method is only effective if the axes of the selection plane are orthogonal to the fiber and ROI Selection is done there where the fiber is the thickest. This effect is shown in the following diagrams. The effect would not occur if tracking would also be done in the directions of the other eigenvectors.

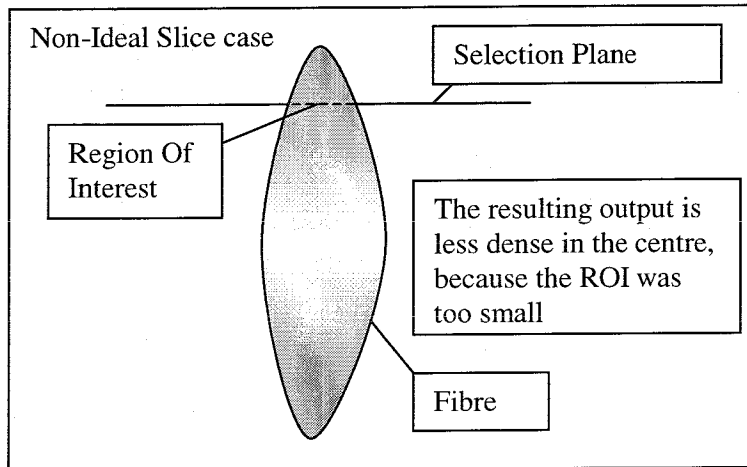
A.3.1 The Ideal Case

In the ideal case, the selection plane is orthogonal to the fiber direction and is oriented there where the fiber is the thickest.



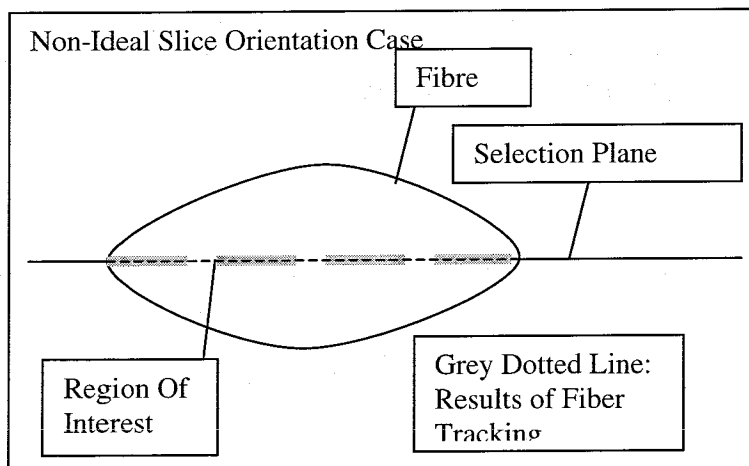
A.3.2 *Non-Ideal Slice Case*

In the non-ideal slice case, the ROI slice is not located where the fiber is thick, and as a result, only parts of the fiber will be tracked where the fiber is thicker.



A.3.3 *Non-Ideal Slice Orientation Case*

In the Non-Ideal Slice Orientation Case, the selection plane is parallel to the direction of the fiber. As a result, only very little of the fiber is actually discovered.



A.3.4 *Normal Case*

Normally, the fiber is neither orthogonal nor parallel to the selection plane, and the result of Susumo Mori's algorithm will provide reasonable results. Nevertheless the result can be improved by using better voxel comparison algorithms, and can sometimes be incorrect due to limitations of the fiber tracking phase and ROI selection phase. This should be avoided at all times due to the critical nature of medical systems. It is recommended to improve the algorithm before it will actually be used for anything other than research purposes.

A.4. *Fiber Visualization Phase*

The tracked fiber is visualized by a 3x3 matrix of images. A column contains images of the three planes. In a row of this matrix, various representations of the slice are shown. First an image with the anatomical data, overlaid with bright red when a fiber is present at that voxel. The second image shows the vector field, where the vector direction determines the color and the length the luminance of that pixel. The third image shown the ADC_{iso} overlaid with bright red when fiber is present at that voxel.

The user can browse through the various slices by clicking on coordinates in the images.

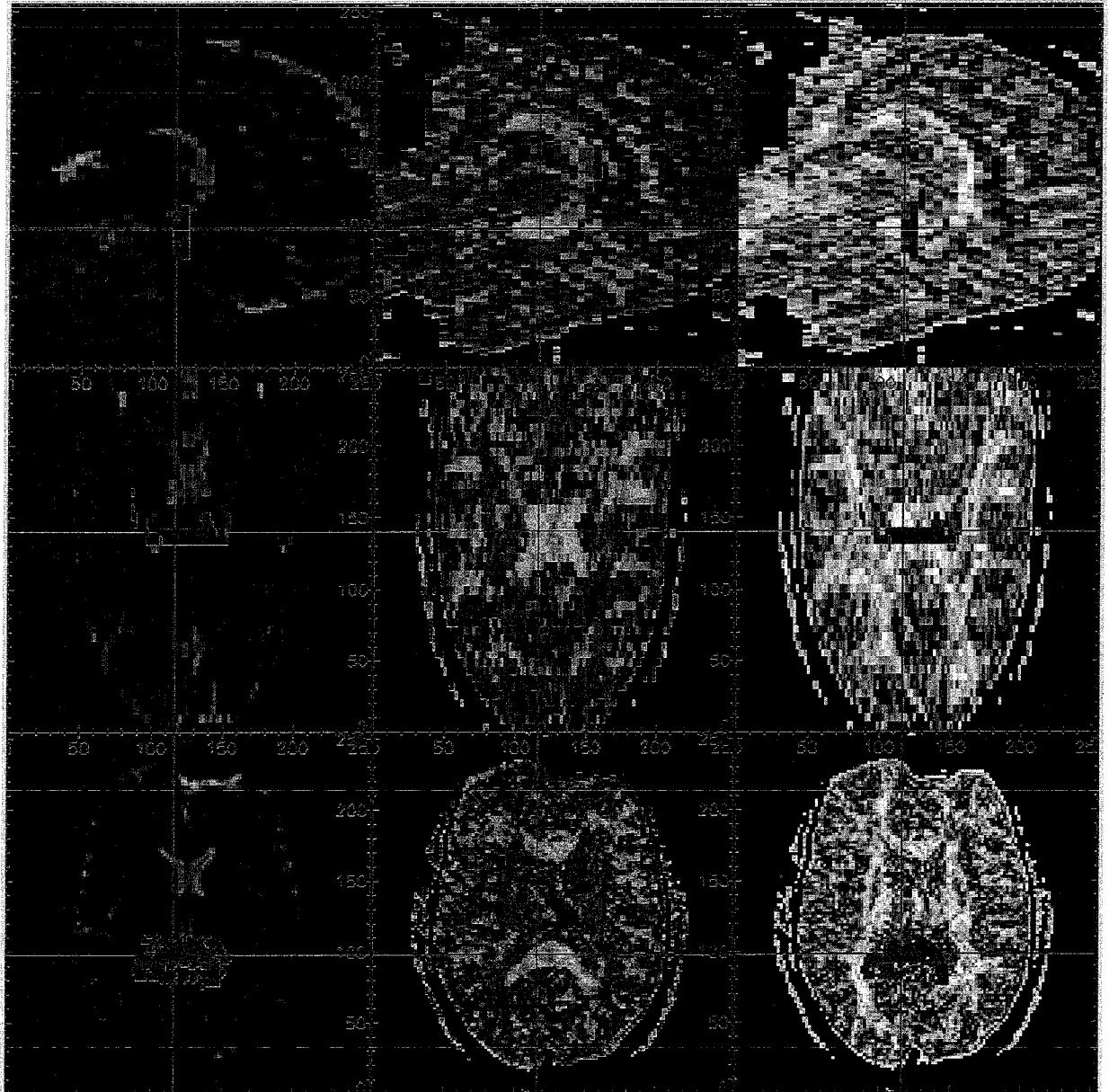


Figure 28 Sample output of the original visualization program

Appendix B. Diffusion Storage Format

Various coordinate systems are used to store the data. An overview of the coordinate systems, and the transformations between them is provided in this appendix.

The gradient directions (measurement directions) can be either predefined (Figure 31) or custom. The coordinate systems in which these gradient directions are stored also depend on the type of scan.

The actual voxels in the dataset are stored in different coordinate systems the the coordinate system used to define the gradient directions. In order to be able to run the fiber tracking algorithm, both the voxels and the tensors need to be in the same coordinate system: Before calculating the tensor, the gradient directions need to be in the same coordinate system as the voxels in the dataset. This transformation is performed when loading the data.

In case the scan type is not 'overplus' nor 'not-overplus', the gradient directions need to be manually entered in the Fiber Tracking tool via a text-file. These manually entered gradient directions must be in the appropriate coordinate system, as defined by Figure 29.

Figure 29 shows all coordinate systems (Yellow boxes) and the transformations between them (Arrows). The blue boxes show the coordinate system of the voxels (pixels) and of the gradient directions of various scan types:

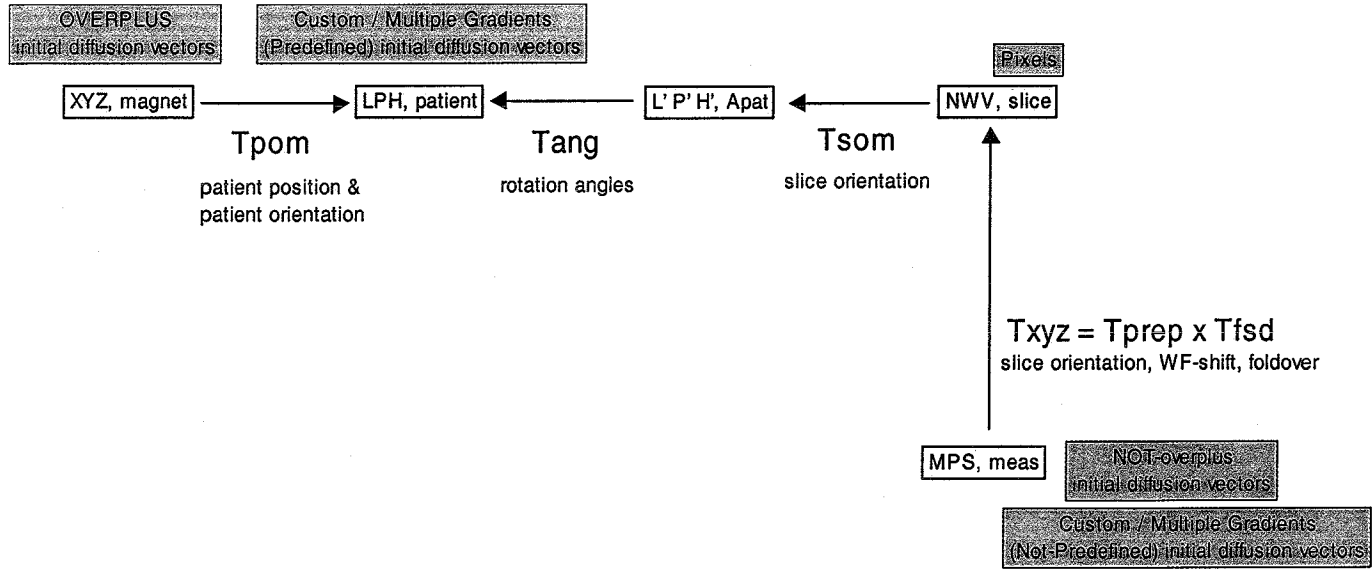


Figure 29 The coordinate systems and their corresponding transformations

The actual voxels of the data are stored in the NWV coordinate systems. The gradient directions however are defined in different coordinate systems:



- Overplus scans define the gradient directions in the XYZ coordinate system.
- Non-overplus scans define the gradient directions in the MPS coordinate system.
- Custom defined gradients define the gradient directions either in the MPS coordinate system in case of manually entered gradients, or in LPH in case of predefined gradients.

Transformations between these coordinate systems are available and depend on parameters such as: Release Version, Patient Position, Rotation angle of the scan, Foldover, WF-Shift direction and Slice Orientation.

Figure 30 shows the transformation matrices of the transformations between various coordinate systems.

		R 8.1.1 or before			R 8.1.1 or after																										
		Tprep	Tsd = 1	Tsd	Tprep	Tsd	Tsd, m-flip #																								
Tpom = Tpo x Tpp																															
Tpo, supine	Tpp, feefirst	Ttl	Tsom, sag	Tprep, par *	Tprep, par *	Tsd, m-flip #																									
1 0 0	0 -1 0	1 0 0	0 0 -1	1 0 0	1 0 0	-1 0 0																									
0 1 0	-1 0 0	0 cos(t) -sin(t)	0 -1 0	0 1 0	0 1 0	0 1 0																									
0 0 1	0 0 1	0 sin(t) cos(t)	1 0 0	0 0 1	0 0 1	0 0 1																									
Tpo, prone	Tpp, headfirst	Tap	Tsom, cor	Tprep, par *	Tprep, par *	Tsd, p-flip #																									
-1 0 0	0 1 0	cos(ep) 0 sin(ep)	0 -1 0	0 1 0	0 -1 0	1 0 0																									
0 -1 0	-1 0 0	0 0 0	0 0 1	0 0 1	1 0 0	0 -1 0																									
0 0 1	0 0 -1	-sin(ap) 0 cos(ap)	1 0 0	0 0 1	0 0 1	0 -1 0																									
Tpo, rightdecubitus		Tth	Tsom, tra			Tsd, s-flip #																									
0 -1 0		cos(fh) -sin(fh)	0 -1 0			1 0 0																									
1 0 0		sin(fh) cos(fh)	-1 0 0			0 1 0																									
0 0 1		0 0 1	0 0 1			0 0 -1																									
Tpo, leftdecubitus																															
0 1 0																															
-1 0 0																															
0 0 1																															
		* par stands for parallel: when the P(eparation) direction of PMS is parallel to the W(est) direction in the NWS system, 'par' (from perpendicular) applies when the above is not true.																													
		To choose: see table below!																													
		<table border="1"> <tr> <td colspan="4">foldover direction</td> </tr> <tr> <td>FH</td> <td>AP</td> <td>LR</td> <td></td> </tr> <tr> <td>per</td> <td>par</td> <td>par</td> <td></td> </tr> <tr> <td>COR</td> <td>per</td> <td>par</td> <td></td> </tr> <tr> <td>SAG</td> <td>per</td> <td>par</td> <td></td> </tr> </table>		foldover direction				FH	AP	LR		per	par	par		COR	per	par		SAG	per	par									
foldover direction																															
FH	AP	LR																													
per	par	par																													
COR	per	par																													
SAG	per	par																													
		<table border="1"> <tr> <td colspan="4">initial direction</td> </tr> <tr> <td>TRA</td> <td>per</td> <td>par</td> <td></td> </tr> <tr> <td>COR</td> <td>per</td> <td>par</td> <td></td> </tr> <tr> <td>SAG</td> <td>per</td> <td>par</td> <td></td> </tr> </table>		initial direction				TRA	per	par		COR	per	par		SAG	per	par													
initial direction																															
TRA	per	par																													
COR	per	par																													
SAG	per	par																													
		<p># In the table on the right the choices for the m-, p- or s-flip are given: it depends on the initial slice orientation, foldover direction and the applied fat-shift direction.</p> <p>The yellow areas indicate the possibilities for SE-EPI diffusion weighted imaging.</p>																													
		<table border="1"> <tr> <td colspan="4">CORONAL</td> </tr> <tr> <td>FH</td> <td>L</td> <td>F</td> <td>A</td> </tr> <tr> <td>RL</td> <td>p</td> <td>p</td> <td>m</td> </tr> <tr> <td></td> <td>m</td> <td>p</td> <td>m</td> </tr> <tr> <td></td> <td>p</td> <td>p</td> <td>m</td> </tr> <tr> <td></td> <td>m</td> <td>p</td> <td>s</td> </tr> </table>		CORONAL				FH	L	F	A	RL	p	p	m		m	p	m		p	p	m		m	p	s				
CORONAL																															
FH	L	F	A																												
RL	p	p	m																												
	m	p	m																												
	p	p	m																												
	m	p	s																												
		<table border="1"> <tr> <td colspan="4">SAGITAL</td> </tr> <tr> <td>FH</td> <td>L</td> <td>F</td> <td>A</td> </tr> <tr> <td>AP</td> <td>s</td> <td>m</td> <td>p</td> </tr> <tr> <td></td> <td>m</td> <td>p</td> <td>m</td> </tr> <tr> <td></td> <td>s</td> <td>m</td> <td>p</td> </tr> <tr> <td></td> <td>m</td> <td>p</td> <td>m</td> </tr> </table>		SAGITAL				FH	L	F	A	AP	s	m	p		m	p	m		s	m	p		m	p	m				
SAGITAL																															
FH	L	F	A																												
AP	s	m	p																												
	m	p	m																												
	s	m	p																												
	m	p	m																												
		<table border="1"> <tr> <td colspan="4">TRANSVERSE</td> </tr> <tr> <td>RL</td> <td>p</td> <td>m</td> <td>s</td> </tr> <tr> <td>AP</td> <td>m</td> <td>p</td> <td>m</td> </tr> <tr> <td></td> <td>p</td> <td>p</td> <td>m</td> </tr> <tr> <td></td> <td>m</td> <td>s</td> <td>m</td> </tr> <tr> <td></td> <td>p</td> <td>m</td> <td>p</td> </tr> </table>		TRANSVERSE				RL	p	m	s	AP	m	p	m		p	p	m		m	s	m		p	m	p				
TRANSVERSE																															
RL	p	m	s																												
AP	m	p	m																												
	p	p	m																												
	m	s	m																												
	p	m	p																												

Figure 30 The transformations between coordinate systems

INITIAL DIFFUSION DIRECTIONS	
OVERPLUS	
User Int. relative to XYZ (magnet coordinates)	
M	1.0 -0.5 1.0
P	-0.5 1.0 1.0
S	1.0 1.0 -0.5
MP	1.5 -1.5 0.0
PS	-1.5 0.0 1.5
MS	0.0 -1.5 1.5
NOT-overplus relative to MPS	
M	1 0 0
P	0 1 0
S	0 0 -1
MP	1 -1 0
PS	0 1 1
MS	1 0 1
RELEASE 8.13 or before 5X6	
PLUS OPTION FOR SIGN CHANGE OF S VECTOR AND CONSECUTIVE DERIVED VECTORS (NOT often but possible)	
	
RELEASE 9.1 (or later) 5X7	
NOT-overplus relative to MPS	
M	1 0 0
P	0 1 1
S	0 0 1
MP	1 -1 0
PS	0 1 1
MS	1 0 -1
	

The images are read from file in the NWV coordinate system.

The images are put into the database in the following order:
P / M / S / MP / PS / MS
MP represents "M - P" etc.

Figure 31 The gradient directions for predefined scans

Appendix C. Dealing with crossing Fibers

In case there is more than one fiber present in a single voxel, a 3x3 symmetric tensor is inadequate to describe the anatomical situation in that voxel. Voxels that are troubled with crossing fibers or other sources of interference will exhibit the following properties:

- 1) Anisotropy in the voxel will be low.
- 2) At least both the major and the medium eigenvalues will be large.

An 'Ideal' crossing fiber is shown in 2D in Figure 32.

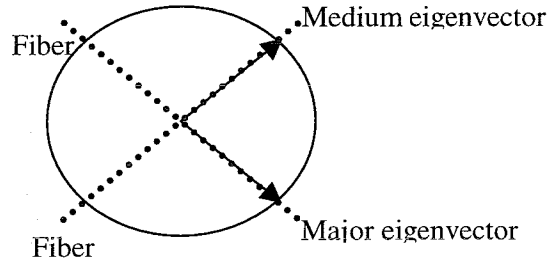


Figure 32 Two crossing fibers result in low anisotropy and high eigenvalues

In Figure 33, an 'ideal' fiber crossing is illustrated.

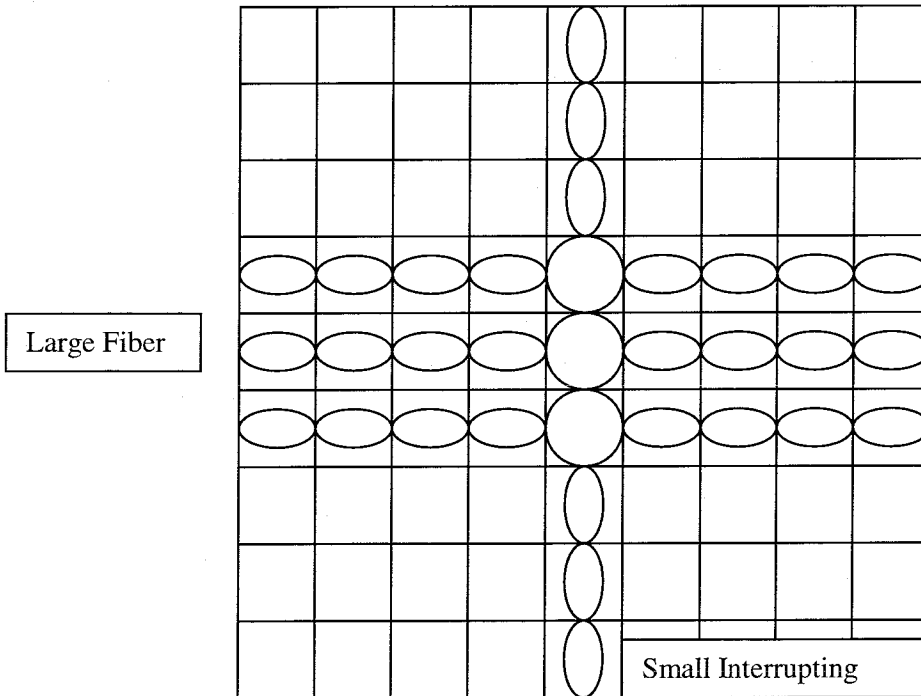


Figure 33 Crossing fibers result in localized interruptions of the major eigenvector tracking.

In case the program is tracking the large fiber in Figure 33, fiber tracking will normally terminate when it reaches the voxels that have low anisotropy. However, with a modification to the fiber tracking algorithm, the algorithm will be able to cope with small interruptions. The normal fiber tracking algorithm can be summarized as (Euler's method):

- 1) Start a starting position x
- 2) $y = x + \text{step_size} * \text{major_eigenvector}_x$
- 3) if $\text{angle}(x, y) < \text{angle_threshold}$ or $\text{anisotropy}(y) < \text{anisotropy_threshold}$ then $x = y$ and goto 2).
- 4) Track fiber in the other direction from x

We change the fiber tracking algorithm to handle small interruptions. These extra parameters are needed:

Max_anisotropy_of_crossing_fibers: The maximum anisotropy allowed in order to consider the voxel as a 'Crossing Fiber' voxel.

Min_avg_eigenvalue: The minimum average of the major and medium eigenvalues to consider the voxel as a 'Crossing Fiber' voxel.

Max_crossing_fiber_voxels: The maximum number of 'Crossing Fiber' voxels allowed while continuing tracking.

The new algorithm can be summarized as:

- 1) Start a starting position x
 $\text{Last_valid_eigenvector} = \text{invalid}$
 - 2) If $\text{Is_Crossing_Fiber_Voxel}(x)$ and
 $\text{Last_valid_eigenvector} \neq \text{invalid}$ and
 $\text{nr_of_crossing_fiber_voxels} < \text{Max_crossing_fiber_voxels}$ then
 $y = x + \text{step_size} * \text{Last_valid_vector}$
 $\text{nr_of_crossing_fiber_voxels} = \text{nr_of_crossing_fiber_voxels} + 1$
 else
 $y = x + \text{step_size} * \text{major_eigenvector}_x$
 $\text{nr_of_crossing_fiber_voxels} = 0$
 $\text{Last_valid_eigenvector} = \text{major_eigenvector}_x$
 endif
 - 3) if $\text{angle}(x, y) < \text{angle_threshold}$ or $\text{anisotropy}(y) < \text{anisotropy_threshold}$ then $x = y$ and goto 2).
 - 4) Track fiber in the other direction from x
- Where $\text{Is_Crossing_Fiber_Voxel}(x) = \text{anisotropy}(x) < \text{Max_anisotropy_of_crossing_fibers}$ and
 $(\text{major_eigenvalue}(x) + \text{medium_eigenvalue}(x))/2 > \text{Min_avg_eigenvalue}$

Both algorithms are demonstrated in Figure 34.

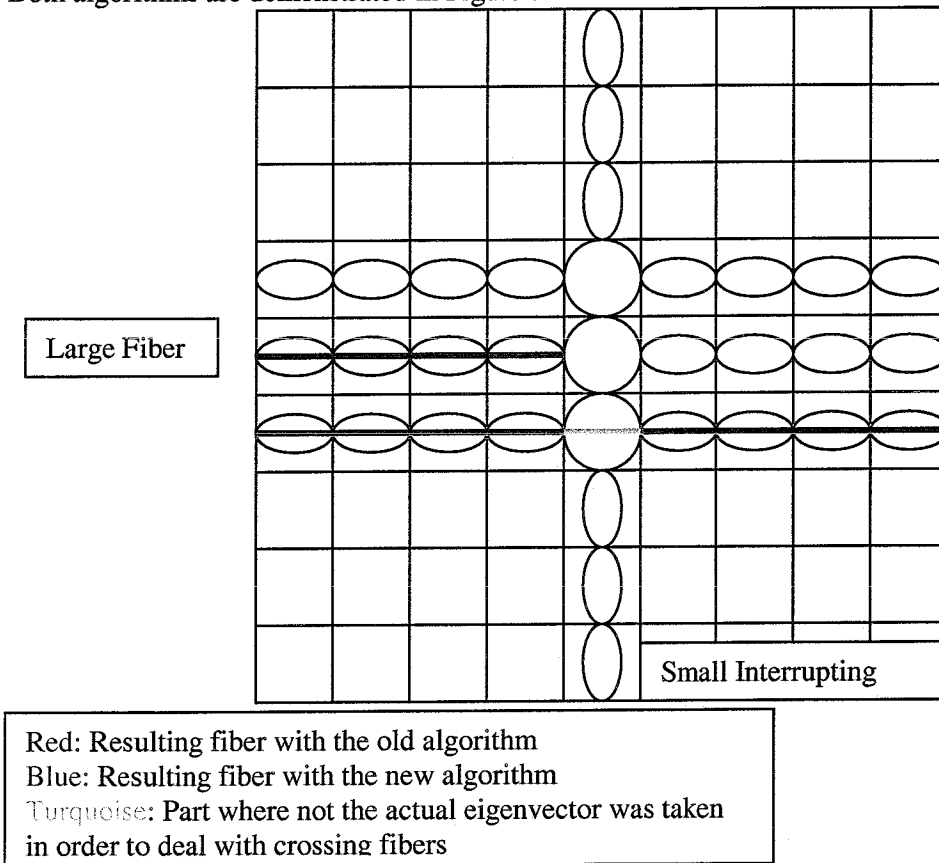


Figure 34 New algorithm that handles crossing fibers

The new algorithm suffers from these drawbacks:

- The new algorithm will only work in case the fiber does not bend in 'Crossing Fiber' areas
 - The two fiber parts on each side of a 'Crossing Fiber' area may not be anatomically connected
- The first drawback is limited. Fiber tracking is simply not possible with that available data with either the old or the new algorithm.

By minimizing the Max_crossing_fiber_voxels parameter, the risk of the second drawback can be controlled. As a result only small interruptions can be dealt with. In large areas of crossing fibers this method is not suitable.