dMRI Preprocessing: from raw data to coregistered tractographies

Thien Bao Nguyen, Eleftherios Garyfallidis, Emanuele Olivetti

1 Abstract

dMRI-Diffusion Magnetic Resonance Imaging is evolving into a potent tool in the examination of the central nervous system. Actually, it is a technique that produces in vivo images of biological tissues by exploiting the constrained diffusion properties of water molecules. In neuroscience, dMRI is used to the extraction of neuronal fibers from brain DMRI and it offers a good look into brain micro-structure at a scale that is not easily accessible with other modalities, in some cases improving the detection and characterization of white matter abnormalities. Because of that, dMRI can help understand brain connectivity deeply. This has lead to numerous beneficial in both diagnosis and clinical applications. However, optimal utilization of the widerange of data provided by many directional diffusion dMRI measurements requires careful attention to acquisition and preprocessing. This article will review the preprocessing of dMRI data inlcuding these main steps: reconstruction, tracking and coregistration. The outcome after these steps is the tractography of the whole brain from discrete measured diffusion tensor MRI data. The tractography is also registered in world space which makes it easy to view in any visulization tool. The implementation is done with the help from Dipy (Diffusion Imaging in Python) package, and FSL is used for visualization.

2 Introduction

Diffusion imaging is a method for measuring the displacement distribution of water molecules in vivo. From the displacement distribution, we can infer the fibre orientation or orientations in each imaging volume element (or voxel). More recently, several groups have proposed tractography methods and have reported success in following fiber tracts. However, there are still some problems with the dataset for doing these things. First, the resolution and quality of diffusion images in vivo was not adequate for this demanding application. Second, the macroscopic fiber-tract direction field is obtained from measured dMRI data that is discrete, coarsely sampled, and noisy. It is difficult to construct fluid streamlines accurately from discrete, noisy, velocity field data.A pre-processing stage which is capable of generating a continuous, smooth representation and highly standard of the measured dMRI data first has to be done in order to ensure the reliability and robustness of dMRI fiber tractography. The aims of this article are to 1) propose and describe a process to perform tractography from discrete measured diffusion tensor MRI data; 2) present a general framework for testing this process in Python language; 3) demonstrate this process in finding fiber tracts in the brain using the dataset of Cambridge university and 4) visulize the tractography in FSL for dotor can see and manulate on it 5)describe potential works in future. This article is organizied as follow:

1. From raw data to Nifti format.
2. Reconstruction.
3. Tracking.
4. Coregistration.
5. Future works.
6. Datasets.

3 From raw data (DICOM format) to NIfTI format

Digital Imaging and Communications in Medicine (DICOM) is the most common standard for receiving scans from a hospital. The DICOM standard was created by the National Electrical Manufacturers Association

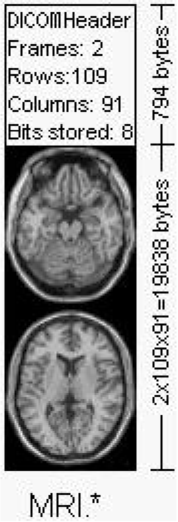

Figure 1: An example of DICOM image file

(NEMA) to aid the distribution and viewing of medical images, such as CT scans, MRIs, and ultrasound. A single DICOM file contains both a header (which stores information about the patient's name, the type of scan, image dimensions, etc), as well as all of the image data (which can contain information in three dimensions).

In figure 1, there is an example of the hypothetical DICOM image file. In this one, the first 794 bytes are used for a DICOM format header, which describes the image dimensions and retains other text information about the scan. The size of this header varies depending on how much header information is stored. Here, the header defines an image which has the dimensions 109x91x2 voxels, with a data resolution of 1 byte per voxel - so the total image size will be 19838. The image data follows the header information (the header and the image data are stored in the same file). More information about DICOM format can be found on the official webpage of DICOM [1]

Most of the MRI scanner produces T2-weighted DICOM data (.dcm). But DICOM data is quite complex leading to difficultly understanding. DICOM data needs to be converted in the format of NIfTI (Neuroimaging Informatics Technology Initiative). NIfTI is a modern incarnation of the

[1] http://medical.nema.org/

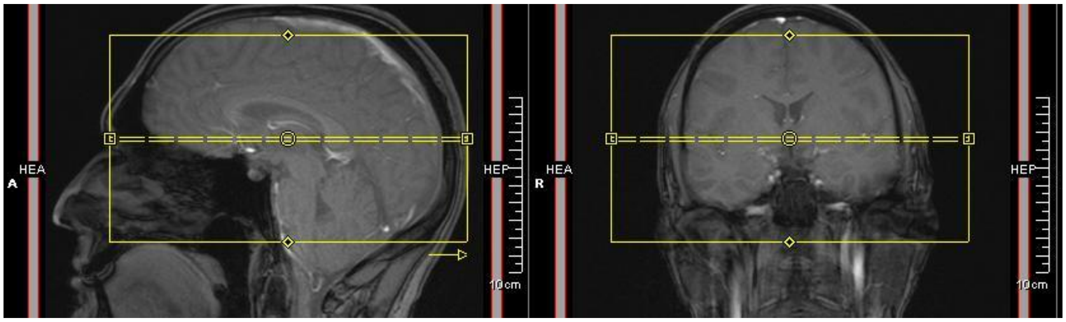

Figure 2: Image acquired orthogonal to scanner bore

Analyze format, but includes important information like the orientation of the image. It was for scientific analysis of brain images [2]. The format is simple, compact and versatile. The images can be stored as a pair of files (hdr/img, compliant with most Analyze format viewers), or a single file (nii). Many tools like FSL, NiBabel, MRIcron ... can also read compressed (nii.gz) images. One nice feature about NIfTI is that the format attempts to keep spatial orientation information. Therefore, NIfTI softwares that can read the spatial information should reduce your chance of making left-right errors.

In this project, NiBabel is used to convert DICOM data to NIfTI. NiBabel is a pure python package [3], and easy to run on any system. Result of this processing is NIfTI file image (.nii). NIFTI file image contains actual dMRI data (4D numpy.array, (I,J,K,W)), called *native image*, and an*affine*[4] transformation (2D, 4×4 numpy.array). The affine stores the information to map the volume i,j,k voxels in *World coordinates*, which is a space in millimiter units. The world coordinate system, or *native space*, is the coordinate system of the scanner.

Beside the NIfTI file image, most of DICOM image conversion tools also generate gradient vector (.bvec) and b-value (.bval) files. These files are very important to reconstruct diffusion properties. Because in diffusion tensor imaging (DTI) we construct tensors by collecting a series of direction-sensitive diffusion images. Therefore, in addition to recording the images, the scanner also saves these directions. A potential concern

[2] `http://nifti.nimh.nih.gov/`

[3] `http://nipy.org/nibabel`

[4] `http://en.wikipedia.org/wiki/Affine_transformation`

is that the scanner manufacturers can choose to either report the vectors with reference to the scanner bore, or with reference to the imaging plane (i.e., imaging grid). This is not a problem if the images are always acquired precisely orthogonal to the scanner bore (figure 2), as the image and scanner have the same frame of reference. However, problems can arise when the image plane is not aligned with the scanner bore (i.e., oblique acquisitions). In this situation, it is important to ensure that these vectors are in the same frame of reference as the image.Moreover, the eigenvectors of the tensor, and consequently tractography programs are sensitive to proper interpretation of the bvecs relative to the imaging plane.

Load NIfTI file image from memory

```
nii_filename = < path_of_nifti_file_name >
img = nib.load(nii_filename)
```

Get data and affine transformation

```
data = img.get_data()
affine = img.get_affine()
```

Get gradient vector (.bvec) and b-value (.bval) from files

```
    bvec_filename = < path_of_bvec_file_name >
    bval_filename = < path_of_bval_file_name >
    bvals = np.loadtxt(bval_filename)
    gradients = np.loadtxt(bvec_filename)
    assert(bvals.size==gradients.shape[0])
```

End of this step, all the necessary information for tracking have been extracted from the DICOM raw data including: actual dMRI data (4D numpy.array, (I,J,K,W)), affine transformation, gradient vector (.bvec) and b-value (.bval). In next stage, we will discuss how to create tractography from the actual dMRI data.

4 Reconstruction

From the actual dMRI data we can create the *tractrography*, the *fractional anisotropy* (FA) volume and the *mean diffusivity* (MD) volume. A tractography is created in two steps: reconstruction and tracking. Reconstruction

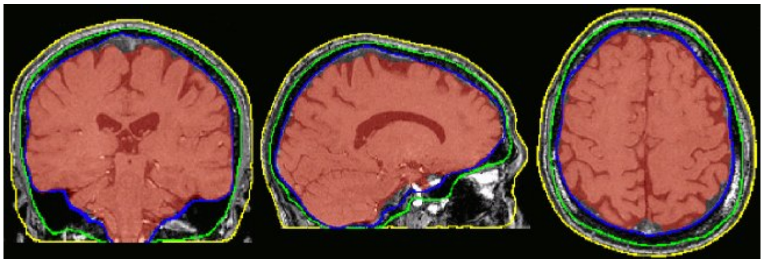

Figure 3: Brain extraction

is about computing the information about the spatial distribution of the diffusion signal within each voxel. While tracking tries to connect many signals to form a tractography based on orientation signal of each voxel. In this paragraph, the main focus is on how to reconstruction of dMRI data.

However, before doing reconstruction, the actual dMRI data in NIfTI image needs to extract brain image only. Because the result of scanner usually contains not only brain but also other things close to brain which can distract the processing of tracking. Brain extraction is the process of removing the skull and the rest of the head from the brain, therefore is necessary to be done before further analysis. Example of brain extraction can be seen in figure 3. The resulting file only contains a representation of the brain's anatomy.

Brain extraction can be done with the FSL program BET (Brain Extraction Tool) [28]. FSL[5] is a comprehensive library of analysis tools for FMRI, MRI and DTI brain imaging data. FSL is written mainly by members of the Analysis Group, FMRIB, Oxford, UK. FSL runs on Apple and PCs (Linux and Windows), and is very easy to install. Most of the tools can be run both from the command line and as GUIs (”point-and-click” graphical user interfaces).

BET [28] takes an image of a head and removes all non-brain parts of the image. It uses a deformable model that evolves to fit the brain's surface by the application of a set of locally adaptive model forces. This method is very fast and requires no preregistration or other pre-processing before being applied. Result of this is a file saved with a brain extension at its end. BET can run from the console (with the command bet). FSL recommend that do not use the BET-GUI because it almost always needs options

[5] `http://www.fmrib.ox.ac.uk/fsl/index.html`

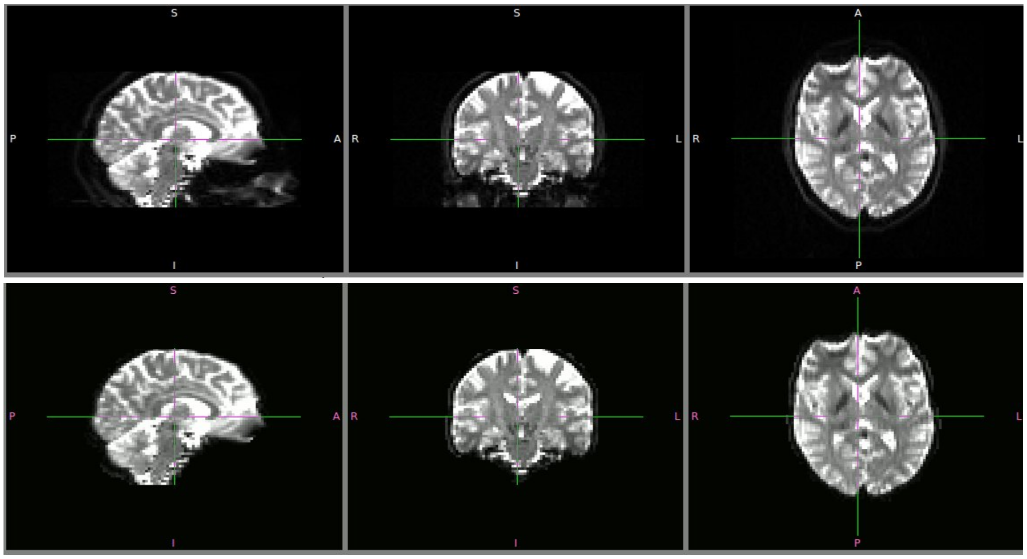

Figure 4: Origional NIfTI (top) and BET (bottom) image file of subject 3 in Cambridge datase

that are not available in the GUI. Also always use FSLView to inspect the output of the bet procedure and to fine-tune the bet operation. Run bet in the console to see the available options and its command line:

```
bet(<ori_NIfTI_image_file>, <brain_extract_file>)
```

The result of BET can be seen in the bottom line of the figure 4. This image is taken from subject 3 of the Cambridge dataset. While origional image date is in the top line of this figure 4.

After brain extraction, we can do reconstruction step. The main purpose of this is to estimate the orientation information from the diffusion signal within each voxel which is adequate for accurate tractography generation. In the last few years, an increasing number of techniques have been proposed to recover the signal directions inside the voxel from diffusion MRI data. The most simple methods is Diffusion Tensor model proposed by Basser et. al [3]. But in many cases this model is not sufficiently, because most of voxels inside brain contain multiple fiber bundles crossings while this model is only working with single tensor. Many other reconstruction methods have been proposed to overcome the limitations of this Diffusion Tensor model. The earliest studies distinguished between

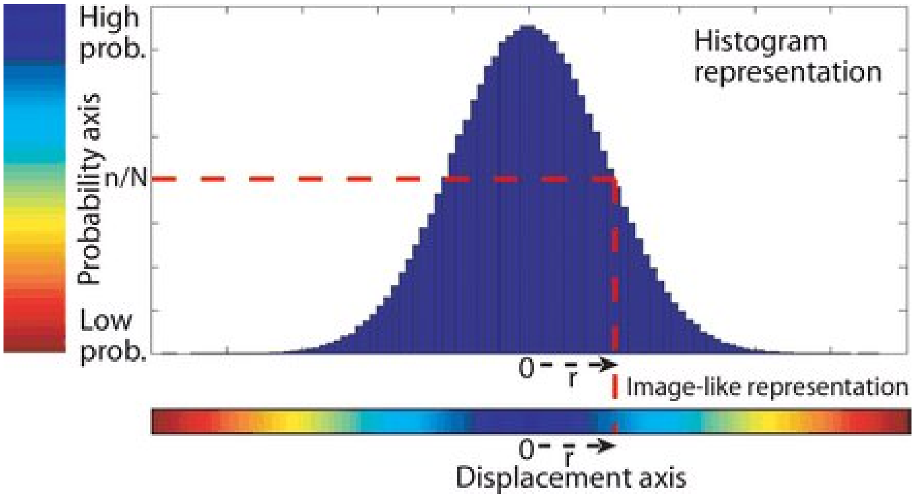

Figure 5: Histogram of displacement distribution

voxels with isotropic, single-fiber anisotropic, and multi-fiber anisotropic complexity and have reported clustered and symmetric regions of increased complexity, supporting genuine effects consistent with anatomical knowledge. Following are some other advanced methods such as Diffusion Spectrum Imaging [7] or Higher Order Tensors [26]. The overview of these model can be found more detail in [15] or [6]. The basic idea for diffusion tensor model is briefly described in the following.

The histogram in figure 5 shows a typical displacement distribution in a one-dimensional model. All the molecules have moved the given distance r. For each displacement distance r (x-axis) there is a corresponding probability n/N (y-axis), which is the proportion of molecules within a voxel that were displaced that distance within a time interval Δ (the duration of the diffusion experiment). The top of the histogram is centered on zero, indicating that most molecules had the same position at $t = 0$ and $t = \Delta$. The proportion of molecules that traveled the given distance r is indicated by the dotted red line. The horizontal color bar, in which blue signifies a high probability and red a low probability of displacement, shows the same Gaussian distribution.

But, a histogram like in figure 6 is adequate for the display of one-dimensional data, because it is not practical for visualizing displacement

[6] `http://radiographics.rsna.org/content/26/suppl_1/S205.full`

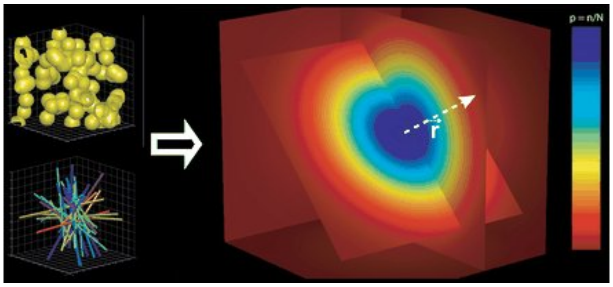

Figure 6: The 3D diffusion probability density function in a voxel which contains intersect orientations

in multiple dimensions. A useful approach is instead to color-code the probability. With such a representation, the one-dimensional problem can be visualized as a colored displacement axis (x-axis) in which blue codes for high and red for low probability (figure 5). According to this rule, we can represent actual three-dimensional (3D) diffusion as a 3D image in which the probability of displacement in three intersecting planes is coded in color (figure 6). The central voxel of the image is the origin, and its value codes for the probability, or the proportion of molecules (n/N) that do not undergo displacement between $t = 0$ and $t = \Delta$. This 3D diagram represents the displacement distribution. On a photograph taken at $t = \Delta$, the dye will have been diluted (diffused), and the relative color density will indicate the proportion of dye molecules displaced a given distance.

Diagram shows the 3D diffusion probability density function in a voxel that contains spherical cells (top left) or randomly oriented tubular structures that intersect, such as axons (bottom left). This 3D displacement distribution, which is roughly bell shaped, results in a symmetric image (center), as there is no preferential direction of diffusion. The distribution is similar to that in unrestricted diffusion but narrower because there are barriers that hinder molecular displacement. The center of the image (origin of the r vector) codes for the proportion of molecules that were not displaced during the diffusion time interval. The color bar (right) shows the spectrum used in color coding to represent probability, from the lowest value, which is indicated by red, to the highest, which is indicated by blue.

The figure 7 shows the diffusion probability density function within a

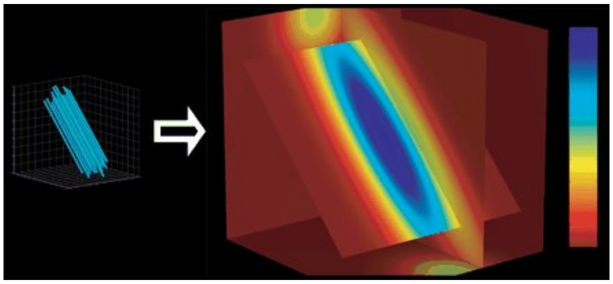

Figure 7: The 3D diffusion probability density function in a voxel in which orientation alignes with the axons

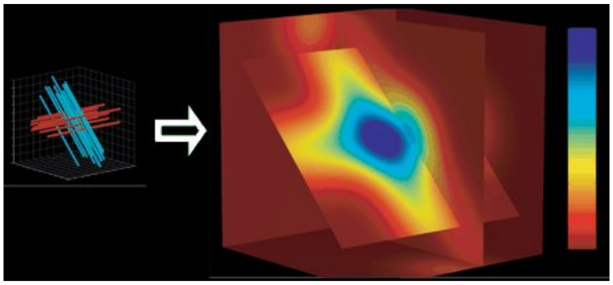

Figure 8: The 3D diffusion probability density function in a voxel with fibers intersecting at an angle of 90

voxel in which all the axons are aligned in the same direction. The displacement distribution is cigar shaped and aligned with the axons.

While the diagram in figure 8 shows the diffusion probability density function within a voxel that contains two populations of fibers intersecting at an angle of 90. The molecular displacement distribution produces a cross shape. Diffusion in a homogeneous medium is well described as having a Gaussian distribution. Depending on the type of molecule, the temperature of the medium, and the time allowed for diffusion, the distribution will be wider or narrower. The spread of the Gaussian distribution is controlled by a single parameter: variance (σ^2). Variance, in turn, depends on two variables, so that $\sigma^2 = 2 \cdot D \cdot \Delta$, where Δ, the diffusion coefficient, characterizes the viscosity of the medium or the ease with which molecules are displaced. The longer the diffusion time interval, the larger the variance, because there is more time in which molecules may be displaced.

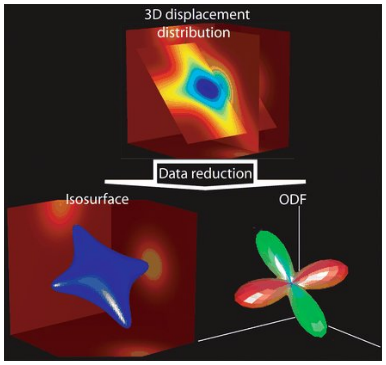

Figure 9: Two approaches for visual representation of 3D diffusion data

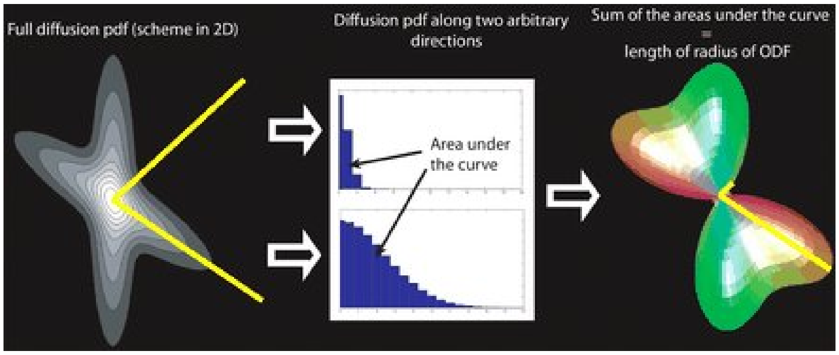
Full diffusion pdf (scheme in 2D)
Diffusion pdf along two arbitrary directions
Sum of the areas under the curve = length of radius of ODF
Area under the curve

Figure 10: How to compute ODF

Typically, at diffusion imaging, there is less interest in obtaining a detailed diffusion profile than in determining the direction of the most rapid diffusion, because the latter parameter likely corresponds to the orientation of axons or other fibrillar structures. One possible approach would be to replace the diffusion probability density function with an isosurface, which is a surface that passes through all points of equal value, or, in other words, where the value of (r) is a constant (figure 9). A more commonly used technique that is less sensitive to noise involves the computation of the orientation distribution function from the displacement distribution.

Figure 9 shows two approaches that may be used to simplify the visual representation of 3D diffusion data: replacement of the displacement distribution with an isosurface, and computation of the orientation distribution function (ODF). And figure 10 shows how an orientation distribution

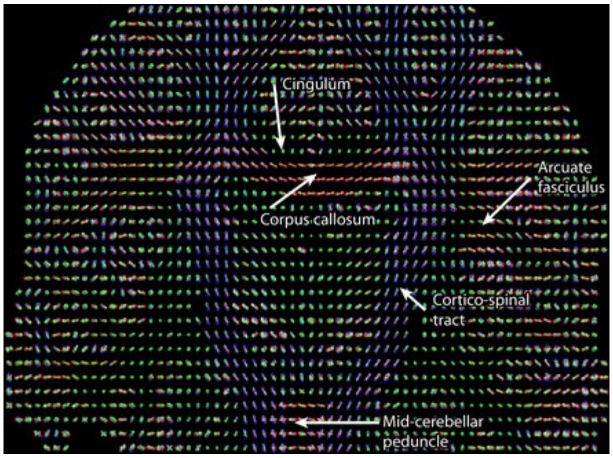

Figure 11: Orientation distribution function map of a coronal brain section

function (ODF) is computed and represented. The left image is a section through a schematized 3D displacement distribution. The value of the orientation distribution function is computed along two axes (yellow lines). While the center one is histogram that represents the displacement distribution along the two axes. The value of the orientation distribution function along those axes equals the area under the curve for each axis. In this example, the two areas under the curve are respectively small and large, indicating that there is much less diffusion in the one direction than in the other. In the right image, the sum of the areas under the curve is represented by a deformed sphere in which the lengths of the two radii (yellow lines) are short and long, corresponding to little diffusion and much diffusion, respectively. To compute the orientation distribution function, the area under the curve is computed for every direction.

An orientation distribution function may be considered a deformed sphere whose radius in a given direction is proportional to the sum of values of the diffusion probability density function in that direction. For ease of visualization, we color code the surface according to the diffusion direction ([x,y,z] = [r,b,g], where r = red, b = blue, and g = green). An orientation distribution function or isosurface can be plotted for each individual MR imaging voxel in a section. In the figure 11, for every brain position p, an orientation distribution function is plotted to characterize the local diffusion probability density function. It is easy to identify the corticospinal tract, in which the dominant color is blue, and the corpus

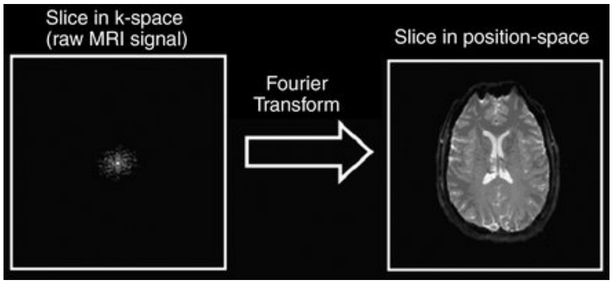

Figure 12: The process to obtain a standard position-encoded MR image

callosum, in which red is predominant. More difficult to see are the cingulum and the arcuate fasciculus, depicted predominantly in green, and the middle cerebellar peduncle, in red. From this, we can easy to tracking tractography. But more detail will be discuss on the next paragraph about tracking.

MR imaging is the result of the application of gradients in different directions and with different intensities at specific moments of the acquisition sequence. The values of the measured signal are organized in a coordinate system known as k-space. Performing the acquisition enables the filling of k-space. To transform the raw MR imaging data from k-space into a position-encoded visual image, a mathematical operation known as a Fourier transform is applied. Figure 12 shows the process with which a standard position-encoded MR image is obtained. First, the MR signal that was generated by the application of phase and frequency encoding gradients is sampled to fill k-space (a coordinate system used to organize the signal measurements). Next, the raw data (k-space images) are subjected to a mathematical operation known as a Fourier transform to reconstruct an image in the standard position space.

The application of a single pulsed gradient spin-echo (SE) sequence produces one diffusion-weighted image. This diffusion gradient can be represented as a 3D vector q whose orientation is in the direction of diffusion and whose length is proportional to the gradient strength. And the coordinates which are defined by all these vector q are called **q-space**. The gradient strength, or more often the diffusion weighting, is sometimes expressed in terms of the **b value**, which is proportional to the product of the square of the gradient strength q and the diffusion time interval ($b \approx q^2\Delta$). One diffusion-weighted image corresponds to one position in q-space or,

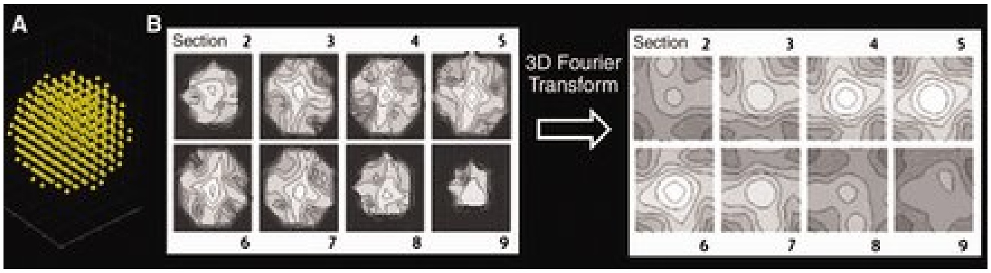

Figure 13: The process to obtain a standard position-encoded MR image

more precisely, that depicts the diffusion-weighted signal intensity in a specific position q for every brain position. Repeated applications of the sequence with gradients that vary in strength and in direction (ie, with variations of q) allow data sampling throughout q-space. Like the data from conventional MR imaging, in which a Fourier transform is applied to the data in k-space, the q-space data are subjected to a Fourier transform in every brain position. The result is a displacement distribution in each brain position (ie, voxel). In other words, a single application of the pulsed gradient SE sequence produces one brain image with a given diffusion weighting. Multiple repetitions of the sequence, each with a different diffusion weighting, are necessary to sample the entirety of q-space; the result is hundreds of brain images, each of which reflects the particular diffusion weighting used. One must then imagine that the data are reorganized so that in every brain position there is a q-space signal sample that consists of hundreds of values and that in every brain position a Fourier transform relates the raw q-space data to the diffusion probability density function

The diagram in figure 13 shows the process with which a 3D diffusion probability density function is obtained for one voxel (one brain position). In A, a 3D grid that represents q-space, each yellow dot corresponds to an MR signal sampling point. The signal is sampled at each point by varying the direction and strength of the diffusion gradient (q vector) of the pulsed gradient SE sequence. With a single application of the pulsed gradient SE sequence, one point in q-space is sampled for each brain position simultaneously, and the result is one diffusion-weighted image. In B, the left panel shows sections through the MR signal sampled in q-space for a specific brain position (one voxel), and the right panel shows the diffusion

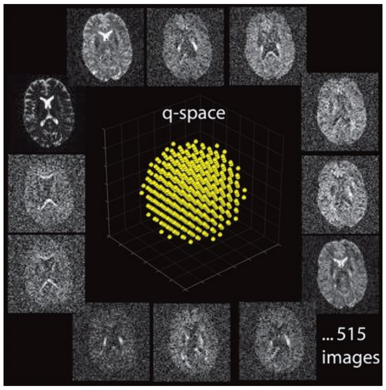

Figure 14: Q-space example

probability density function in the same voxel after a 3D Fourier transform of the MR signal in q-space is performed. The cross-shaped appearance of the diffusion probability density function is often seen in voxels in the brainstem, where axons of the corticospinal tract cross with axons of the middle cerebellar peduncle.

Series of diffusion-weighted MR brain images are obtained with variations in the direction and strength of the diffusion gradient in the pulsed gradient SE sequence (figure 14). Each image shows the signal sampled at one point in q-space (one yellow dot). Every sampling point in q-space corresponds to a specific direction and strength of the diffusion gradient.

To describe the parameters applied in sampling q-space, the term b value is often used. The b value is proportional to the product of the diffusion time interval and the square of the strength of the diffusion gradient. All diffusion images should be compared with a reference image that is not diffusion weighted (a standard SE image)in other words, one for which the strength of the diffusion gradient is zero (q = 0 and b = 0).

When we sample at least six points in q-space with $q \neq 0$ (diffusion-weighted images) and one point with q = 0 (reference image), we solve a set of six equations and the result is a diffusion tensor that is proportional to the Gaussian covariance matrix [19]. This diffusion tensor is a 3×3 matrix that fully characterizes diffusion in 3D space, assuming that the displacement distribution is Gaussian. The diffusion tensor is usually represented by an ellipsoid or an orientation distribution function. Figure 15 describes a series of diffusion-weighted images obtained for diffusion ten-

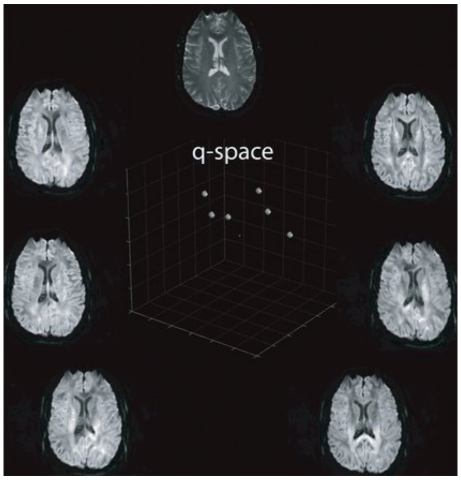

Figure 15: Series of diffusion-weighted images obtained for diffusion tensor imaging

sor imaging, in which q-space is sampled in at least six different directions and in which a nondiffusion-weighted reference image is obtained. The direction but not the strength of the diffusion gradient is changed for each sampling.

There are two ways to describe diffusion tensor. In figure 16, the diffusion tensor is shown as an ellipsoid (an isosurface) with its principal axes along the eigenvectors $(\lambda_1, \lambda_2, \lambda_3)$. While in B, the diffusion tensor is shown as an orientation distribution function.

The direction of the diffusion maximum is called the principal direc-

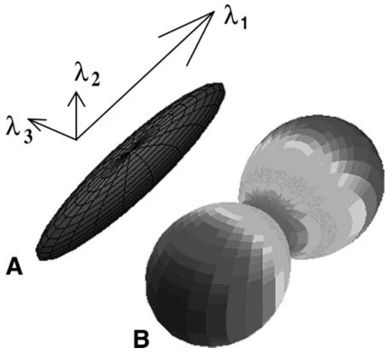

Figure 16: Two ways to present diffusion tensor

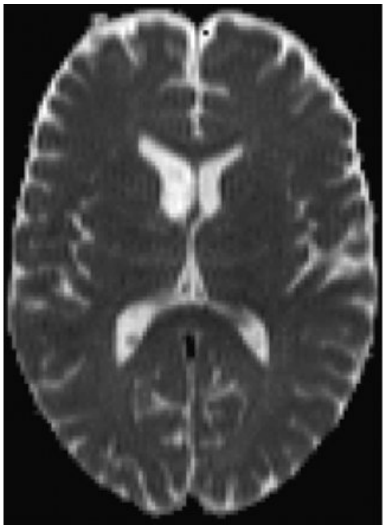

Figure 17: Mean diffusion tensor image

tion of diffusion and can be directly obtained by computing eigenvectors and eigenvalues of the tensor. Eigenvectors are orthogonal to one another, and, with eigenvalues, describe the properties of the tensor. Eigenvalues are ordered as $\lambda_1 \geq \lambda_2 \geq \lambda_3$, and each corresponds to one eigenvector. The eigenvector that corresponds to the largest eigenvalue (λ_1) is the principal direction of diffusion. If the eigenvalues are significantly different from each other, diffusion is said to be anisotropic (figure 16). If λ_1 is much larger than the second eigenvalue, λ_2, the diffusion is cigar shaped (figure 16). If λ_1 and λ_2 are similar but are much larger than λ_3, the diffusion is said to be planar or disc shaped. When all the eigenvalues are approximately equivalent, diffusion is isotropic and may be represented as a sphere [29].

The mathematical properties of the diffusion tensor make it possible to extract several useful scalar measures from diffusion tensor images. The mean diffusion, also known as the trace, is computed by averaging the diagonal elements of the matrix [19](Equation: 1). The result is an image like that in figure 17. Frequently, mean diffusivity (MD) is used instead of the trace defined as

$$MD = \frac{trace(D)}{3} = \frac{\lambda_1 + \lambda_2 + \lambda_3}{3} \tag{1}$$

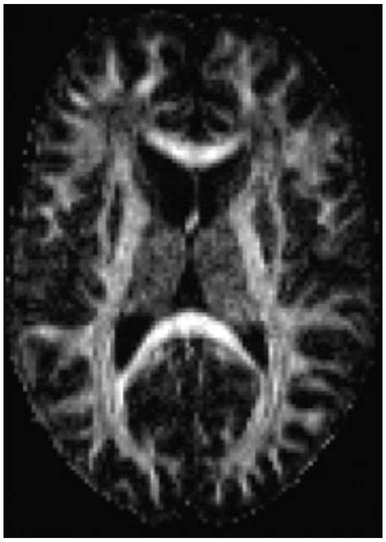

Figure 18: The fractional anisotrophy image

The relationship between the eigenvalues reflects the characteristics of diffusion. To describe the shape of diffusion with a scalar value, fractional anisotropy is most often used [19]. However, other measures, such as those described by Westin et al [29], are available. Fractional anisotropy is computed by comparing each eigenvalue with the mean of all the eigenvalues ($\langle\lambda\rangle$), as in the following equation:

$$FA = \frac{1}{\sqrt{2}} \frac{\sqrt{(\lambda_1 - \lambda_2)^2 + (\lambda_2 - \lambda_3)^2 + (\lambda_3 - \lambda_1)^2}}{\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}} \tag{2}$$

where FA is the fractional anisotropy. The fractional anisotropy of a section of diffusion tensors can be seen in figure 18. This image shows the fractional anisotropy, which is computed from the eigenvalues of the diffusion tensor. If FA is equal to 1 that means very anisotropic (infinitely prolonged ellipsoid or a stick) and if FA is equal to 0 that means completely isotropic (sphere). Whenever we are using FA volumes we are implicitly assuming that the propagator of the spin displacements in every voxel has a 3D Gaussian distribution. This assumption is used in most of the diffusion related literature where Diffusion Tensor Imaging is the prevailing term. Unfortunately, in reality things are much more complicated; inside our brain the axons are semi-permeable (restriction), the water molecules

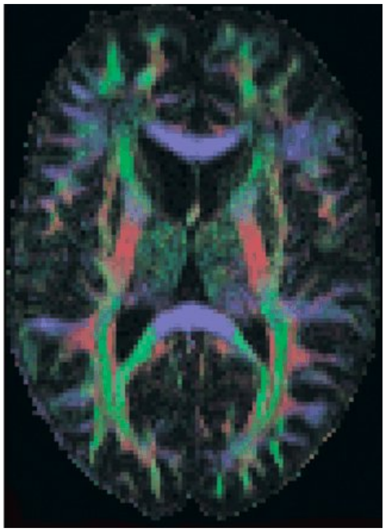

Figure 19: The principal direction of diffusion

interact with many different elements in the complex intra fibre fluid, the fibres might cross, kiss, divert or bend inside a voxel or between voxels. Therefore, assuming a Gaussian distribution is a nontrivial approximation. However, FA is still prevalent as it is easy to calculate and it gives similar values across different acquisitions.

The ellipsoid or the orientation distribution function is the most accurate method for visualizing the diffusion tensor data, but sometimes it is difficult to represent it over an imaging section on a display monitor [3]. Color coding of the diffusion data according to the principal direction of diffusion may be a more practical way of visualizing the data [19]. In the color coding system, red corresponds to diffusion along the inferior-superior axis (x-axis); blue, to diffusion along the transverse axis (y-axis); and green, to diffusion along the anterior-posterior axis (z-axis). The intensity of the color is proportional to the fractional anisotropy. An example of this color coding scheme is shown in figure 19. In this image, the color-coded part shows the orientation of the principal direction of diffusion, with red, blue, and green representing diffusion along x-, y-, and z-axes, respectively. The color intensity is proportional to the fractional anisotropy.

In general, diffusion tensor imaging (**DTI**) and q-ball imaging (**QBall**) are hypothesis-based simplifications that are used to shorten image acquisition time and reduce hardware requirements. They can be used to obtain maps of the orientation distribution functions. Care must be taken when interpreting diffusion tensor imaging data and q-ball imaging data, as there may be brain areas in which the underlying hypotheses are not applicable. In this experiment, we used DTI method to find orientation information of signal within each voxel. The implement of DTI has already been done in Dipy (Diffusion Imaging in Python)[7]. Actually, in Dipy, authors develop DTI algorithm basing on the work of Yeh. et. al [30] who extends the current derivations for diffusion signal reconstruction and provides both scalar and vector metrics that facilitate the understanding of the underlying signal and its use for fast and accurate tractography. The key issue is the reentrance of the spin density as an important part of the diffusion voxel reconstruction. The spin density together with the diffusion propagator give rise to the spin propagator. From the spin propagator both old metrics like FA (equation 2) and MD (equation 1) can be calculated non-parametricly. Following will describe some basic command lines in Dipy for calculating tensor directionality.

- **First, import the necessary modules**

  numpy is for numerical computation

  ```
  import numpy as np
  ```

  nibabel is for data formats

  ```
  import nibabel as nib
  ```

  dipy.reconst is for the reconstruction algorithms which we use to create directionality models for a voxel from the raw data

  ```
  import dipy.reconst.dti as dti
  ```

- **Accessing the necessary datasets**(more detail can be found in section 3)

  Load the nifti file found at path fimg as an NIfTI image format

[7] http://nipy.sourceforge.net/dipy/index.html

```
img = nib.load(nii_filename)
```

Read the datasets from the NIfTI image format

```
data = img.get_data()
```

The raw diffusion weighted MR data is 4-dimensional, the last one is number of gradient directions while the first triple is the dimension of 3d volume.

Read the affine matrix

```
affine = img.get_affine()
```

this gives the mapping between volume indices (voxel coordinates) and world coordinates.

Read the b-values

```
bvals=np.loadtxt(fbvals)
```

this gives a function of the strength, duration, temporal spacing and timing parameters of the specific paradigm used in the scanner, one per gradient direction.

Read the b-vectors

```
gradients=np.loadtxt(fbvecs).T
```

this is the unitary direction of the gradient.

Note that *fbvals* and *fbvecs* are results from converting DICOM to NIfTI format.

- **Calculating models and parameters of directionality**

  We are now set up with all the data and parameters to start calculating directional models for voxels and their associated parameters-FA (fractional anisotropy).

  Calculate the Single Tensor Model (STM)

```
tensor=dti.Tensor(data,bvals,gradients,thresh=50)
```

Calculate Fractional Anisotropy (FA) from STM

```
FA = tensors.fa()
```

FA is a 3-d array with one value per voxel. Dimension of FA is equal to the dimension of the data volume.

In this section, an overview of DTI - diffusion tensor image - one of the most popular and simple reconstruction method is presented. After that, we went to Dipy (diffusion imaging in python) command lines to create tensors and fractional anisotrophy which are necessary for tractography. The next part will discuss about theory of tracking methods and how to do it in python with help from Dipy.

5 Tracking

In the previous part 4, information about orientation of fibres at every voxel has been gained. Based on this information, we can connect these directions to reconstruct complete tracks. This processing is known as tracking, which connect voxels in order to create *tracks*, or *streamlines*, using the spatial information computed during the reconstruction step. An example about creating track from orientation of signal fibres within voxels is presented in figure 20

Brain fiber tractography is a rendering method for improving the depiction of data from diffusion imaging of the brain. Tractography is one of the most powerful tools developed to aid image interpretation. The primary purpose of tractography is to clarify the orientational architecture of tissues by integrating pathways of maximum diffusion coherence. Fibers are grown across the brain by following from voxel to voxel the direction of the diffusion maximum. The fibers depicted with tractography are often considered to represent individual axons or nerve fibers, but they are more correctly viewed in physical terms as lines of fast diffusion that follow the local diffusion maxima and that only generally reflect the axonal architecture. This distinction is useful because, for a given imaging resolution and signal-to-noise ratio, lines of maximum diffusion coherence (ie, the computer-generated fibers) may differ from the axonal architecture in some brains. Tractography adds information and interest to the MR imaging depiction of the human neuronal anatomy.

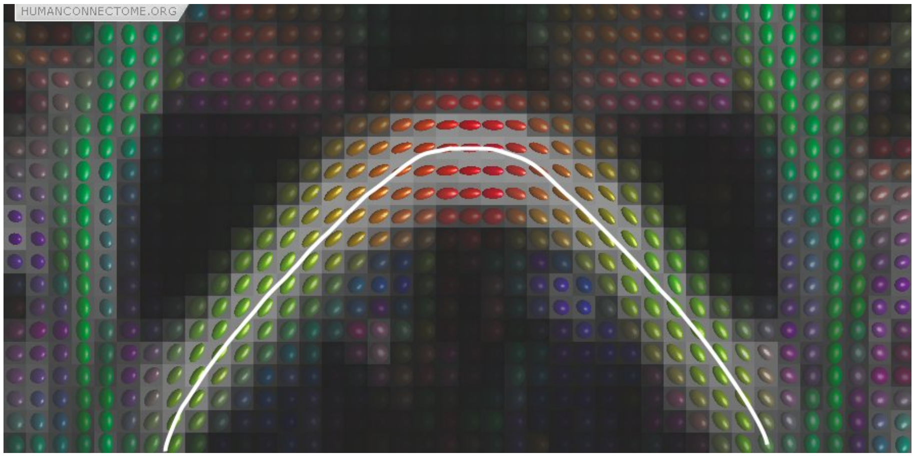

Figure 20: Tracking from tensor direction information

The connectivity maps obtained with tractography vary according to the diffusion imaging modality used to obtain the diffusion data. For example, diffusion tensor imaging provides a Gaussian approximation of the actual displacement distribution, and since the representation of that distribution is restricted to variations of an ellipsoid, this method creates various biases in the tractography result. In contrast, diffusion spectrum imaging with tractography overcomes many of those biases and allows more realistic mapping of connectivity.

The application of fiber tractography to data such as those obtained with diffusion spectrum imaging or q-ball imaging results in the depiction of a large set of fiber tracts with a more complex geometry [13]. The greater complexity obtained with this method, compared with that from tractography with diffusion tensor MR imaging data, is due to the consideration of numerous intersections between fibers that can be resolved or differentiated.

Most tractography techniques can be grouped in three categories: local, global and simulated. Local approaches propagate a curve from a starting (seed) point using locally greedy criteria, i.e. tracking sequentially through orientation estimates in adjacent voxels. Global approaches identify the best path between two points of interest, according to some optimization criterion, rather than identifying paths arising from a single point [18], [16]. Simulated approaches comprise of algorithms that simulate the diffusion process or solve the diffusion equation to reconstruct white matter tracks [12] [17]. However, solving a partial differential equation increases execution time. Furthermore, it is not always easy with these approaches to obtain a connectivity map across the whole brain volume- and there is usually a large number of parameters to set. Due to the need of creating all tractography for the whole brain, only local techniques are described in this part.

In local techniques, the two best known families of track propagation algorithms are: deterministic and probabilistic [9]. Deterministic fiber tracking from diffusion tensor imaging uses the principal direction of diffusion to integrate trajectories over the image [25] but ignores the fact that fiber orientation is often undetermined in the diffusion tensor imaging data. To overcome this limitation of the data, statistical fiber tracking methods based on consideration of the tensor as a probability distribution of fiber orientation [5] [14], [27], [6].

In general, the deterministic algorithms propagate tracks by making a series of discrete locally optimum decisions. These are fast, simple and easy to interpret. Usually, we depict them using streamlines. A streamline is a curve that is always tangent to the velocity vector of the flow. The disadvantages of deterministic algorithms are that a pathway either exists or not (no uncertainty) and that they do not explore the entire space of possible white matter tracts. In other words, local threshold makes the tractography vulnerable to small noise aberrations. Probabilistic tractography is meant to deal with this problem of noise and propagate tracks even in regions where the tracking is unclear. This is made possible by assuming that uncertainty exists concerning the orientation of the fibre at each point of the track.

An example about the difference between deterministic and probabilistic is shown in the figure 21. In this figure, the yellow line in (i) shows the result of deterministic tractography which is given by a single trajectory and in (ii) the tractography is given by connectivity matrix depicting with red the probability of different pathways throughout the hole slice. Here only 3 possible pathways are depicted for the ease of understanding. Finally, in (iii) an example is given where it shows that probabilistic tractography weights more closer connections. However it can tract further deep than deterministic tractography.

When using deterministic tractography we are using only the local direction information in every pixel therefore in a direction field as this of figure 21(i) we will have to use only one track and this is the diagonal path-

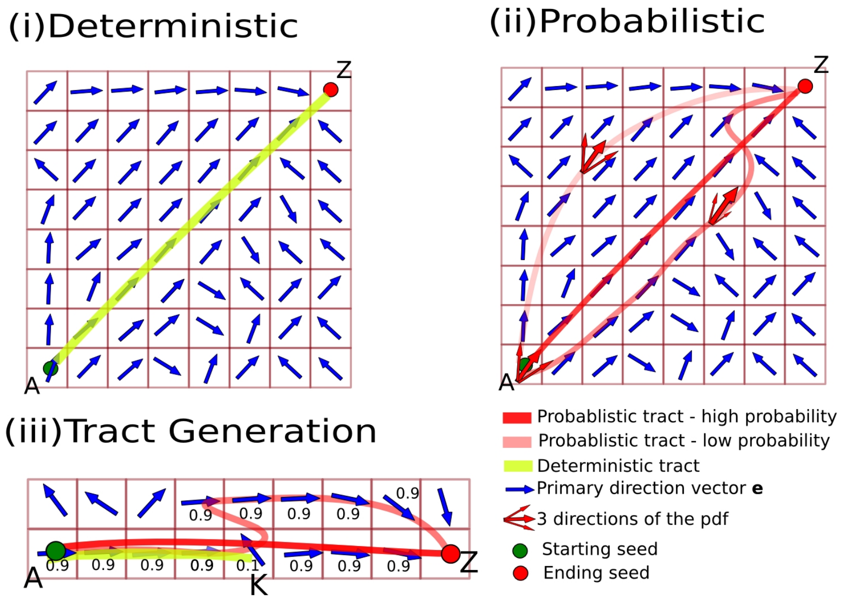

Figure 21: the result of deterministic and probabilistic tractography

way with yellow colour. However there are other possible tracks as well in this diagram e.g rather than taking the diagonal we could go first up north from A and then right. Probabilistic tractography aims to identify all the possible tracks by assigning to each one of them a weight. All the weights of all the tracks together sum to 1. This is possible by generating samples from a probability distribution for every pixel. In this toy example shown in figure 21(ii) the orientation of the blue vectors can be represented by a single parameter, which is a random variable that takes values from a probability distribution function (PDF). Now we have many possible directions to move next but with different probabilities. The weights of all directions again sum to 1. After this explanation we can identify in figure 21 (ii) that the most likely track is again the diagonal (with deep red) but there are other possible tracks (with lighter red) that are less likely. In the same diagram we show with 3 combined red arrows some of the many directions that are possible in each point.

EuDX-Euler integration Delta Crossing There are many deterministic algorithms for tracking, like FACTS [24] and Runge-Kutta [4], PiCO.

In these methods, streamlines are created as trajectories orthograde and retrograde along an initial direction at a specific point (seed) in the 3D volume. Eleftherios proposes a new method, EuDX (Euler integration Delta Crossing). EuDX is a purely deterministic method which is fast, accurate and comprehensive. But most importantly it can have as input model-based or model-free reconstruction algorithms of any known algorithm or complexity. Eu stands for Euler integration, D stands for Delta which is a function that combines together all the different stopping criteria and X stands for crossings. EuDX can deal with any number of crossings as long as your data support them. The purpose of this algorithm is to be faithful to the reconstruction results rather than try to correct or enhance them by introducing regional or global considerations which is the topic of other methods reviewed below. Therefore EuDX serves mainly as a robust method for quickly inspecting different reconstruction results using streamlines. EuDX is noise-friendly i.e. if a voxel is too noisy then EuDX will stop tracking on that voxel. This property is often useful when validating underlying orientation models. Branching is also supported by using trilinear interpolation. This method is an extension of the method used by Yeh et al. [30] and Conturo et. al. [8] with the addition of supporting multiple anisotropic functions and multiple crossings.

$$\mathrm{EuDX}(A, D, A_{thr}) \tag{3}$$

where A, the anisotropic function, could be the FA or QA (quantitative anisotropy, from GQI) volume, D represent a direction (i.e. a point on the unit sphere), and A_{thr} is the anisotropy threshold. How to get these information can be found more detail in the previous section about reconstruction 4. An example about EuDX tracking from the seed point is presented in figure 22

The main input parameters of EuDX are:

- an anisotropic scalar metric e.g. FA
- the indices for the peaks on the sampling sphere.

In its simplest form, this consists of starting at a seed location and following the preferred direction until we reach a new voxel. We can then change to this voxels referred direction and carry on until an entire track is propagated as in figure 20). Due to that, some other important options of this algorithm are:

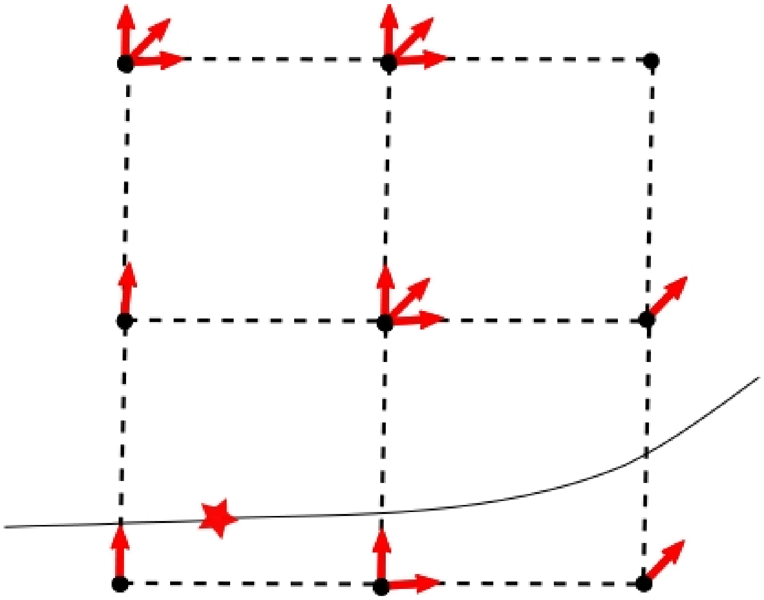

Figure 22: An example of EuDX tracking

- the number of random seeds where the track propagation is initiated,
- a stopping criterion, for example a low threshold for anisotropy. For instance if we are using Fractional Anisotropy (FA) a typical threshold value might be $a\_low = .2$

And the syntax for calling EuDX function as below:

```
eu=EuDX(a=FA,ind=ten.ind(),seeds=10000,a_low=.2)
```

EuDX returns a generator class which yields a further track each time this class is called. In this way we can generate millions of tracks without using a substantial amount of memory. However, in the current example that we only have 10000 seeds, and we can load all tracks in a list using list comprehension([]) without having to worry about memory. An example of what to do for generating millions of tracks with minimum memory usage can be found on the examples directory of Dipy website [8].

```
ten_tracks=[track for track in eu]
```

[8] http://www.dipy.org

In dipy we usually represent tractography as a list of tracks. Every track is a numpy array of shape (N,3) where N is the number of points in the track.

```
print ('The number of FA tracks is %d' % len(ten_tracks))
print ('The points in ten_tracks[130] are:')
print ten_tracks[130]
```

Another way to represent tractography is as a numpy array of numpy objects. This way has an additional advantage that it can be saved very easily using numpy utilities. In theory, in a list it is faster to append an element, and in an array is faster to access. In other words both representations have different pros and cons. Other representations are possible too e.g. graphtheoretic etc.

```
ten_tracks_asobj=np.array(ten_tracks,dtype=np.object)
np.save('ten_tracks.npy',ten_tracks_asobj)
print('FA tracks saved in ten_tracks.npy')
```

In the case there is a need to visualize the tractography, it can be easy to do by using FVTK, a module for visualization of Dipy. First, it is necessary to importe fvtk

```
from dipy.viz import fvtk
```

Then create an instance of class ***ren*** from fvtk and add tracts into this instance.

```
renderer = fvtk.ren()
fvtk.add(renderer, fvtk.line(ten_tracks,fvtk.red,opacity=1.0))
fvtk.show(renderer)
```

Even that you want to save these tracts for using in the next time, it can be done as bellow:

```
dpw = Dpy(<dpy_filename>, 'w')
dpw.write_tracks(ten_tracks)
dpw.close()
```

An example of tractography of subject 09 from Cambridge dataset is showed in the figure 23

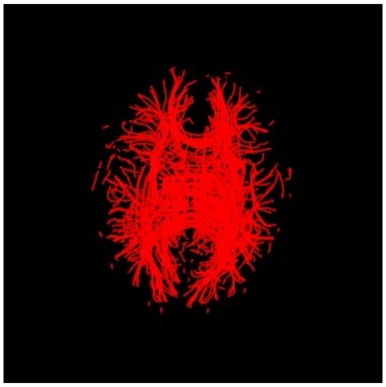

Figure 23: Tractography of subject 9 from Cambridge dataset

In conclusion, tractography is a visualization technique that can be used to extract lines of maximum diffusion coherence from any orientation distribution function map. These lines reflect the anatomy of the axonal trajectories. Many uses are foreseen for fiber tractography, from single-tract analysis to whole-brain connectivity analysis. This part describes some common techniques for tracking. There is also a detail about the deterministic methods. At the end of this section, the EuDX algorithm is presented together the Python command line. In the next part, the last step of pre-processing of dMRI, the co-registering will be discussed.

6 Coregistration: from native space to MNI space

In the previous parts, tractographies were created using EuDX with FA from the raw data in dicom format. But, all these tractographies were initially in native space or space of scanner. Because every measurement has it own coordinator, it is very difficult for doctor or neuroscientist can compare, integrate or further studying them. They are needed to be warped into the common space. In the other way, registration is the process of transforming from native space into the coordinate system.

Image registration algorithms can also be classified according to the transformation models they use to relate the target image space to the reference image space. The first broad category of transformation models includes linear transformations, which include translation, rotation, scaling, and other affine transforms. Linear transformations are global in nature, thus, they cannot model local geometric differences between images. The second category of transformations is non-rigid or non-linear transformations. These transformations are capable of locally warping the target image to align with the reference image. Non-rigid transformations include radial basis functions (thin-plate or surface splines, multi-quadrics, and compactly supported transformations), physical continuum models (viscous fluids), and large deformation models (diffeomorphisms).

The current known methodologies on this subject is that where Leemans et al. [20] uses the invariance of curvature and tortion under rigid registration along with procrustes analysis to co-register together different tractographies. Mayer et al. used iterative closest point applied to register preselected bundles (bundles of interest - BOI) [21] [23] and extended it using probabilistic boosting tree classifiers for bundle segmentation in [22]. Durrleman et al. [10] reformulate the tracks as currents and implemented a currents based registration, Zvitia et al. [33], [34], used adaptive mean shift clustering to extract a number of representative fiber-modes. Each fibre mode was assigned to a multivariate Gaussian distribution according to its population thereby leading to a Gaussian Mixture model (GMM) representation for the entire set of fibers. The registration between two fiber sets was treated as the alignment of two GMMs and is performed by maximizing their correlation ratio. Ziyan et al. [32] developed a nonlinear registration algorithm based on the log-Euclidean poly-affine framework [2], however this is not a direct tractography registration algorithm as they first create scalar volumes, therefore they dont try to register the tracks as they are in their space.

In this section, the goal is to warp the tractography from native space into MNI space (Montreal Neurological Institute) [9], a standard brain based on the averaging of 58 peoples, as in figure 24. In contrast to the all the method mentioned before, in this implement we use FA registration mappings or affine transformation applied on tractography which is also most commonly used in the literature along with other tensor based methods [11]. Native image (i,j,k) coordinates are mapped to native world coordinates by the affine of the image. In the same way the MNI image coordinates are mapped to MNI world coordinates through the MNI affine. The registration of the subject from native to MNI is about registering the native world to the MNI world. This can be done according to many models,

[9] `http://www.mni.mcgill.ca/`

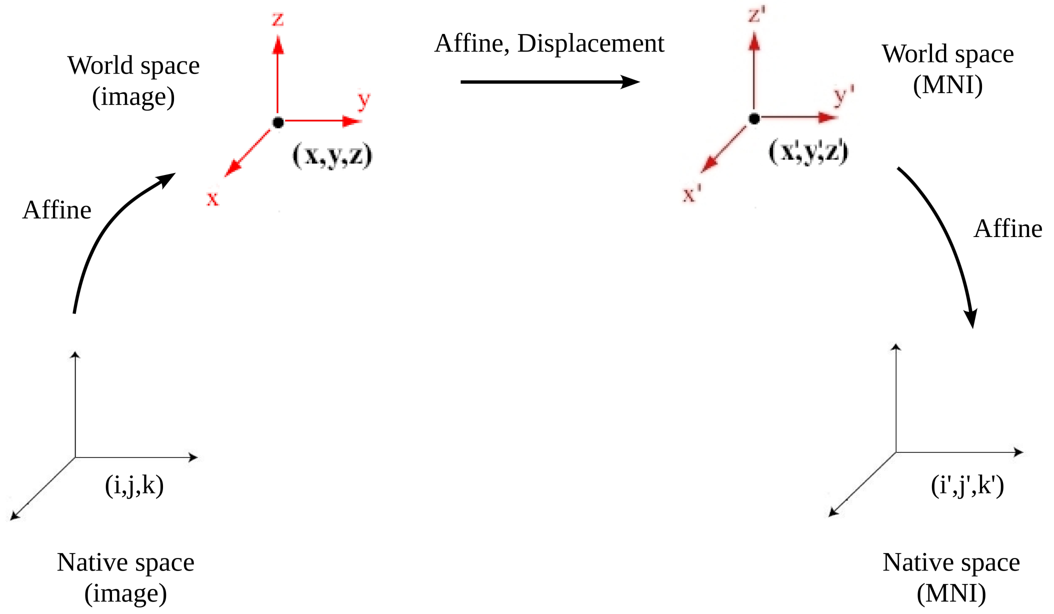

Figure 24: From native image space to MNI space.

e.g. affine transformation, rigid transformation or non-linear transformation, but always minimizing some loss function (e.g. correlation or mutual information) [31]. The data from the native image space are projected in native world and fitted to the template which is initially in MNI image space and through the affine in MNI world space.

```
 ==================================================

Coregistering
Input:   Tractography T in native space
Output:  Tractography T in MNI space

 ==================================================
```

There are three steps that have been done for this processing. In order to co-registering the tractography, we must know two parameters: affine and displacement. Affine is a linear transformation matrix containing information about converting between image space and world space. And the displacement is a point wise mapping from native space to MNI space.

It is technically very difficult with the FSL tools as they assume that these displacements will be applied only on volumetric data and not with point data as those used in tractographies.

The question is how to compute these two parameters. From the section ***reconstruction*** 4, we have the FA (fractional anisotrophy). This FA volume is also in native space. If there is an FA after warping in MNI space, it is possible to infer the affine and displacement. After that, these parameters can be used for registering the tractography. Due to this purpose, a standard template $FMRIB58_{F}A_{1}mm$ from the FSL toolbox was used as the reference volume. The FMRIB58 FA 1mm template has image coordinates (i,j,k) which corresponds to mm, because the size of one voxel is 1mm.

The transformation is usually done with FSL/FLIRT, which is linear, or FSL/FNIRT which is non-linear. To get the affine transformation matrix, the linear transformation FLIRT is usually used:

```
FLIRT
Input:  FA in native space
        Reference template

Output: Affine matrix

flirt -ref FMRIB58_FA_1mm  -in <FA> -omat  <affine>
```

or using non-linear transformation FNIRT

```
FNIRT
Input:  FA in native space
        Linear Affine matrix
        Reference template

Output: Nonlinear matrix

fnirt -in <FA> -aff <affine> -cout <nonlin>
```

Actually, after some considerable effort we found a combination of flirt,fnirt,invwarp,fnirtfileutils and fnirtfileutils-withaff which gave us the correct displacements. As this being very technical we will not describe it further here but the code is available in module (dipy.external.fsl) [10]. The

[10] http://nipy.sourceforge.net/dipy/index.html

command line for this can be done as follow

```
Create_displacements
Input:  FA in native space
        Linear Affine matrix
        Non-linear matrix
        Inverse warping matrix

Output: Displacement
        Displacement with affine
create_displacements(<FA>,<Aff>,<Non>,<Inv>,<Disp>,<DispA>)
```

After creating the displacement, the tractography will be applied to map them in the native space in the MNI space of voxel size 1 1 1mm3. Having all tractographies in MNI space is something very useful because we can now compare them against available templates or against each other and calculate different statistics.

```
Warp_displacements_tracts
Input:  Tractography in native space
        FA in native space
        Linear Affine matrix
        Inverse warping matrix
        Displacement
        Displacement with affine
        Reference template

Output: Warped tractography
warp_displacements_tracks(<T>,<FA>,<A>,<I>,<D>,<DA>,<R>,<TW>)
```

6.1 Linear vs. Non-linear registration: Issues

We attempted non-linear registration of FA native into MNI/FA using FSL/FNIRT. The result is an affine matrix and a displacement volume. This volume represents the displacements that each voxels has in order to match the template after the affine transformation. We used affine and displacements on tractography data to warp it to MNI space. This is a meaningful operation since both the tractography and the FA volume come from the same dMRI, i.e. they are in the same space. So far all good.

The issues start when we need to nonlinearly coregister T1 native into T1 MNI in order to superimpose T1 of the subject to his tractography in MNI space. The are 2 orders of issue:

1. Using FSL/FNIRT from T1 native to the MNI T1 template failed. The reasons are not clear but the result was visually not acceptable. This must be investigated more.

2. Even if nonlinear registration of T1 native to MNI T1 template succeed it is unclear whether the warping of T1 of the subject would be "compatible" to the warping of its tractography. T1 and tractography go along two different paths of registration and their results must be checked before accepting. It seems that until now no one attempted this nonlinear registration of both tractography and T1 that we attempted so there is no knowledge about this issue.

Since nonlinear registration suffered this issue and we needed to have both T1 and tractography in the same space we reverted to linear registration. So native FA/tractography was registered to MNI/FA template and then native T1 was registered to MNI T1. The result is not great but still fine. Note that this whole registration problem is an open issue in neuroimaging. But in the case that you do not need to have T1 coregistered, e.g. when doing clustering, then nonlinear coregistration via FSL/FNIRT is good enough. Another option can be using the linear transformation provided by NiPy, and it will be evaluated later.

6.2 Where is the Origin?

The importance of this topic becomes evident when trying to show volumes and tractographies from MNI space within an OpenGL window. In image (i, j, k) space the origin is on one corner of the $3D$ matrix/volume, i.e. the $(0, 0, 0)$ cell/entry. When moving to world space each voxel, which is a cube, has its own origin in the centre of the cube. This last one is an assumption made by nibabel and it might be different in other tools (e.g. Trackvis assume that the origin of a voxel/cube is in one corner). In native world space the origin is defined by the MRI vendor but usually it is the center of the volume of the scan. In MNI world space the origin is in the middle of the brain. In MNI image space the origin, i.e. $(i, j, k) = (0, 0, 0)$ is again in one corner of the $3D$ matrix/volume.

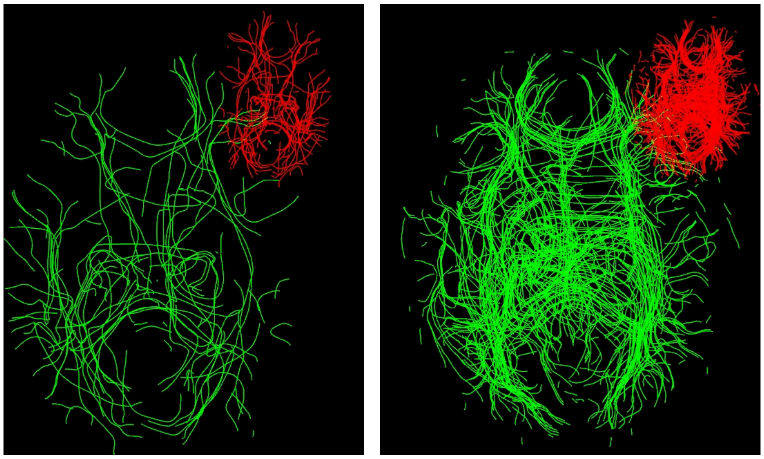

Figure 25: Tractography before and after coregistration

Given that the rotation and magnification parameters of the affine between MNI image and MNI world is $(-1.0, 1.0, 1.0)$, one voxel is $1mm$ and so the magnification is correct. But the origin is different: it is the corner of the volume for MNI image and center of the brain for MNI world, and a shift of half a volume is needed. More precisely the shift is $\frac{\text{Vol}-1}{2}$. In figure 25, the red tractographies in top right corner at each sub-image are the original tractographies in native space. The green big ones are warped tractography. The sub-image on the left is 100 tract fibers of subject 1, while the right one shows the whole tractography of subject 9. Both subject 1 and subject 9 are collected from Cambrigde dataset.

7 Future works

We first see what have done before and then move to something we should work on in future.

7.1 What Eleftherios Garyfallidis has done in his PhD period

Eleftherios is a final year PhD candidate at the University of Cambridge and a visiting student at the MRC-CBU under the supervision of Dr. Ian Nimmo-Smith (MRC-CBU) and Dr. Guy Williams (WBIC). The focus of his work is the development of models, methods and tools for brain tractography using diffusion imaging (dMRI). His work involves signal processing, machine learning, scientific visualisation and medical imaging. The thesis is focussed on finding new ways to reconstruct the diffusion signal and creating new highly efficient algorithms to segment tractographies. Most of the dMRI work is being implemented in a free, multi-platform and open tool called DiPy (Diffusion Imaging in PYthon http://www.dipy.org). Most of the visualization work is implemented to another open platform called fos (Free On Shades https://github.com/Fos). He stay at NILab from October 12th to November 12th, 2011. Here is what he has done before

1. DiPy (Diffusion Imaging in PYthon `http://www.dipy.org`).
2. FOS (Free On Shades `https://github.com/Fos`).
3. DNI, EIT.
4. EuDX (Euler integration Delta Crossing (for tracking).
5. QB (QuickBundles for registering or tractography directly without T1 or FA - feature based registrator).

7.2 Interesting works

Some of interesting projects that we can join in future are listed bellow: for Data standardlization, for Tractography and for Reconstruction step

1. Eddy current - to correct the data from removing noise origional from wrong megietic signal.
2. BET (Brain Etraction - although it is already intergrated in Nipy, but there are many things to study more).
3. Connectivity-tractograms (find the tractography between two given points).
4. Graph-theory tractography (apply graphic theory for tractography).
5. MultiTensor (Reconstruction step - currently Nipy and Dipy only works on one-tensor. Combination many tensors will provide more information about the spatial distribution of the diffusion signal within each voxel).

6. Diffusion Kurtosis[11] (Reconstruction step)
7. ActiveAx (microlevel - CHARMED) web [12] and paper [1] [13]

8 Cambridge dataset

The dataset used in our implementation is from Cambridge University. The Cambridge DT-MRI dataset consists of a 3D array of voxels, for each of which a 2.5x2.5 diffusion tensor is calculated. Here are some features of this dataset

- 12 subjects (taken on 12 healthy persons)
- 101 ($+1$, i.e. $b = 0$) gradients
- bvalues arrange from 0 to 4000
- Voxel size: $2.5 \times 2.5 \times 2.5mm^3$.

References

[1] Daniel C. Alexander, Penny L. Hubbard, Matt G. Hall, Elizabeth A. Moore, Maurice Ptito, Geoff J. M. Parker, and Tim B. Dyrby. Orientationally invariant indices of axon diameter and density from diffusion MRI. *NeuroImage*, 52(4):1374–1389, October 2010.

[2] Vincent Arsigny, Olivier Commowick, Nicholas Ayache, and Xavier Pennec. A Fast and Log-Euclidean Polyaffine Framework for Locally Linear Registration. *J. Math. Imaging Vis.*, 33:222–238, February 2009.

[3] P. J. Basser, J. Mattiello, and D. LeBihan. MR diffusion tensor spectroscopy and imaging. *Biophysical journal*, 66(1):259–267, January 1994.

[11] `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5711642&tag=1`

[12] `http://cmic.cs.ucl.ac.uk/camino/index.php?n=Tutorials.ActiveAx`

[13] `http://www.sciencedirect.com/science/article/pii/S1053811910007755`

[4] P. J. Basser, S. Pajevic, C. Pierpaoli, J. Duda, and A. Aldroubi. In vivo fiber tractography using DT-MRI data. *Magnetic resonance in medicine : official journal of the Society of Magnetic Resonance in Medicine / Society of Magnetic Resonance in Medicine*, 44(4):625–632, October 2000.

[5] T. E. J. Behrens, H. Johansen Berg, S. Jbabdi, M. F. S. Rushworth, and M. W. Woolrich. Probabilistic diffusion tractography with multiple fibre orientations: What can we gain? *NeuroImage*, 34(1):144–155, January 2007.

[6] T. E. J. Behrens, M. W. Woolrich, M. Jenkinson, H. Johansen-Berg, R. G. Nunes, S. Clare, P. M. Matthews, J. M. Brady, and S. M. Smith. Characterization and propagation of uncertainty in diffusion-weighted MR imaging. *Magn. Reson. Med.*, 50(5):1077–1088, 2003.

[7] P. T. Callaghan, C. D. Eccles, and Y. Xia. NMR microscopy of dynamic displacements: k-space and q-space imaging. *Journal of Physics E: Scientific Instruments*, 21(8):820–822, 1988.

[8] T. E. Conturo, N. F. Lori, T. S. Cull, E. Akbudak, A. Z. Snyder, J. S. Shimony, R. C. McKinstry, H. Burton, and M. E. Raichle. Tracking neuronal fiber pathways in the living human brain. *Proceedings of the National Academy of Sciences of the United States of America*, 96(18):10422–10427, August 1999.

[9] M. Descoteaux, R. Deriche, T. R. Knosche, and A. Anwander. Deterministic and Probabilistic Tractography Based on Complex Fibre Orientation Distributions. *Medical Imaging, IEEE Transactions on*, 28(2):269–286, 2009.

[10] Stanley Durrleman, Pierre Fillard, Xavier Pennec, Alain Trouvé, and Nicholas Ayache. Registration, Atlas Estimation and Variability Analysis of White Matter Fiber Bundles Modeled as Currents. *NeuroImage*, November 2010.

[11] Alvina Goh and RenÃ© Vidal. Algebraic Methods for Direct and Feature Based Registration of Diffusion Tensor Images. pages 514–525, 2006.

[12] N. S. Hageman, D. W. Shattuck, K. Narr, and A. W. Toga. A diffusion tensor imaging tractography algorithm based on Navier-Stokes

fluid mechanics. In *Biomedical Imaging: Nano to Macro, 2006. 3rd IEEE International Symposium on*, pages 798–801. IEEE, April 2006.

[13] P. Hagmann, T. Reese, W. Tseng, R. Meuli, J. Thiran, and V. Wedeen. Diffusion Spectrum Imaging Tractography in Complex Cerebral White Matter: An Investigation of the Centrum Semiovale. In *International Society for Magnetic Resonance in Medicine, ISMRM TWELFTH SCIENTIFIC MEETING, Kyoto, Japan, 15-21 May 2004*, volume 12 of *Proc. Intl. Soc. Mag. Reson. Med*, page 623. International Society for Magnetic Resonance in Medicine, 2004.

[14] P. Hagmann, J. P. Thiran, L. Jonasson, P. Vandergheynst, S. Clarke, P. Maeder, and R. Meuli. DTI mapping of human brain connectivity: statistical fibre tracking and virtual dissection. *NeuroImage*, 19(3):545–554, July 2003.

[15] Patric Hagmann, Lisa Jonasson, Philippe Maeder, Jean-Philippe Thiran, Van J. Wedeen, and Reto Meuli. Understanding Diffusion MR Imaging Techniques: From Scalar Diffusion-weighted Imaging to Diffusion Tensor Imaging and Beyond1. *Radiographics*, 26(suppl 1):S205–S223, October 2006.

[16] S. Jbabdi, M. W. Woolrich, J. L. R. Andersson, and T. E. J. Behrens. A Bayesian framework for global tractography. *NeuroImage*, 37(1):116–129, August 2007.

[17] Ning Kang, Jun Zhang, E. S. Carlson, and D. Gembris. White matter fiber tractography via anisotropic diffusion simulation in the human brain. *Medical Imaging, IEEE Transactions on*, 24(9):1127–1137, August 2005.

[18] Denis Le Bihan and Heidi Johansen-Berg. Diffusion MRI at 25: Exploring brain tissue structure and function. *NeuroImage*, November 2011.

[19] Denis Le Bihan, Jean-François Mangin, Cyril Poupon, Chris A. Clark, Sabina Pappata, Nicolas Molko, and Hughes Chabriat. Diffusion tensor imaging: Concepts and applications. *J. Magn. Reson. Imaging*, 13(4):534–546, 2001.

[20] A. Leemans, J. Sijbers, S. De Backer, E. Vandervliet, and P. Parizel. Multiscale white matter fiber tract coregistration: A new feature-based approach to align diffusion tensor data. *Magnetic Resonance in Medicine*, 55(6):1414–1423, 2006.

[21] A. Mayer and H. Greenspan. Bundles of interest based registration of White Matter tractographies. In *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*, pages 919–922. IEEE, May 2008.

[22] A. Mayer, G. Zimmerman-Moreno, R. Shadmi, A. Batikoff, and H. Greenspan. A Supervised Framework for the Registration and Segmentation of White Matter Fiber Tracts. *Medical Imaging, IEEE Transactions on*, 30(1):131–145, January 2011.

[23] Arnaldo Mayer and Hayit Greenspan. *Direct Registration of White Matter Tractographies with Application to Atlas Construction*. 2007.

[24] S. Mori, B. J. Crain, V. P. Chacko, and P. C. van Zijl. Three-dimensional tracking of axonal projections in the brain by magnetic resonance imaging. *Annals of neurology*, 45(2):265–269, February 1999.

[25] Susumu Mori and Peter C. M. van Zijl. Fiber tracking: principles and strategies a technical review. *NMR Biomed.*, 15(7-8):468–480, 2002.

[26] Evren Özarslan and Thomas H. Mareci. Generalized diffusion tensor imaging and analytical relationships between diffusion tensor imaging and high angular resolution diffusion imaging. *Magnetic Resonance in Medicine*, 50(5):955–965.

[27] Geoffrey J. M. Parker, Hamied A. Haroon, and Claudia A. M. Wheeler-Kingshott. A framework for a streamline-based probabilistic index of connectivity (PICo) using a structural interpretation of MRI diffusion measurements. *J. Magn. Reson. Imaging*, 18(2):242–254, August 2003.

[28] Stephen M. Smith. Fast robust automated brain extraction. *Hum. Brain Mapp.*, 17(3):143–155, November 2002.

[29] 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, June 2002.

[30] Fang-Cheng Yeh, V. J. Wedeen, and W. Y. I. Tseng. Generalized - Sampling Imaging. *Medical Imaging, IEEE Transactions on*, 29(9):1626–1635, 2010.

[31] Yang-Ming Zhu and Steven M. Cochoff. Influence of Implementation Parameters on Registration of MR and SPECT Brain Images by Maximization of Mutual Information. *J Nucl Med*, 43(2):160–166, February 2002.

[32] Ulas Ziyan, Mert R. Sabuncu, Lauren J. O'Donnell, and Carl F. Westin. Nonlinear registration of diffusion MR images based on fiber bundles. In *Proceedings of the 10th international conference on Medical image computing and computer-assisted intervention - Volume Part I*, MICCAI'07, pages 351–358, Berlin, Heidelberg, 2007. Springer-Verlag.

[33] O. Zvitia, A. Mayer, and H. Greenspan. Adaptive mean-shift registration of white matter tractographies. In *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*, pages 692–695. IEEE, May 2008.

[34] O. Zvitia, A. Mayer, R. Shadmi, S. Miron, and H. K. Greenspan. Coregistration of White Matter Tractographies by Adaptive-Mean-Shift and Gaussian Mixture Modeling. *Medical Imaging, IEEE Transactions on*, 29(1):132–145, January 2010.