ÉCOLE POLYTECHNIQUE
PROMOTION X2001
BOUCHER Maxime

RAPPORT DE STAGE D'OPTION SCIENTIFIQUE

Titre du Rapport
Regularization of Fiber Tractography in Diffusion Tensor Magnetic Resonance Imaging

NON CONFIDENTIEL

Option : INFORMATIQUE
Champ de l'option :
Directeur de l'option : Gilles DOWEK
Directeur de stage : Carl-Fredrik WESTIN
Dates du stage : 6 avril 2004 au 26 août 2004
Adresse de l'organisme :
Surgical Planning Laboratory,
Laboratory of Mathematics in Imaging
Brigham And Women's Hospital *a teaching affiliate of* Harvard Medical School
75, Francis Street
Boston, MA, USA
02130

Regularization and Fiber Tractography in Diffusion Tensor Magnetic Resonance Imaging

Maxime Boucher <maxime.boucher@polytechnique.org>

September 2, 2004

This report will describe the work I accomplished as part of my "Scientific Option" internship of my third year of study at the Ecole Polytechnique. However, as many people helped me to realize the work which will be described in the following pages, I would like to use this first page to thank those without whom I would not have been able to accomplish what I did and have a great experience here at the Surgical Planning Laboratory.

I would like to thank my supervisor, Dr. Carl-Fredrik Westin, the Laboratory of Mathematics in Imaging and the people I was working with at the SPL for giving me the chance to study an interesting problem in a stimulating and interesting research environment. I am very grateful for the suggestions and constructive comments on my work and the time you took to interest yourself in my problems to help finding solutions. I learned a lot during the few months of the internship and I am very grateful to all those without whom it would not have been possible.

Résumé

L'imagerie de la diffusion par résonnance magnétique s'est révélé être un outil puissant d'analyse pour des besoins cliniques allant des plus simples aux tâches les plus complexe d'établissement de diagnostiques spécialisés et la planification de traitements médicaux. Plus particulièrement, l'imagerie des tenseurs de diffusion (DTI) et la tractographie des fibres neurocérébrales ont ouverts des champs de recherches qui, auparavant, se basaient principalement sur des études post mortem. La tractographie par tenseurs de diffusion commence à devenir un outil de diagnostique pratique et plusieurs applications biomédicales en faisant usage font leur apparitions. Cette méthode utilise la direction de plus grande diffusion mesurée à l'aide de la DTI pour déterminer le chemin suivit par les nerfs dans le cerveau. Malheureusement, puisque les données mesurées sont bruitées et ont une résolution limitée, la qualité de l'information qui en est extrait en souffre grandement. Ce rapport étudie une nouvelle méthode pour réduire l'impact du bruit dans les données d'origine. Il étudiera l'utilisation de la transformée en ondelettes comme outil d'analyse du contenu des données et ensuite montrera comment pour chaque coefficients d'ondelettes si les variations locales dans le signal doivent être conservées ou enlevés. Afin d'obtenir une estimation multirésolution du signal, une méthode de seuillage récursif est utilisé. Ce rapport étudie aussi comment il est possible d'améliorer davantage l'analyse des images en rééchantillonnant le signal le long des lignes de flôts. Par la suite, une méthode pour résoudre les équations différentielles ordinaires est présentée et son application à la tractographie et les tenseurs de diffusion est discutée. Une méthode d'interpolation de données discrètes pour les tenseurs et un critère d'arrêt sont aussi étudiées. Une implémentation algorithmique à l'intérieur d'un logiciel est présentée et quelques résultats sont montrés.

Abstract

MR diffusion imaging has become a powerful, multi-faceted tool both for very basic clinical needs and for advanced, specialized diagnosis and treatment planning. In particular, diffusion tensor imaging (DTI) and nerve fiber tractography have opened up new research possibilities in areas that hitherto relied largely on postmortem studies. Diffusion tensor tractography is starting to become a diagnostically helpful tool and new applications are emerging. It uses the principal diffusion direction measured with DTI to compute the pathways of complete nerve fiber tracts. However, as the data acquired has limited resolution and contains noise, the quality of the information extracted from it will be greatly limited by this. This report investigates novel methods for reducing the impact of noise in the original data. We investigate the use of the wavelet transform to analyze the content of the data and then decide for each coefficient if the local variation in the signal should be kept or removed. To obtain a multiresolution filtered estimation of the noisy signal, a recursive thresholding scheme is applied to the remaining scaled signal. We further investigate how to improve the overall filtering scheme by resampling the signal along the flow lines. A method for solving ordinary differential equation is also introduced and applied to the special case of tensor data. We investigate a continuous description of the tensor data and how to introduce a stopping criteria in the tractography. Finally we describe algorithmic implementations and provide results of reconstructing white matter tracts in the human brain in vivo.

Contents

Chapter 1

Introduction

Water diffusion is a well known phenomenon since the Brownian motion was described by a botanist, Robert Brown, at the beginning of the 19^{th} century and the theoretical framework given by Einstein at the beginning of the 20^{th} successfully explained it. However, the scientific community had to wait until recent development in Magnetic Resonance Imaging (MRI) to be able to effectively measure water diffusion inside complex structure like the human brain. Nowadays, diffusion tensor imaging (DTI) prooved to be useful in application ranging from basic clinical needs to highly sophisticated imaging application. Hence, the intricate connective architecture of the most complex human organ can be studied non-invasively.

In this internship, I had the chance to study the regularization of fiber tractography, an application of DTI, which ables us to fully reconstruct the inner structure of the white matter by following fiber tracts in a water diffusion tensor map. Moreover, the main objective was to program a fully working tractography program implemented as a module inside the main software developped at the Surgical Planning Laboratory, the 3D-Slicer. This complex project required to solve many problems and the purpose of this report is to describe the fiber tractography module I implemeted. The structure of the report is the same as the way the reconstruction of in-vivo white matter contruction should be done. I first remind the principle of water diffusion and explain how it can be measured using magnetic resonnance imaging. Afterwards, I describe a method to regularize the signal by filtering the noise inside the slices. The method described is based on a novel bandelet transform [7] and in this report, practical results of its ability to preserve edges in signal denoising is given. After that, I explain the tractography method

implemented into the module to extract the white matter structure from the diffusion tensor data with the regularization scheme to avoid discontinuous fibers and stop where the fibers end. Finally, I explain how I complied with the main objective by describing the implementation inside the 3D-Slicer and the programming environment with the wide range of tools available at the SPL. I will also show the interface of the module and some results.

Chapter 2

Diffusion Tensor Magnetic Resonance Imaging

Diffusion tensor imaging is an imaging technique which can be used to investigate the structure of highly oriented fibers - such as muscles or the white matter - inside the human body without performing any kind of surgical or chemical invasion. It is based on the ability of magnetic resonance imaging to measure water diffusivity inside a fluid. However, the images, depending on numerous factors such as body motion and cardiac pulse, can be really noisy and those factors impair our ability to perform fiber tractography. Hence, in order to better understand the origin of the noise, it is important to earn a better understanding of this imaging technique.

2.1 The water diffusion phenomenon

Water diffusivity is a well known phenomenon since Robert Brown observed the random motion of pollen inside water and described what is called the Brownian motion at the beginning of the 19^{th} century. However, the scientific community had to wait until 1905 when Einstein gave a theoretical framework to water diffusion to really understand this phenomenon. Nowadays, it is well known that the water diffusion is related to the random paths water molecules follows from the point where they start. Hence, the probability to find a water molecule around its starting position after a small amount of time can be seen in the left figure 2.1. This is the isotropic diffusion case. However, this probability distribution assumes that the water molecule is free

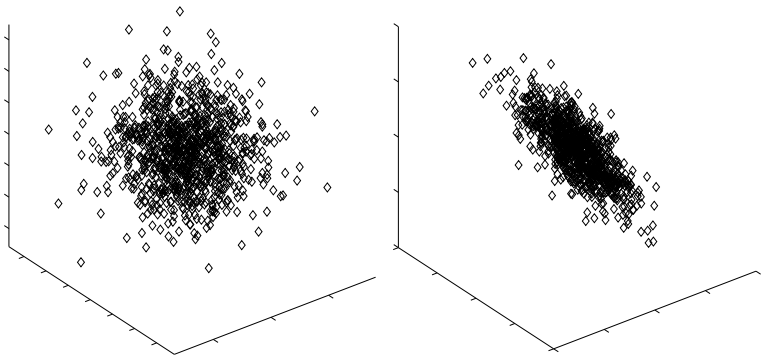

Figure 2.1: Probability distribution of the diffusion of water inside an isotropic medium like water and an anisotropic medium like white matter

to diffuse in any direction. Therefore, if there are biological barriers such as cell membranes, macromolecules, axons, which tend to slow water diffusion along particular directions, it will interestingly modify the measured diffusion pattern like it can be seen in the right image in figure 2.1. Those constraints can reveal information about the inner structure of the investigated tissue. However, in most tissue, diffusivity barriers such as cell membranes present no particular macroscopic voxel size direction and they tend to void each other which brings us back to the case of isotropic diffusion. Obviously, such diffusivity does not tell us much about the inside structure of a tissue but, however, in white matter, the myelinated fibers covering the axons of the neurons are tightly packed to connect the different regions and offer a great barrier to water diffusion as it is shown in figure 2.2. This brings an interesting case, where analyzing water diffusivity inside a tissue gives us an overall orientation field of the fibers it contains. Moreover, inside white matter this field can give us directly the direction of the axon propagation inside it and given a good tracking method, it makes it possible to reconstruct the overall structure of the connection inside the white matter.

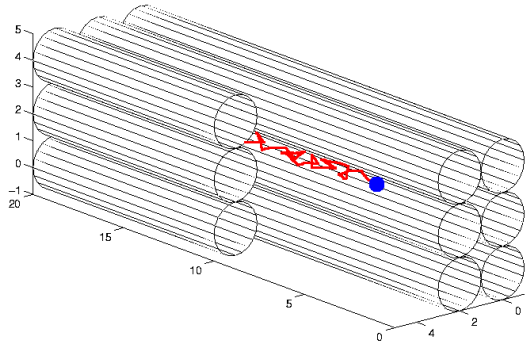

Figure 2.2: Diffusion of water inside tightly packed myelinated fibers

2.2 Tensor Notation

Since this report greatly uses tensors, a tensor notation is used to note matrices and vectors. Therefore, throughout this report, vectors, or first order tensors, are noted $\underline{v}$. Matrices, or second order tensors, are noted $\underline{\underline{m}}$. The tensor cross product is noted $\otimes$ and if $\underline{\underline{m}} = \underline{v} \otimes \underline{w}$ then $m_{ij} = v_i \cdot w_j$ if m_{ij} is the i^{th} component of the j^{th} column of $\underline{\underline{m}}$ and v_i is the i^{th} component of $\underline{v}$. Finally, the tensor double dot product is noted : and if $s = \underline{\underline{m}} : \underline{\underline{n}}$ then $s = \sum_{ij} m_{ji} \cdot n_{ij}$.

2.3 Measuring Diffusion

Unless it is possible to measure diffusion, any theoretical analysis of it remains useless. Luckily, through different contributions, it is possible nowadays to measure such diffusion. The first contribution was made by Stejskal and Tanner in 1965 [8]. They described an imaging sequence able to measure diffusion inside water. The second contribution was made by LeBihan in 1986 [5] which introduce diffusion weighted imaging The third was made by Basser [14] a few years later in 1994 which introduced diffusion tensor magnetic resonance imaging.

The imaging sequence introduced by Stejskal and Tanner uses the water molecule property of being electromagnetically oriented along a particular

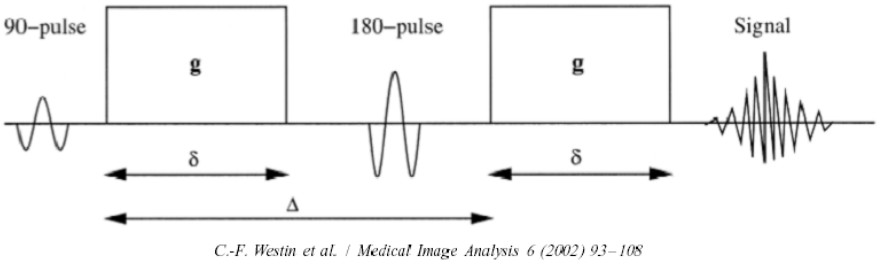

Figure 2.3: The Stejskal-Tanner Imaging Sequence. Picture is from [18]

direction, to measure its diffusivity. It consist of two strong diffusion pulse gradients $\underline{g}$ positioned around a 180° refocusing pulse. Such a signal can be seen in figure 2.3. The first 90° pulse induces a phase shift in the spin of every water molecule and the 180° second corrects it for the water molecule which refocused to their original position. For the molecules which did not, a signal loss occurs and the remaining signal is measured. Therefore, by inserting gradients ($\underline{g}$) along particular directions, the displacement along specific direction can be amplified if no biological contraint is presnet resulting in greater signal loss. Hence, since a diffusion tensor has six independant components, by measuring the signal along six different gradient directions, it should be possible to build diffusion tensor. However, the images obtained has to be corrected to take into account the density of spin, e.g. the density of water molecules as proposed by LeBihan [5]. Such a density can be obtained by using the same signal with no gradients $\underline{g}$. To better understand how it works, an example signal is shown in figure 2.4. This figure shows the measured water density in a brain slice. By looking in figure 2.5 which represents six different measured gradients slices, it is possible to observe the difference in the signal loss by inserting different diffusion gradients $\underline{g}$. The actual diffusion is obtained from two images by using the formula introduced by Stejskal and Tanner [8]

$$S = S_0 e^{-bD} \tag{2.1}$$

where D is the water diffusion and b is the weighting factor introduced by LeBihan [5] with

$$b = \gamma^2 \delta^2 (\Delta - \frac{\delta}{3}) \|\underline{g}\|^2 \tag{2.2}$$

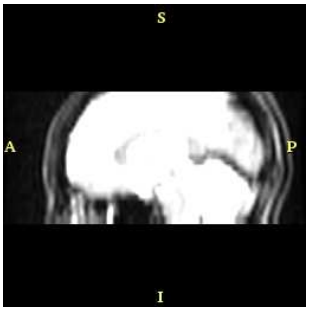

Figure 2.4: Measured density of spin inside the brain

where γ is the proton gyromagnetic ratio (42 MHz/Tesla), δ is the duration of the gradient pulse $\underline{g}$ and Δ is the duration between the two gradient. Finally, to build tensors from the signals, the employed method in this report is the line scan diffusion imaging [10, 15]. This method has been proven to be relatively insensitive to bulk-motion and it can be deduced from equation 2.1 by writing the tensor form of the Stejskal-Tanner formula

$$S = S_0 e^{-\frac{b}{\|\underline{g}\|^2}\underline{g}^T \underline{\underline{D}}\underline{g}} \tag{2.3}$$

where $\underline{\underline{D}}$ is the diffusion tensor and $\underline{g}$ is the diffusion gradient in $3D$ coordinates. Therefore, with the six gradients

$$\begin{array}{ccc} \underline{g}_1 = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\\0\end{pmatrix} & \underline{g}_2 = \frac{1}{\sqrt{2}}\begin{pmatrix}0\\1\\1\end{pmatrix} & \underline{g}_3 = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\1\end{pmatrix} \\ \underline{g}_4 = \frac{1}{\sqrt{2}}\begin{pmatrix}0\\1\\-1\end{pmatrix} & \underline{g}_5 = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\\0\end{pmatrix} & \underline{g}_6 = \frac{1}{\sqrt{2}}\begin{pmatrix}-1\\0\\1\end{pmatrix} \end{array} \tag{2.4}$$

the six different diffusion images are obtained

$$\beta_k = \underline{g}_k^T \underline{\underline{D}}\underline{g}_k = \underline{\underline{D}} : (\underline{g}_k \otimes \underline{g}_k), k = 1 \dots 6 \tag{2.5}$$

Such images can be seen in figure 2.5. The basis of the tensor can be written as $\underline{\underline{G}}_k = \underline{g}_k \otimes \underline{g}_k, k = 1 \dots 6$. Reconstructing the diffusion tensors from the six images β_k is simple if the tensor basis $\underline{g}_k \otimes \underline{g}_k$ allows a dual basis which

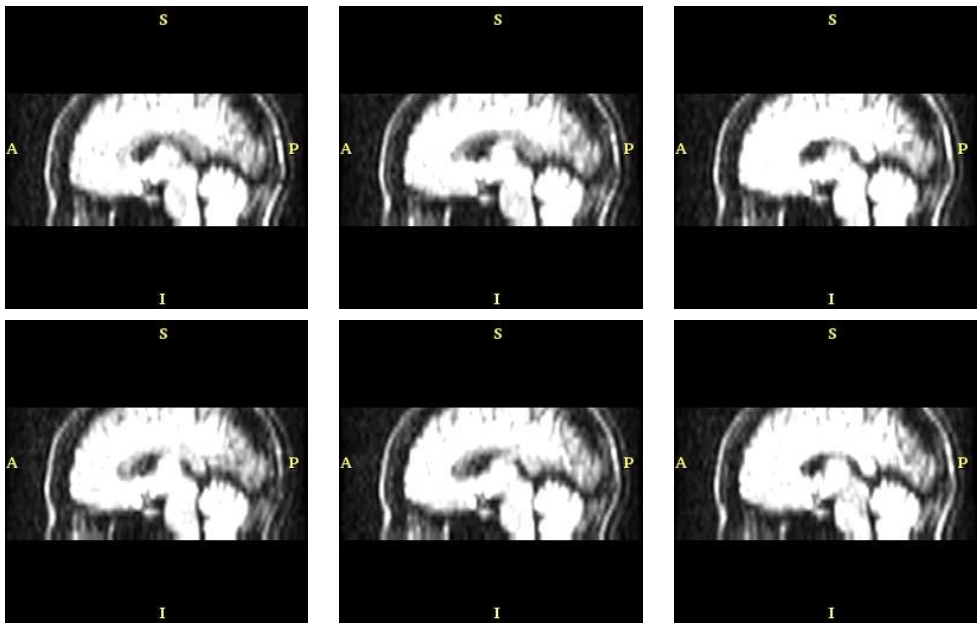

Figure 2.5: Six gradient images from which it is possible to reconstruct tensor data

is given in equation 2.6

$$\underline{\underline{\tilde{G}}}_1 = \frac{1}{2}\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \quad \underline{\underline{\tilde{G}}}_2 = \frac{1}{2}\begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \quad \underline{\underline{\tilde{G}}}_3 = \frac{1}{2}\begin{pmatrix} 1 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$
$$\underline{\underline{\tilde{G}}}_4 = \frac{1}{2}\begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{pmatrix} \quad \underline{\underline{\tilde{G}}}_5 = \frac{1}{2}\begin{pmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \quad \underline{\underline{\tilde{G}}}_6 = \frac{1}{2}\begin{pmatrix} 1 & 0 & -1 \\ 0 & -1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \tag{2.6}$$

It is easily possible to verify that $\underline{\underline{\tilde{G}}}_i : \underline{\underline{G}}_j = 0$ for $i \neq j$ and $\underline{\underline{G}}_i : \underline{\underline{\tilde{G}}}_j = 1$ if $i = j$. Therefore, since it has at least as much component (six) as the number of independant component in a diffusion tensor, it can be deduced from 2.5 that

$$\underline{\underline{D}} = \sum_{k=1}^{6} \beta_k \underline{\underline{\tilde{G}}}_k \tag{2.7}$$

A longer and more complete explanation of this reconstruction can be found in [18]. A reconstruction of the tensor data can be seen in figure 2.6. What

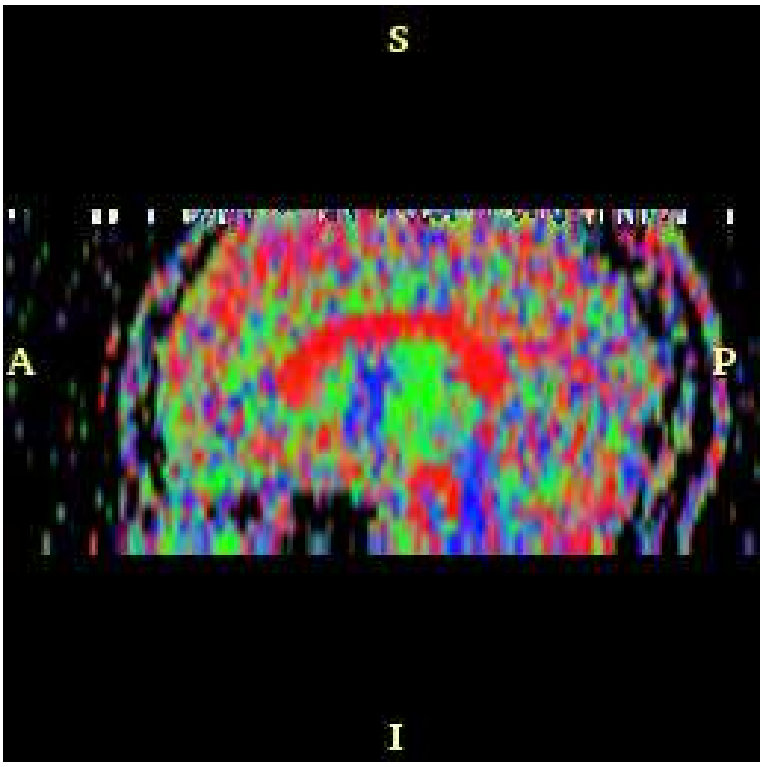

Figure 2.6: A Diffusion Tensor Slice. The colors represent the direction of the eigenvector of the highest eigenvalue of the tensor. Red means the principal direction is perpendicular to the slice

is plotted in this figure is a color coded orientation of the highest diffusion direction. This shows that, using magnetic resonance imaging, it is possible to acquire a diffusion tensor map in-vivo of a living brain. Nevertheless, as it is shown in figure 2.6, the data contain noise and if it not removed or if the data is not regularized, it may make it harder to extract fibers and perform tractography. This is the main problem I had to solve in this internship and it shows the importance of regularization of fiber tractography.

Chapter 3

Signal Filtering

The previous chapter explored a method for measuring water diffusion inside the white matter using magnetic resonance imaging. However, like any other analogic signal, it might contain noise and if it is not filtered, it will affect the quality and the to regularity of the information extracted from it. Low pass filtering the signal can help regularize it, but using only a linear filter will fail to keep some important features such as the edges of the white matter. Therefore, an estimator which offers some robustness to those features is needed. This chapter will describe a multiresolution spatially adaptive filtering technique based on a novel bandelet transform proposed by [7] and will give example of regularized data using this filter.

3.1 Wavelets: A Time-Frequency Analysis

The filtering method will use a powerful mathematical tool to describe a basis over which information will be concentrated along a few coefficients and noise along every coefficients. This distribution will permit easy discrimination between the original signal and noise and the filtering method will consist of a simple thresholding of the smallest coefficients associated to the noise. However, in order to understand the filtering method, it is important to make a brief introduction of the mathematical tool in question, the wavelets. Nevertheless, a complete introduction to this field would be too fastidious and is not the intended purpose of this report. If the reader wants to read a good book about the wavelet theory, he might want to refer to [16].

3.1.1 The Windowed Fourier Transform

Using proper equipment, it is very easy to record a human speech and obtain a discrete representation of what was said. Using the Fourier transform it is also possible to analyze its specter. Nevertheless, the specter of the signal will hardly give us useful information about what was actually said. It will be possible to tell if there was "a" or "e" in the speech but the specter reveals no information about when the "a" or the "e" was positionned in the speech. To tell such a thing a time-frequency analysis tool rather than a frequency analysis like the Fourier transform is needed.

Suppose we have a signal $S(t)$ and its Fourier transform $\mathcal{F}\{S(t)\} = \hat{S}(\xi)$. We will try to tell where are located the features in the signal. A feature is anything relevant to the signal such an "a" or an "e" for a speech or an edge in an image. An easy way to do this is to perform the Fourier transform only on a small part of the signal. We therefore define a window function $W(x,t)$ which is centered around t and is defined by (as an example)

$$W(x,t) = \frac{1_{\|t-x\|<a}}{2a} \tag{3.1}$$

The windowed Fourier transform of the signal is simplify defined by projecting the signal on this function.

$$W\mathcal{F}\{S(t)\} = W\hat{S}(x,\xi) = \int_{\mathcal{R}} S(t)W(x,t)e^{-i\xi t}dt \tag{3.2}$$

The windowed Fourier transform makes possible to detect features in a signal by analyzing small parts of it. However, what is gained in time resolution is lost in frequency resolution. Multiplying the signal by the windowing function is in fact the same as convoluating the frequency function by a low pass filter $\hat{W}(x,\xi)$. The associated low pass filter is

$$\begin{aligned} \hat{W}(x,\xi) &= \int_{\mathcal{R}} \frac{1_{\|t-x\|<a}}{2a} e^{-i\xi t}dt \\ &= \frac{1}{2a}\int_{x-a}^{x+a} e^{-i\xi t}dt \\ &= \frac{e^{-i\xi x}sin(a\xi)}{a\xi} \\ &= e^{-i\xi x}sinc(a\xi) \end{aligned} \tag{3.3}$$

From this function, it is possible to see that the smaller is a, the more information we will get about "when the feature happened" (sharper box function) but, however, the least information we will get about "what actually happened". This the Heisenberg theorem applied to signal processing,

it is impossible to obtain perfect information about the frequency and the time. In fact, it is possible to proove that if we note Δx the resolution in the time domain and $\Delta\xi$ the resolution in the frequency domain, we will always have $\Delta x \cdot \Delta\xi \geq \frac{1}{2}$ and the equality is obtained for a gaussian like windowing function (see [16] for a proof).

3.1.2 A Projection Basis

Using this knowledge about the time-frequency resolution problem, we will design an orthonormal basis to project a discrete signal to obtain the more useful information as possible. To be complete, the projection basis should cover the whole domain where the function is defined in time and in frequency. We suppose we have a windowing function $W(t)$ with a time resolution Δt and a frequency resolution Δf. An easy orthonormal projection basis can be build by translating this windowing function in time by an amount of Δt and in frequency by an amount of Δf. This therefore form an orthonormal basis of the whole time-frequency domain.

$$W_{j,m}(t) = e^{-ij\Delta ft} W(t - m\Delta t) \tag{3.4}$$

Of course, this basis will be orthonormal for any function $W(t)$ if and only if

$$\begin{aligned} < W_{j_1,m_1}, W_{j_2,m_2} > &= 0 \text{ if } j_1 \neq j_2 \text{ or } m_1 \neq m_2 \\ &= 1 \text{ otherwise} \end{aligned} \tag{3.5}$$

where $< f, g >= \int_D f(t)g^*(t)dt$ is the scalar product. The projection of a discrete signal onto this basis is simply

$$W\mathcal{F}\{S(t)\} = \sum_{j,m} < S, W_{j,m} > W_{j,m}(t) \tag{3.6}$$

It is interesting to note that by using different value of Δt and Δf, it is possible to define the discrete fourier transform [16]. We can see how the time-frequency space is divided for different basis in figure 3.1.

3.1.3 The Discrete Wavelet Transform

When analyzing a signal, it might be more convenient to have a higher time resolution when analyzing high frequencies and a higher frequency resolution when analyzing lower frequencies. For example, when looking for a sharp

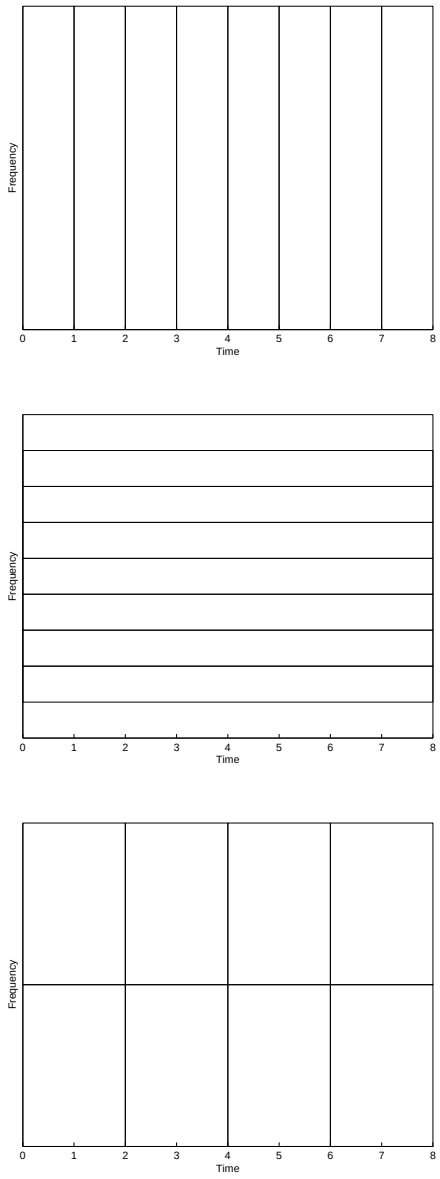

Figure 3.1: Different basis for the time-frequency domain.
Up: Pure Time basis
Middle: Pure Frequency Basis
Bottom: A Time-Frequency Basis

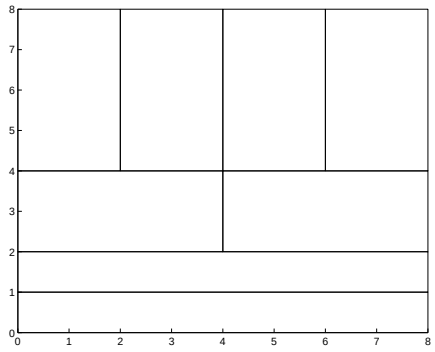

Figure 3.2: Division of the time-frequency domain with a wavelet basis. Higher frequencies have a smaller time support than lower frequencies.

edge, we would like to have a higher time resolution to be able to localize sharp edges that when looking to the global variation in the signal. This is why the wavelet basis was designed. A time-frequency representation of the wavelet basis is shown in figure 3.2. The projection is now performed over a wavelet function $\psi(t)$ instead of a windowing function $W(t)$. The wavelet basis is simply defined by translating and scaling the wavelet function.

$$\psi_{j,m}(t) = \frac{1}{\sqrt{2^j}}\psi(\frac{t-2^j m}{2^j}) \tag{3.7}$$

The discrete wavelet transform can be implemented easily by a pair of two biorthogonal filter [16]. Such a pair of filter can be seen in figure The projection is obtained recursively by filtering the signal by the two filters and then by downsampling the result by two. A scaled signal and a detail signal associated to the first order wavelet decomposition (highest frequencies) is obtained using this method. The recursive decomposition scheme consist of applying the two filters to the scaled version of the signal and downsampling again the result. An example of the first order of the decomposition of $2D$ signal (figure 3.4) is shown in figure 3.5. It is possible to see in this figure that the wavelet decomposition was able to highlight where are the highest variation in the image.

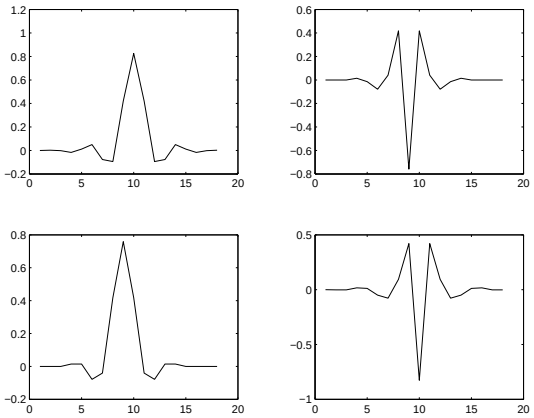

Figure 3.3: A scaling filter (left) with its associated wavelet (right) for decomposing a signal (top) and reconstructing it (bottom)

Figure 3.4: Lena, an example image widely used in signal processing

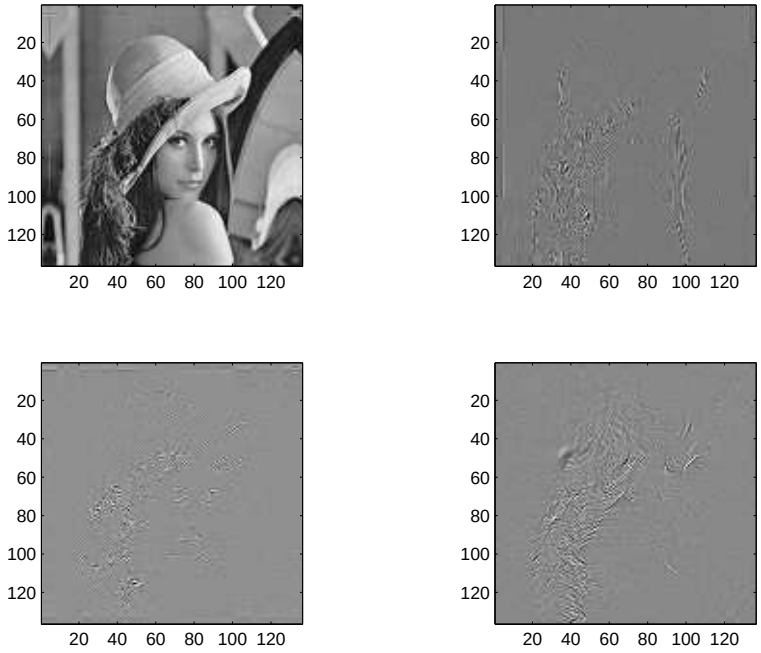

Figure 3.5: The wavelet decomposition of Lena
Top left: The remaining scaled signal
Top right: vertical details
Bottom left: Horizontal detail
Bottom right: Diagonal detail

Since the wavelet form an orthogonal basis, it is possible to obtain the original signal from the wavelet decomposition by upsampling the signal by two, then applying the conjugate mirror of the biorthogonal filters to the result and combining the two signal [16]. A representation of the decomposition and reconstruction algorithm is shown in figure 3.6.

3.2 An estimator for a noisy signal

One interesting property of the wavelet transform is that noise present in the original signal contribute to every coefficient of the detail images or wavelet coefficients while the original signal only contributes to a few coefficients. We will use this property to filter out the signal. Therefore, as the energy of the noise is distributed in a large number of coefficients and the energy of the signal in only a few, it should be easy to remove noise from an image by thresholding the smallest wavelet coefficients.

To better understand the filtering method we will use an example one-dimensional signal (top figure 3.7). We suppose that the signal was modified by a gaussian white noise (bottom figure 3.7). In fact, [13] showed that in the case of a noise level higher than 3db in Magnetic Resonnance, it is possible to make this hypothesis. Therefore, we now have a signal sampled at n different time t.

$$Y_i = S_i + Z_i, i = 1, \ldots, n \tag{3.8}$$

where S_i is the signal and Z_i is the gaussian white noise. We thus define a threshold estimator of a signal

$$F'_i = F_i \cdot 1_{\{|F_i| > T_c\}} \tag{3.9}$$

As stated above, using this thresholding operator to remove the smallest wavelet coefficients or, in other words, to smooth the smallest local variation, it should be possible to remove the noise from the signal. In fact, if we define a risk estimator as

$$R(F', F) = \frac{1}{n} E \|F' - F\|_{L_2}^2 \tag{3.10}$$

Donoho and Johnstone [6] showed that thresholding the smallest coefficient is a method to remove noise that is optimal in the sense that it minimizes

A Wavelet Tour of Signal Processing
Stéphane Mallat, Academic Press 1999 (2nd edition)

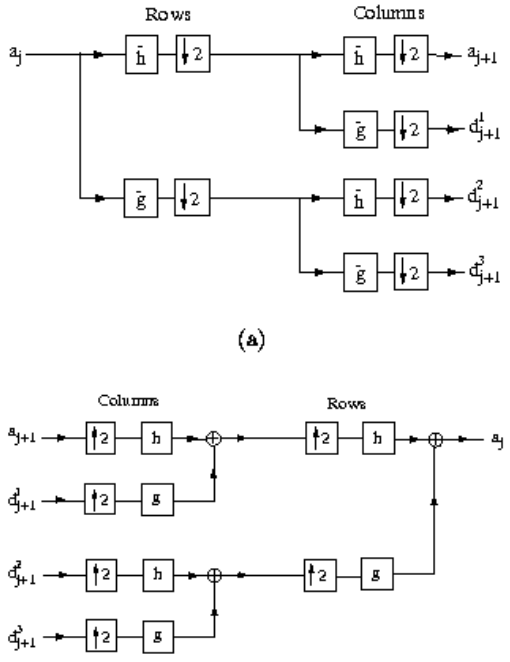

Figure 7.27: (a): Decomposition of a_j with 6 groups of one-dimensional convolutions and subsamplings along the image rows and columns. (b): Reconstruction of a_j by inserting zeros between the rows and columns of a_{j+1} and d^k_{j+1}, and filtering the output.

Figure 3.6: The wavelet decomposition and reconstruction scheme. The figure was taken from [16]

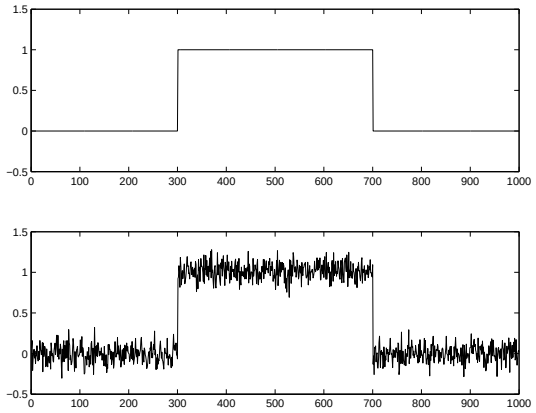

Figure 3.7: A signal and the same signal with added gaussian white noise of $\sigma = 0.1$

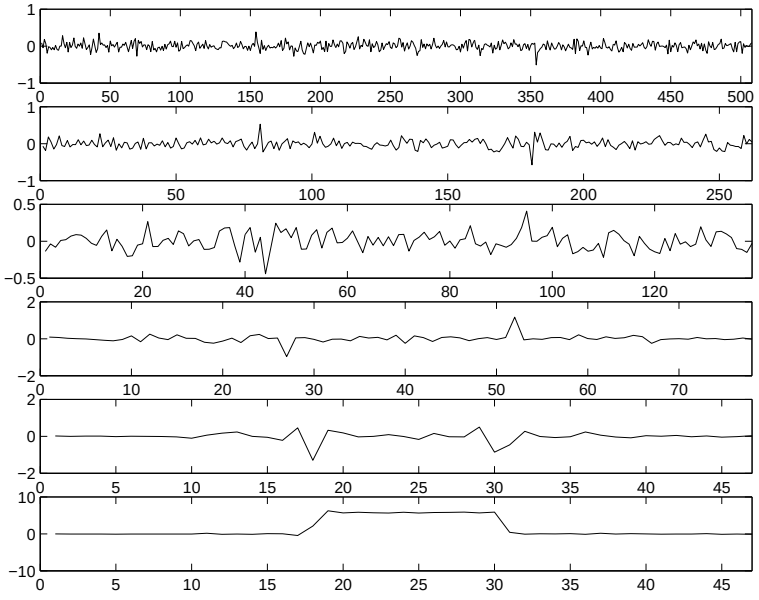

Figure 3.8: Wavelet decomposition for orders of 1 to 5. The last image at the bottom is the remaining scaled signal

the risk R while it maximizes signal recovery. They also prooved that the previous statement is true for a threshold value

$$T_c = \gamma\sqrt{2\log_e n}\cdot\sigma \tag{3.11}$$

where σ is the noise variance and γ is a parameter we can set to account for how much signal loss is acceptable. Therefore, this filtering technique can be applied easily to every signal by thresholding the wavelet coefficient and the multiresolution decomposition scheme consists of recursively applying it to the approximation signal.

As it is always better to look over a real example, we will try to remove the noise from the bottom signal in figure 3.7. We can see in figure 3.8 the order one to five wavelet decomposition of the signal and the remaining scaled signal. As we can see in the decomposition, the wavelet decomposition can relatively well detect the sharp edge while the remaining signal contains less noise while it also smooths the original sharp edge. If we choose $\gamma = 0.3$, the result of the thresholding is shown in figure 3.9. From this figure, we can see that the wavelet coefficients were almost all removed while those associated to the edges were kept. Finally, after the reconstruction, we can see on the same figure that the filtering method was able to give a good estimation of the original signal. The method extends directly from one dimension to N dimensions by using the same threshold over the N dimensional wavelet transform.

3.3 Improving Wavelet Filtering with Flow Analysis

Even if a N dimensional filtering method was define from a working one dimensional method, as we will see in the next example, some problems may appear in higher dimensions. In fact, in the first part of this chapter, a discussion about how the time-frequency domain could be divided to better represent the data. In this part, we will discuss how to divide 2D of ND images to highlight edges. Lets first what happens if no adaptation is done. In figure 3.10, we can see a signal and the same signal with noise added. In figure 3.11 we can see the wavelet decomposition of the signal and in figure 3.12 the thresholded version ($\sigma = 0.2$, $\gamma = 1.0$) of the wavelet decomposition. Finally, in figure 3.13 the result of the filtering method is

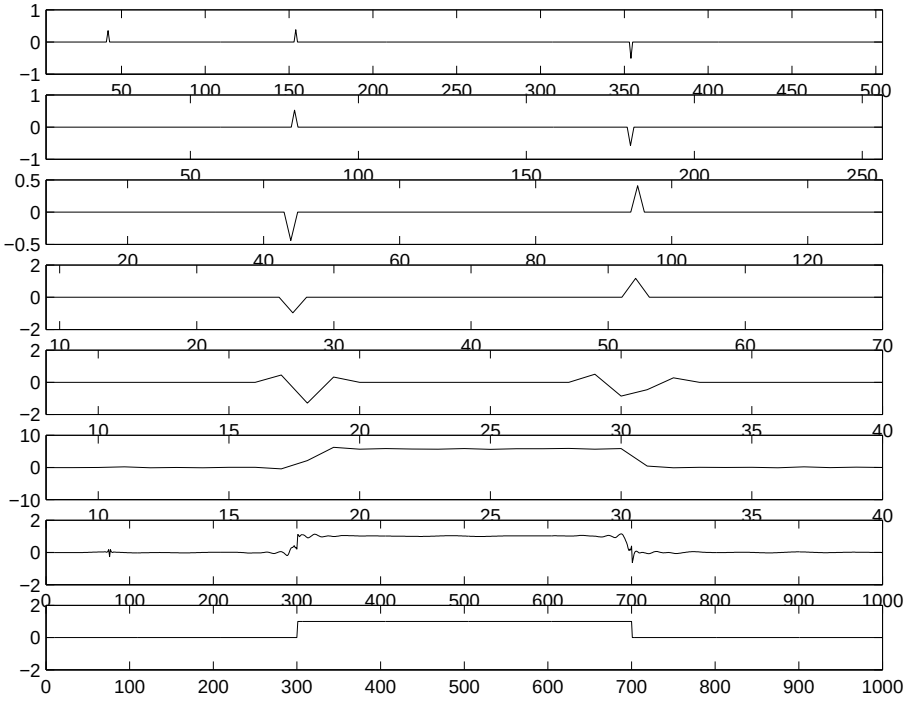

Figure 3.9: Example of the filtering method. The first five signals are the thresholded wavelet decomposition signal, the sixth is the remaining signal, the seventh is the reconstructed signal and the eigth is the original signal

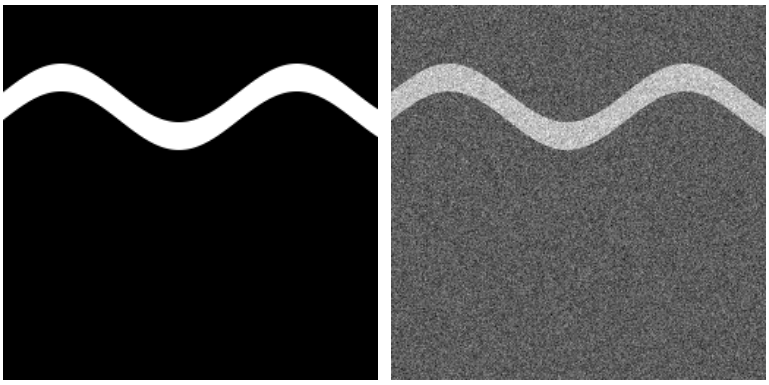

Figure 3.10: An example $2D$ signal (left) and the same signal with some gaussian white noise $\sigma = 0.2$ added (right)

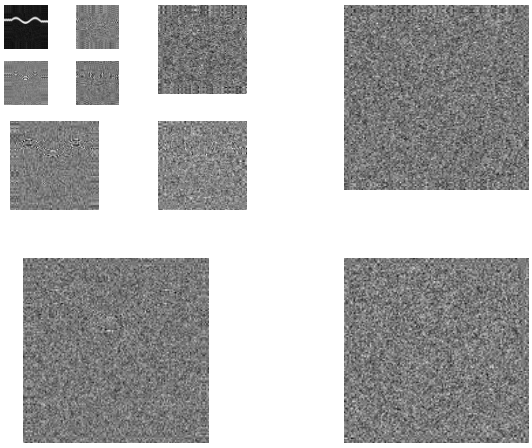

Figure 3.11: Wavelet decomposition of a noisy signal

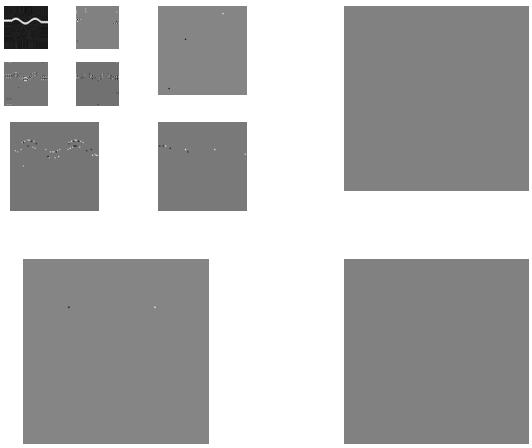

Figure 3.12: Thresholded version of the noisy signal

Figure 3.13: Filtering of 2D signal

shown. The last figure shows mainly two things: the first one is that the wavelet shrinkage can effectively be used to filter data as it can give relatively acceptable result. The second is that using the wavelet transform in an N-dimensional space can sometime fail to detect edges as it did not detected any edges in the horizontal detail nor in the diagonal images (see figure 3.12). This is reflected in the filtered reconstructed image by the "horizontal dispersion" in the signal. We will show how such dispersion can be avoided.

[7] showed that the number of contributing wavelet coefficient can be reduced by analyzing the flow in the signal and adapting the wavelet basis to it. In other word, like the time-frequency domain was adpated to find the edges, the N-dimensional time domain should be divided along the flow lines to highlight edges in this space. Nevertheless, let us first define what is meant by the flow and show how a wavelet basis can be adapted to it. The flow could be seen as the direction of the propagation of the data inside an image like the motion of an object inside it. An example of a vertically perpendicular flow is shown in figure 3.14. We can find the flow in an image by finding the solution of the minimization problem of the flow energy

$$\mathcal{E}(\underline{\tau}) = \int_{\Omega} \left| \frac{\partial (f \circ \theta)(\underline{x})}{\partial \underline{\tau}(\underline{x})} \right|^2 d\underline{x} \tag{3.12}$$

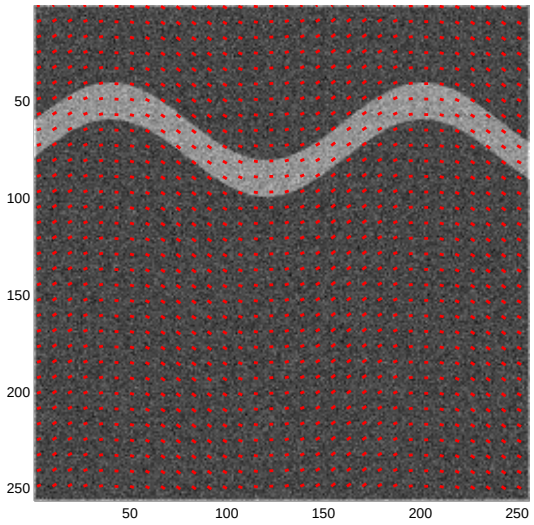

Figure 3.14: Finding the flow in an image

where $\underline{\tau}$ is the flow, Ω is the space where we want to find the flow and $\theta(\underline{x})$ is a regularization gaussian filter that has to be adjusted to the resolution at which we would like to find the flow. For the examples in this report, $\theta(\underline{x})$ was fixed to a gaussian filter of variance $\sigma = 1$.

The easiest way to adapt the wavelet basis to the flow in the image is to resample the image along the flow lines and to perform the filtering on the resampled image. However, to avoid a non-rigid transformation by resampling, we must force the flow to vary along only one dimension. Hence we can express it as $\underline{\tau}(\underline{x}) = \underline{\tau}(x_v)$ where v is the dimension along which the flow varies. As an example, we will use the noisy signal shown in figure 3.10 and assume that the flow is parrallel vertically. Such a flow is shown in figure 3.14. Therefore, using, this specific case and assuming that x_1 is the horizontal coordinate in the image, we can rewrite the flow as $\underline{\tau}(x_1) = (1, c'(x_1))$ and the minimization problem becomes

$$\mathcal{E}(\underline{\tau}) = \int_{\Omega} \left| f \circ \frac{\partial \theta}{\partial x_1}(\underline{x}) + c'(x_1) f \circ \frac{\partial \theta}{\partial x_2}(\underline{x}) \right|^2 d\underline{x} \tag{3.13}$$

The flow shown in figure 3.14 is the result of the minimization problem 3.13.

Afterwards, to resample the signal along the flow lines we define a trans-

lation for every points (x_1, x_2) as $(x_1, x_2 + c(x_1))$ where

$$c(x_1) = \int_0^{x_1} c'(u)du \tag{3.14}$$

Since the signal in only sampled at discrete point, we will have to interpolate it. In the next chapter, we will use B-Splines to interpolate a signal between its sampling points. However, in this problem, the B-Spline is not accurate enough and tends to smooth out the edges. Therefore, we will need to use a perfect interpolation method. We know that a shift of a function induce a phase shift in the fourier domain.

$$\mathcal{F}\{f(t-x)\} = e^{-i\xi x}\hat{f}(\xi) \tag{3.15}$$

Therefore, we can resample the image along the flow lines by taking the inverse transform of this phase shift.

$$f(t - c(x_1)) = \mathcal{F}^{-1}\{e^{-i\xi c(x_1)}\hat{f}(\xi)\} \tag{3.16}$$

The resampled image is shown in figure 3.15.

We can see how the resampling regularized the curve in the image. Therefore, regularizing the flow in the image could be seen as regularizing any information about edges in the image and we will see how this can improve the filtering result. Moreover, [7] proposed to add to the previous decomposition scheme one more wavelet decomposition of the horizontal detail image in the horizontal direction to take advantage of the regularity along the flow lines. This is the bandelet decomposition. The decomposition of image 3.15 is shown in figure 3.16. As we can see in this figure, there is less signal contributing coefficients and they have a greater magnitude. We can see in figure 3.17 the thresholded version of this signal with the same parameters as in section 3.2 ($\gamma = 1.0, \sigma = 0.2$). The reason for using the same parameter as in the previous section is to offer a good basis for comparison of the effect of resampling along the flow lines. However, as we can see in figure 3.17, the most important coefficients of the first level of the decomposition were all removed because of the level of thresholding. This will cause small ringing in the result around the edges of the curve as we can see in figure 3.18.

3.4 From Local Filtering to Global Filtering

So far, we defined a filtering method that could use the regularity along the flow lines to preserve the important features contained in the signal.

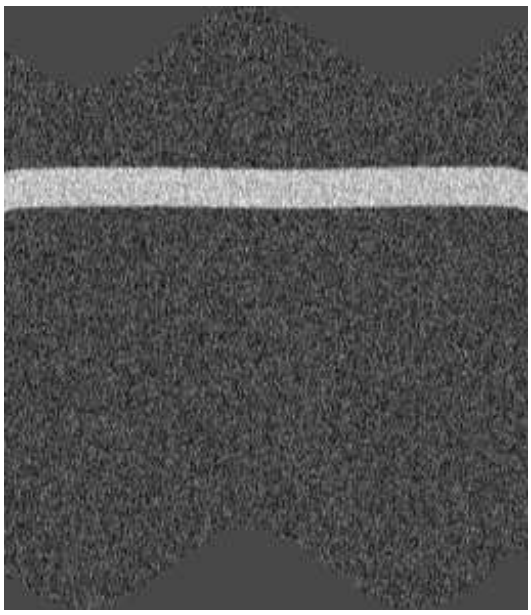

Figure 3.15: A noisy signal resampled along the flow lines

However, the hypothesis that the flow is parallel along $N-1$ dimensions for an N dimensional signal is quite strong and very few real life signals can easily meet this hypothesis. An example of such a complex image is shown in figure 3.19. Also, it is hard to tell wether it is more optimal to find a parallel flow vertically or horizontally. A method to solve for both problems has been proposed in [7]. However, in this report, we will show how a simple method can be used to adapt the filtering method to complex image. It is based on the asumption that even if the complex signal does not have parallel flow globally, it might have one locally. Therefore, we first cut small vertical band in the image (figure 3.20). An example band is shown in figure 3.20 and it now look easier to analyse. This band was resampled along the flow lines and the result in shown in figure 3.21. This shows that finding a flow along a small band succeeded. The filtering was therefore applied to the small bands. Every band was padded with data from the other bands to keep it continuous across the borders and the image was cut vertically first and then horizontally. Therefore, instead of looking only vertically or horizontally for a parallel flow, we looked both direction.

The last thing we need to find before filtering the signal is to estimate the

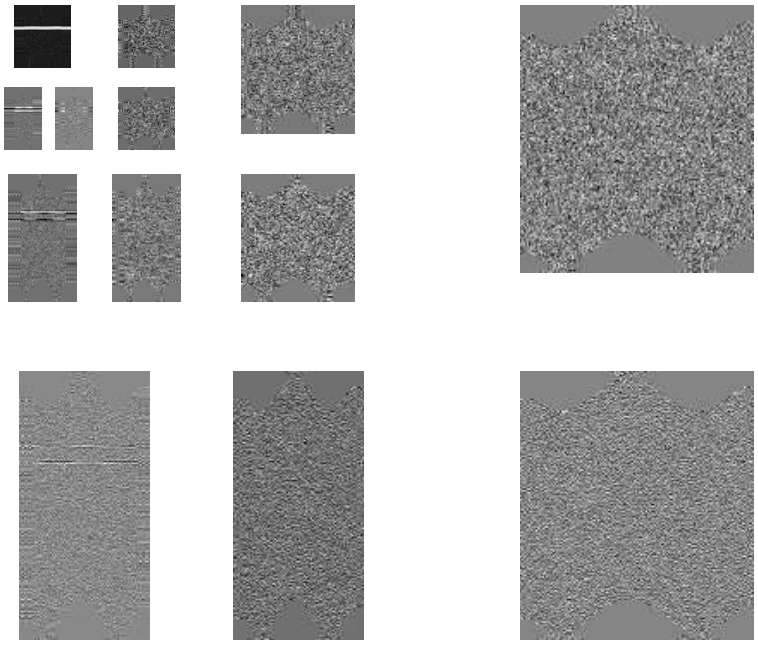

Figure 3.16: The wavelet decomposition of a resampled noisy signal with the horizontal detail image decomposed one more time along the horizontal direction

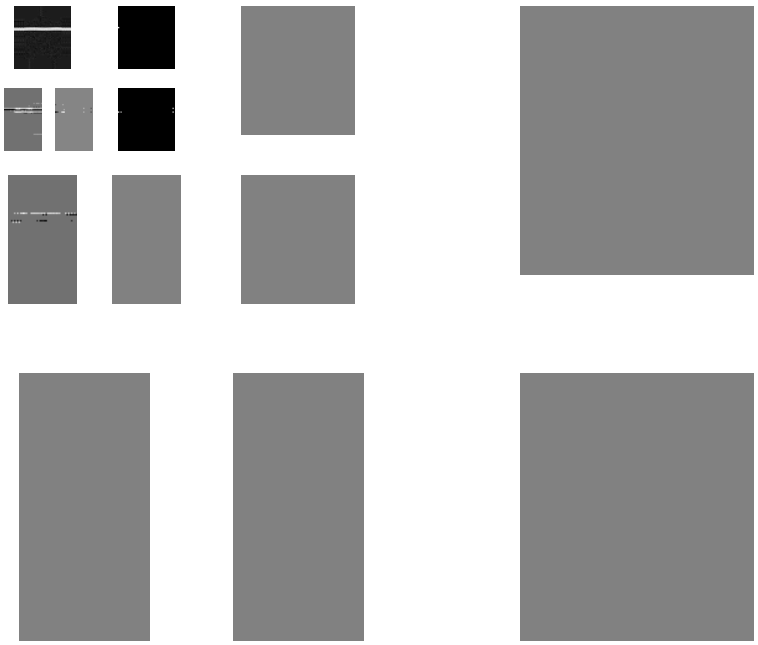

Figure 3.17: The thresholded wavelet decomposition of a resampled noisy signal

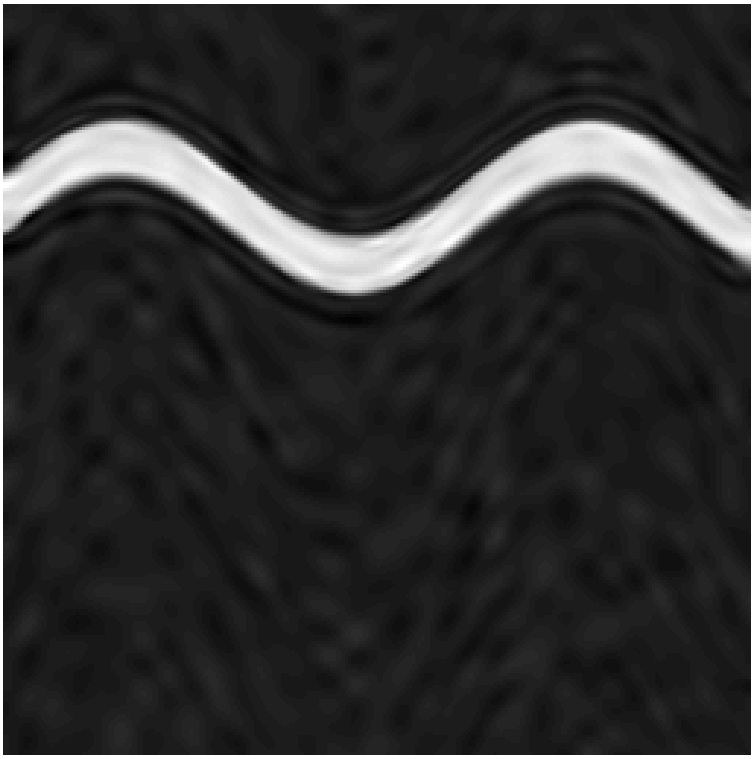

Figure 3.18: A filtered noisy signal reconstructed from a thresholded wavelet decomposition scheme with resampling along the flow lines

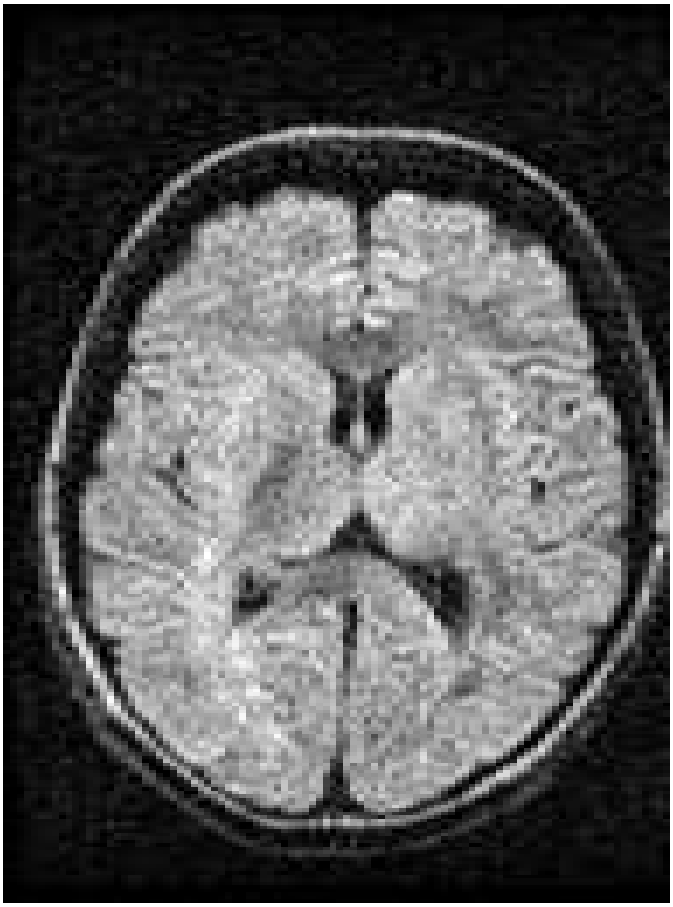

Figure 3.19: A slice of a brain using magnetic resonance imaging

Figure 3.20: A small band of the brain slice

Figure 3.21: A small band of the brain slice resampled along the flow lines.

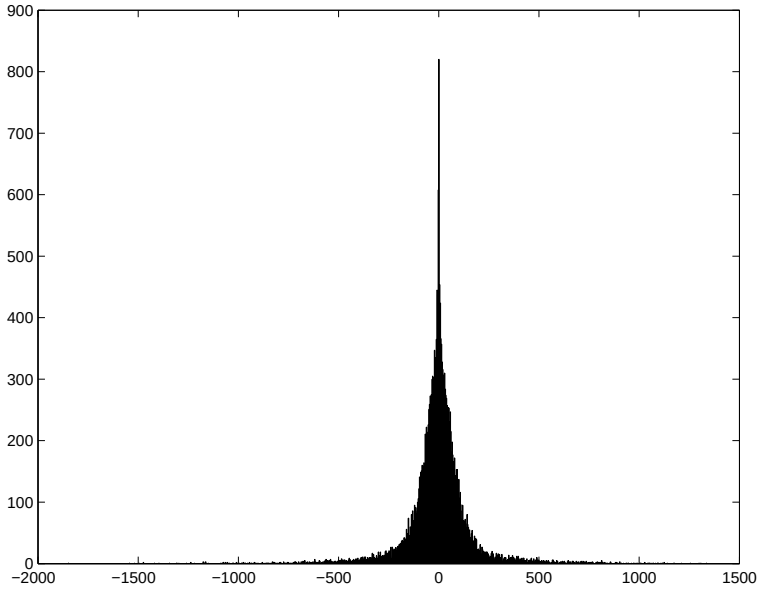

Figure 3.22: Distribution of the wavelet coefficients

noise variance. Since the noise contributes to every coefficient in the wavelet domain, we should be able to estimate it easily in this domain. We can see the distribution of the wavelet coefficients of the brain slice 3.19 in figure 3.22. This distribution shows easily that the wavelet distribution follows approximately a gaussian distribution and, we can set directly the product of $\gamma \cdot \sigma = 100.0$. This choice might seem rather arbitrary and a better way to set the threshold value could be discussed. It however, gave good result as it is possible to see it in figure 3.23. However, looking at the wavelet coefficient distribution can give a good starting point to set the thresholding value.

As a way to test the quality of the result, we can look at the difference between the original signal and the filtered signal in figure 3.24. What we should see on the difference is pure noise. However, we can see around the brain some edges and inside the brain also there is some structured data. This proves that this filtering technique can still be improved but the result is promising. A complete set of images to reconstruct a diffusion tensor slice is shown in figure 3.25. Another way to validate the result could have been to obtain a high resolution image and a low resolution image and then, trying to remove the noise in the low-resolution image and by comparing it to the

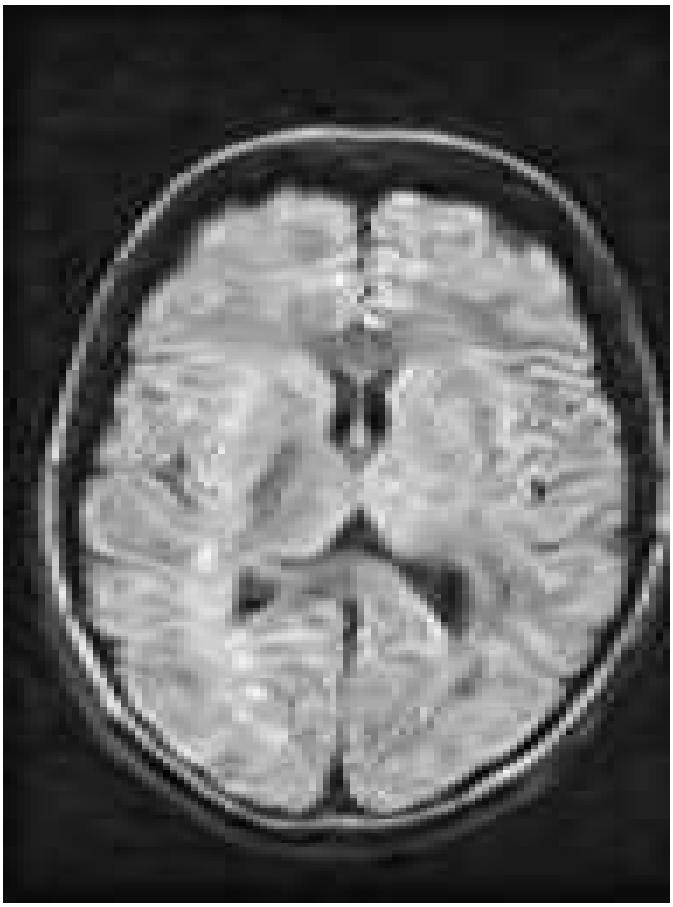

Figure 3.23: A filtered slice of a brain

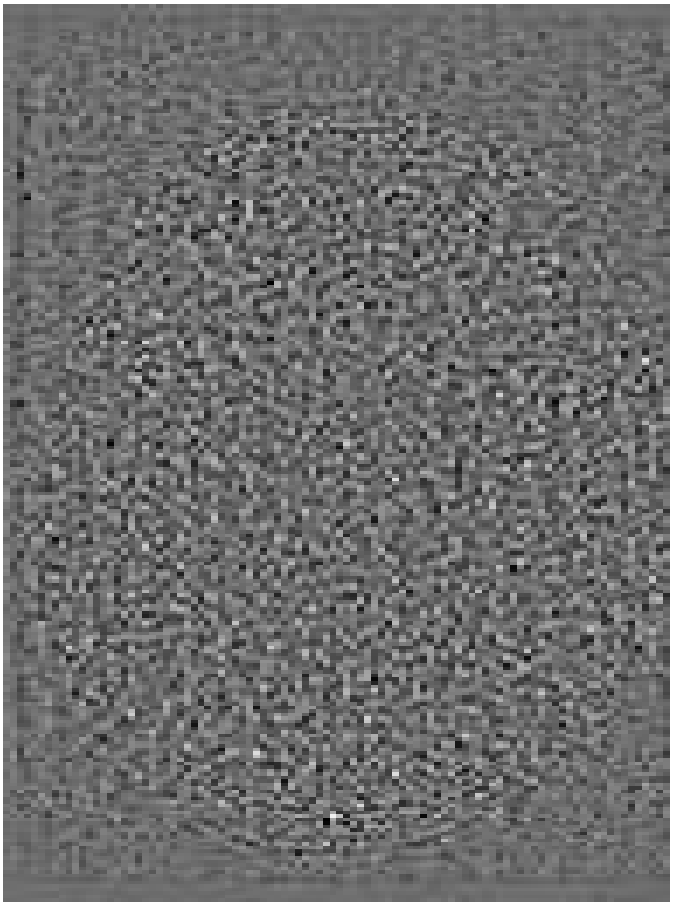

Figure 3.24: The difference between the filtered signal and the original signal

high resolution image. However, in the purpose of this report, I am unable to do such a test even if the result could have been interesting. We can still see from the result shown in figure 3.25 that the filtering method successfully removed a major part of the noise while preserving the edges.

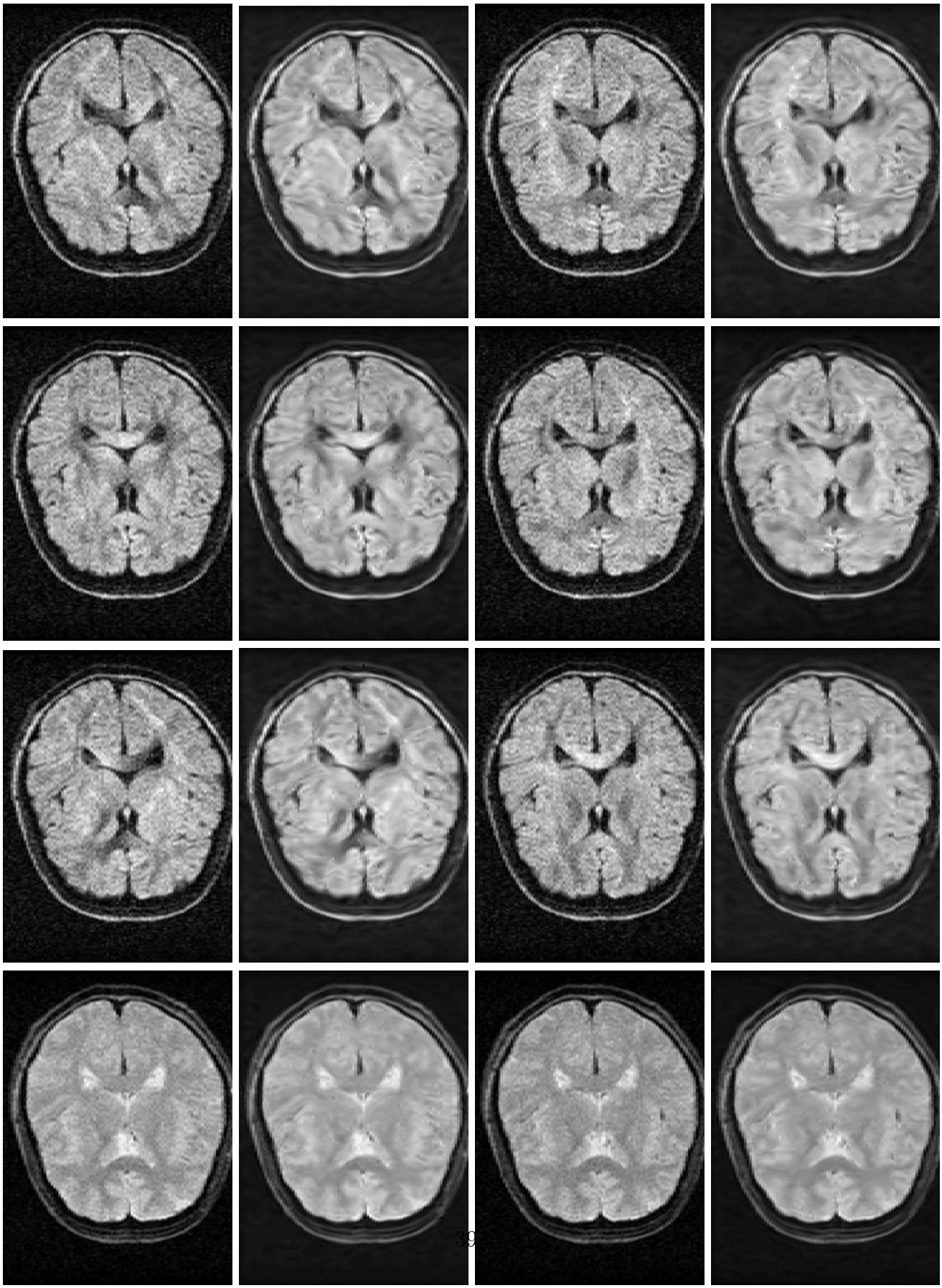

Figure 3.25: A complete filtered data set with for one tensor slice. The original data is also shown for comparison purpose. The same filtering parameter was applied to every image ($\gamma \cdot \sigma = 100$)

Chapter 4

Fiber Tractography

In the previous chapters, the problem of acquiring diffusion tensor information and a possible method to remove noise from it was discussed. As previously discussed, since the diffusion tensor imaging reveals the orientation of the connective structure of the white matter, it should be possible to visulaize it by analyzing this information. [1] proposed a method to extract the connective structure of the white matter and it is presented in this chapter along with other regularization method to regularize the reconstructed fibers.

4.1 Fiber tracking and tractography

With an imaging technique able to measure diffusion tensors inside white matter, it is possible to describe a technique to reconstruct fiber tracts from this data or to do tractography. It is important to note that in this chapter, the notation of upper indices for time dependent tensors like $\underline{\underline{F}}^i = \underline{\underline{F}}(i\Delta t)$ refers to the time elapsed and Δt will be the step size of our alogorithm.

The overall idea of the fiber tractography method as proposed in [1] is to follow at every point of the fiber the highest diffusion direction. Here is a presentation of the fiber tracking method. If we consider we have a diffusion tensor field as:

$$\underline{\underline{\varepsilon}}(\underline{x}) = \begin{pmatrix} \varepsilon_{xx}(\underline{x}) & \varepsilon_{xy}(\underline{x}) & \varepsilon_{xz}(\underline{x}) \\ \varepsilon_{xy}(\underline{x}) & \varepsilon_{yy}(\underline{x}) & \varepsilon_{yz}(\underline{x}) \\ \varepsilon_{xz}(\underline{x}) & \varepsilon_{yz}(\underline{x}) & \varepsilon_{zz}(\underline{x}) \end{pmatrix} \tag{4.1}$$

which represents a second order tensor defined over the whole tridimensional

space, we will try to reconstruct a fiber inside this tensor field starting from the point $\underline{F}^0 = \underline{F}(0)$. From this information, we will see how it is possible to extract a discrete fiber tract $\underline{F}^i = \underline{F}(i \cdot \Delta t)$ where $\underline{F}(t)$ is the analytical solution of the fiber tract. Its associated propagation direction will be noted $\underline{p}^i(\underline{x}) = \underline{p}(\underline{x}, i \cdot \Delta t)$.

The first thing we need to find in order to extract the fiber is the first propagation direction. We note $\underline{\underline{\varepsilon}}^0 = \underline{\underline{\varepsilon}}(\underline{F}^0)$. Since the fiber tract direction is the direction of maximum dffusion, we extract it from the tensor field by diagonalizing the tensor and taking the eigenvector assiciated with the highest eigenvalue. Since we are measuring diffusion, we expect all our eigenvalues to be positive and, therefore, the tensor defined should be positive semi-definite. This hypothesis must be made cautiously because depending on the filtering technique used, some negative eigenvalues can appear. If the eigenvalues are sorted $\lambda_0 > \lambda_1 > \lambda_2$, we now have a starting propagation field

$$\underline{p}^0(\underline{x}) = \left(\ p_x(\underline{x}), p_y(\underline{x}), p_z(\underline{x}) \ \right) \text{ where } \underline{\underline{\varepsilon}}(\underline{x}) \cdot \underline{p}^0(\underline{x}) = \lambda_0 \cdot \underline{p}^0(\underline{x}) \text{ and } \|\underline{p}^0(\underline{x})\|_{L_2} = 1 \tag{4.2}$$

On the first step, we don't know which way the fiber is going therefore the tractography will start in both directions $\underline{d}^0 = \underline{p}^0(\underline{F}_0, 0)$ and $\underline{d}'^0 = -\underline{p}^0(\underline{F}_0)$. For the further propagation direction, we set the eigenvector to be consistent with the previous propagation direction. Hence, we define the propagation field for $t > 0$ as

$$\underline{p}^i(\underline{x}) = \begin{cases} \underline{p}^{i-1}(\underline{x}) \text{ if } \underline{d}^{i-1} \cdot \underline{p}^{i-1}(\underline{x}) > 0 \\ -\underline{p}^{i-1}(\underline{x}) \text{ else} \end{cases} \tag{4.3}$$

This new field defines a velocity field where we set the fiber tract to be parallel to the velocity field in every point of the fiber tract. Therefore, we can relate the velocity field to the fiber tract.

$$\underline{p}(\underline{F}(t), t) = \frac{\partial \underline{F}(t)}{\partial t} \tag{4.4}$$

This differential equation defines an ordinary differential equation that gives us the position of the tract at each propagation time t.

Several method exists in the scientific litterature to solve this kind of equation and we can solve it to obtain a new tract position $\underline{F}^{i+1}$ from a previous tract position F^i. One common method used for equations that are non-stiff, that is that their accuracy depends more on the step size than on the truncation error of the method itself, is the Runge-Kutta method. As a

complete explanation of the method employed would be too long, and since it is already implemented in various libraries, we will refer the reader to [9,4] for a complete explanation of it. However, we will give a detailed explanation of its implementation to give an overall idea of how it works.

We are looking for the new velocity since it will directly give us $\underline{F}^{i+1}$ with this simple equation:

$$\begin{aligned} \underline{F}^{i+1} &= \underline{F}^{i} + \int_{i\Delta t}^{(i+1)\Delta t} \frac{\partial \underline{F}(t)}{\partial t} dt \\ &= \underline{F}^{i} + \int_{i\Delta t}^{(i+1)\Delta t} \underline{p}(\underline{F}(t), t) dt \end{aligned} \tag{4.5}$$

The different technique presented will differ in the way the integral is approximated. Hence, if we write the Taylor development of the fiber tract function around Δt we obtain:

$$\underline{F}^{i+1} = \underline{F}^{i} + \Delta t \frac{\partial \underline{F}^{i}}{\partial t} + O(\Delta t^2) \tag{4.6}$$

Therefore, we can directly approximate $\int_{i\Delta t}^{(i+1)\Delta t} \underline{p}(\underline{F}(t), t) dt \approx \Delta t \frac{\partial \underline{F}^{i}}{\partial t} = \Delta t \cdot \underline{p}^{i}(\underline{F}^{i})$. Therefore, we have

$$\underline{F}^{i+1} = \underline{F}^{i} + \Delta t \underline{p}^{i}(\underline{F}^{i}) + O(\Delta t^2) \tag{4.7}$$

This choice corresponds to the Newton method or a first order Runge-Kutta because of the $O(\Delta t^2)$. This method can give a good approximation given that the stepsize is small enough. However, we will see how it could be improved. If we continue the Taylor development, we obtain:

$$\underline{F}^{i+1} = \underline{F}^{i} + \Delta t \frac{\partial \underline{F}^{i}}{\partial t} + \frac{\Delta t^2}{2} \frac{\partial^2 \underline{F}^{i}}{\partial t^2} + O(\Delta t^3) \tag{4.8}$$

If we take the same first step but we add a second step to adjust for the new term, we obtain:

$$\underline{k}_1 = \Delta t \cdot \underline{p}^{i}(\underline{F}^{i}) \tag{4.9}$$

$$\underline{k}_2 = \Delta t \cdot \underline{p}^{i+\frac{1}{2}}(\underline{F}^{i} + \frac{\underline{k}_1}{2}) \tag{4.10}$$

Here and after, since $\underline{p}^{i+x} = \underline{p}^{i}$ for $0 \leq x < 1$, we will write $\underline{k}_2 = \Delta t \cdot \underline{p}^{i}(\underline{F}^{i} + \frac{\underline{k}_1}{2})$. If we develop this new term, we have:

$$\underline{k}_2 = \Delta t \cdot \frac{\partial \underline{F}^{i}}{\partial t} + \frac{\Delta t^2}{2} \cdot \frac{\partial^2 \underline{F}^{i}}{\partial t^2} \tag{4.11}$$

and we can obtain our new point by this formula:

$$\underline{F}^{i+1} = \underline{F}^{i} + \underline{k}_2 + O(\Delta t^3) \tag{4.12}$$

and the approximation error is $O(\Delta t^3)$. This is a second order Runge-Kutta method . We can continue to develop the Taylor serie to reduce this error even more. Hence, by taking these four steps:

$$\begin{array}{rcl} \underline{k}_1 & = & \Delta t \cdot \underline{p}^i(\underline{F}^i) \\ \underline{k}_2 & = & \Delta t \cdot \underline{p}^i(\underline{F}^i + \frac{\underline{k}_1}{2}) \\ \underline{k}_3 & = & \Delta t \cdot \underline{p}^i(\underline{F}^i + \frac{\underline{k}_2}{2}) \\ \underline{k}_4 & = & \Delta t \cdot \underline{p}^i(\underline{F}^i + \underline{k}_3) \\ \underline{F}^{i+1} & = & \underline{F}^i + \frac{1}{6}\underline{k}_1 + \frac{1}{3}\underline{k}_2 + \frac{1}{3}\underline{k}_3 + \frac{1}{6}\underline{k}_4 + O(\Delta t^5) \end{array} \tag{4.13}$$

the approximation error will be $O(\Delta t^5)$ and this method is called a fourth order Runge-Kutta.

However, computing this coefficients for evey stepsize can take a non-negilgible amount of time and as it is relatively easy to compute the fourth order Runge-Kutta, computing every order above the fourth requires at least more step than that order. This means that at least six steps are required to compute the fifth order and that there is a computational limit to the order of the method. Since the precision of the method depends on Δt, if it was possible to modify the stepsize to keep the approximation error $O(\Delta t^{n+1})$, below a certain value Err_c, it would allow to take larger steps where the approximation error is small enough. Hence we will have to monitor this approximation error by a more accurate method. Therefore, while computing $\underline{F}^{i+1}$ with a fourth order method, we will also compute it with a fifth order method and we will assume the approximation error as being

$$Err_i \approx \|\underline{F}_5^{i+1} - \underline{F}_4^{i+1}\|_{L_2} \tag{4.14}$$

Since we expect the fifth order method to be more accurate, we will simply assume that $\underline{F}^{i+1} \approx \underline{F}_5^{i+1}$. Moreover since we do not want to calculate to much coefficients ($k_{1,2\ldots}$), we will do it in such a way that both the fourth order and the fifth order may be computed from those coefficients. Hence, if

we express the different steps as

$$\begin{array}{rcl} \underline{k}_1 & = & \Delta t \cdot \underline{p}^i(\underline{F}^i) \\ \underline{k}_2 & = & \Delta t \cdot \underline{p}^i(\underline{F}^i + b_{21} \cdot \underline{k}_1) \\ \underline{k}_3 & = & \Delta t \cdot \underline{p}^i(\underline{F}^i + b_{31} \cdot \underline{k}_1 + b_{32} \cdot \underline{k}_2) \\ & \cdots & \\ \underline{k}_6 & = & \Delta t \cdot \underline{p}^i(\underline{F}^i + b_{61} \cdot \underline{k}_1 + b_{62} \cdot \underline{k}_2 + b_{63} \cdot \underline{k}_3 + b_{64} \cdot \underline{k}_4 + b_{65} \cdot \underline{k}_5) \\ \underline{F}^{i+1} & \approx & \underline{F}^i + c_1\underline{k}_1 + c_2\underline{k}_2 + c_3\underline{k}_3 + c_4\underline{k}_4 + c_5\underline{k}_5 + c_6\underline{k}_6 \end{array} \tag{4.15}$$

the coefficients for the fifth order and fourth order Runge-Kutta method are:

Coefficients of the 4^{th} and 5^{th}order Runge-Kutta method

i	b_{ij}					$c_i^*(4^{th})$	$c_i(5^{th})$
1						$\frac{2825}{27648}$	$\frac{37}{378}$
2	$\frac{1}{5}$					0	0
3	$\frac{3}{40}$	$\frac{9}{40}$				$\frac{18575}{48384}$	$\frac{250}{621}$
4	$\frac{3}{10}$	$\frac{-9}{10}$	$\frac{6}{5}$			$\frac{13525}{55296}$	$\frac{125}{594}$
5	$\frac{-11}{54}$	$\frac{5}{2}$	$\frac{-70}{27}$	$\frac{35}{27}$		$\frac{277}{14436}$	0
6	$\frac{1631}{55296}$	$\frac{175}{512}$	$\frac{575}{13824}$	$\frac{44275}{110592}$	$\frac{253}{4096}$	$\frac{1}{4}$	$\frac{512}{1771}$
$j=$	1	2	3	4	5		

(4.16)

The method used to compute the next step size is simple. We can directly note the error made at a given step as

$$Err_i \approx \|\underline{F}_5^{i+1} - \underline{F}_4^{i+1}\|_{L_2} = \|\sum_{i=1}^{6}(c_i^* - c_i) \cdot \underline{k}_i\|_{L_2} \tag{4.17}$$

then

$$\Delta t_{i+1} = \Delta t_i \cdot |\frac{Err_c}{Err_i}|^{0.2} \tag{4.18}$$

which directly gives the step size to take if we have to recompute the iteration because the step was too large or the step size we can take for the next iteration.

4.2 Continuous interpolation of the tensor field

In the previous section, we assumed that we have a tensor field which is defined continuously over the whole space. As with every discrete signal, this is

usually not the case and some interpolation between each value must be done. Nyquist showed that every signal $S(t)$ with is Fourier transform $\hat{S}(\xi)$ that was sampled at a sample rate $\frac{1}{T}$ from a discrete signal can be reconstructed perfectly to the original signal if it had a compact support in the frequency domain which is smaller than half of the sampling frequency (see [16] for a demonstration). In other word, $\hat{S}(\xi) = 0$ if $\xi > \frac{1}{2T}$. If not, a spectral folding will occur for all frequencies $\xi > \frac{1}{2T}$. However, such reconstruction uses sine cardinals which has a slow numerical convergence. Hence, if it was possible to replace the sine cardinals basis by another basis which add a smaller support, this could improve the computation time. We will therefore try to finding a polynomial which is piecewise defined and that best fits the signal we are trying to interpolate. [17] showed that such a polynomial that gives the least error compared to the full nyquist reconstruction for a given region of support and has the highest order of continuity is the B-Spline. [1] also proposed it to the special case of tensor data. This interpolation method is quite simple and consists of convoluating the signal with a box window. The box window is convolved once for a first order polynomial, twice for a second order, etc. as we can see in figure 4.2.

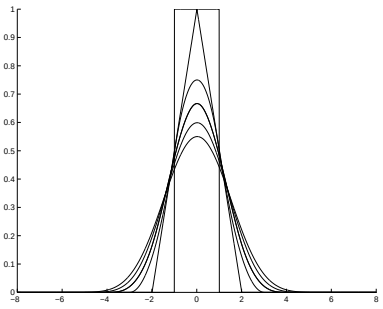

Figure 4.1: BSpline coefficient for different orders

Moreover, in order to make good use of the precision of the tractography method, some restrictions will have to be made on the order of the polynomial used depending on the order of the tractography. Hence, if for instance the tractography method is of the 4th order, meaning that the error is $O(\Delta t^5)$, we will have to make sure the polynomial used has enough non-null derivatives

to make good use of the precision of the method. Therefore, in this case, the polynomial would have to be at least of the third order in order to have a non-null third derivative. In this particular case, the reconstruction would be perfect except where the third derivative in not continous, e.g. between voxels. A practical way to avoid this problem is to stop between voxels to allow tractography to continue in the next voxel. Another way is to use a polynomial one degree higher so every term in the Taylor development is continuous and the tractography can be performed across voxels. This method solves the problem of finding the point at the boundary and the adaptative step size scheme described in the previous section helps to keep the error below a certain specified value. Therefore, for a fifth order Runge-Kutta, the interpolating B-Spline will have to be a fifth order also.

Nevertheless, relating the B-Spline order to the order of the Runge-Kutta method can give another indication on the limitation of this method. As discussed in the previous chapter, the B-Spline tends to smooth the edges because it is associated to a low pass filtering. Even if it helps to extract more regular and continuous fibers, it does not mean that they are more accurate. It could also have the effect of connecting two regions that were separated by smoothing their edges. Some concerns about the magnetic resonance imaging and the B-Spline approximation will also be discussed in the next chapter.

4.3 Termination problems in tractography and tensor regularization

Diffusion tensor imaging is very different from any kind of scalar imaging in a lot of situations and brings its own kind of problem that has to be adressed. In the previous section, we assumed that the main and only diffusion direction is given by the direction of the eigenvector associated with the highest eigenvalue. This is true in the ideal case of anisotropic diffusion but, as the tensor goes farther from this case, some restriction must be made. When eigenvalues are close to each others, even the smallest amount of noise can affect greatly the direction of their eigenvectors. Such case of close eigenvalues can occur in sheets between membranes, when two fiber tracts cross each others or in the most general case of noisy data. Therefore, in order to enhance the tractography and deal with the case of more uncertain direction [3] proposed to insert a bias towards the previous direction in the tractography.

As inserting this bias will make the tractography slightly go outward when following a curve, we will have to insert it only where the tensor requires it. Hence a measure of the anisotropy of the tensor must be introduced. Several measures exist in the scientific litterature, and the one we will use is the fractional anisotropy as proposed in [2]. It can be expressed easily from the eigenvalues of the tensor

$$\nu = \sqrt{\frac{(\lambda_1 - \lambda_2)^2 + (\lambda_1 - \lambda_3)^2 + (\lambda_2 - \lambda_3)^2}{2 \cdot (\lambda_1^2 + \lambda_2^2 + \lambda_3^2)}} \tag{4.19}$$

However, diagonalizing a tensor every time we want to compute the fractional anisotropy can be time consumming. Therefore, we will rewrite this formula to

$$\nu(\underline{x}) = \frac{\sqrt{3}}{\sqrt{2}} \frac{\|\underline{\underline{\varepsilon}}(\underline{x}) - \frac{1}{3} trace(\underline{\underline{\varepsilon}}(\underline{x}))\underline{\underline{I}}\|_F}{\|\underline{\underline{\varepsilon}}(\underline{x})\|_F} \text{ where } \|\cdot\|_F \text{ is the Frobenius norm} \tag{4.20}$$

As fractional anisotropy can be used as a good estimator to threshold white matter from other tissue as we can see in figure 4.3, we will stop the tractography whenever it goes below a critical value ν_l. As stated above, inserting a bias makes the tractography go outward so no correction will be made if the fractional anisotropy is over another critical value ν_h. We will therefore modify the tensor field with a bias towards the previous direction:

$$\underline{\underline{\varepsilon}}^i(\underline{x}) = \begin{cases} \underline{\underline{\varepsilon}}(\underline{x}) + C_b \cdot \frac{\nu_h - \nu(\underline{x})}{\nu_h - \nu_l} \cdot \underline{d}^i \otimes \underline{d}^i \text{ if } \nu_l < \nu(\underline{x}) < \nu_h \\ \underline{\underline{\varepsilon}}(\underline{x}) \text{ else} \end{cases} \tag{4.21}$$

where C_b is the magnitude of the correction bias. This could be seen as inserting inertia in the propagation of the fiber.

The last regularization of the tractography performed is designed to prevent a tract from a different tract to follow a new one without stopping it with the fractional anisotropy criteria. Hence, if the radius of curvature of the tract is too small meaning that we may have stopped to follow the initial direction and we are now following a different tract, we end the tractography.

Using all the above regularization method, the tractography method described in this chapter should be able to extract smooth and continuous fiber tracts. However, as it was previously stated, even if the fibers displayed on the screen is smooth, it does not mean it is accurate. Therefore, other regularization scheme could be implemented to assure that the displayed fiber is

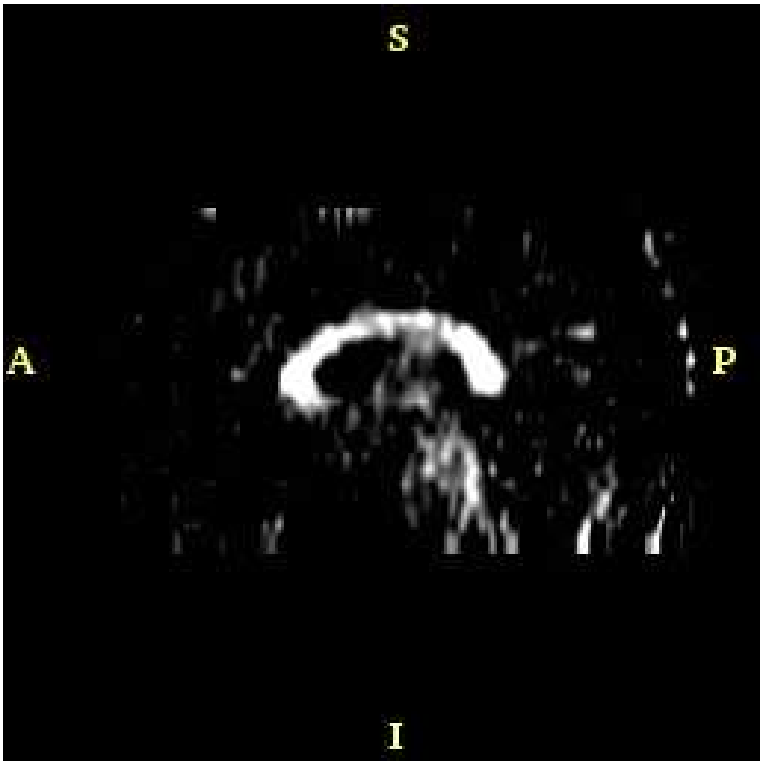

Figure 4.2: Fractional anisotropy of a DTMRI slice

accurate to the measured diffusion tensors. For exemple, by assuming that the tensor is the local representation of a manifold, a fiber tract would be defined by the fastest path from point A to point B (see [12]). Another method could be to choose fibers going from one region of interest to another region of interest where the region of interest would be defined using histological data (see [11]). This give an example of the possible improvement of the fiber tractography method. Comparing the tractography method these two other method could be interesting, and could be an interesting way to compare how this method perform.

Chapter 5

Implementation

This report covers various problems like filtering data and doing tractography in a diffusion tensor field. Implementation of the solutions to these problems requires many libraries and can be done in various environment. In this chapter, we will see how the implementations of the solution to the two problems were done.

5.1 The 3D-Slicer

The main software developed and used here at the Surgical Planning Laboratory is the 3D-Slicer (<www.slicer.org>). This software integrates several facets of medical imaging such as image filtering, extracting interesting features inside an image (segmentation), aligning different data sets together (registration) and volume visualization. I implemented the tractography module in this software. For cross-platform purposes, it is written in the Tcl/Tk scripting language (<www.tcltk.org>). In fact, since the development environment is on various platforms (I worked mainly on Solaris but it is developped also under Windows, Linux, Mac) and on different development site (collaboration exist with universities and research laboratories in Germany, Sweden, Poland ...) the developper of the 3D-Slicer needed to totally abstract the operating system from the programmation. The use of a scripting language makes it possible as interpreters exists on various architectures. It also uses a cross-platform visualization environment named the Visualization Toolkit (<www.vtk.org>). This toolkit consists of a set of libraries and classes written in c++ to give the developer a level of abstraction over

other graphical libraries like OpenGL. Since it needs to be cross-platform, the c++ classes are compiled with cmake (<www.cmake.org>) which is a cross-platform make program. It also automatically wraps Tcl/Tk around those classes which makes it possible to create dynamically loadable libraries from the vtk classes and use them directly inside the Tcl/Tk scripts as it is needed.

From an architectural point of view, since several version of the slicer are maintained (releases, developing versions and other versions using specific modules) the slicer is developed in a modular point of view. There is a base module which consists of the main program of the slicer and several other modules written by different research laboratories. This modularity helps to maintain the overall program working even if a specific module does not. Every module is written in Tcl/Tk and may have a dynamically loadable library of vtk classes and it is compiled through cmake. I worked primarily on three different modules to implement the tractography method. The first one is the Tensor Utility module developed and maintain by the Laboratory of Mathematics in Imaging (<lmi.bwh.harvard.edu>) which is the laboratory I was working in. This module contains several classes which deals with tensor data, tensor mathematics and imaging. I put all the classes I developed inside this module except one that needed a special library to work. Since this module does not have a loadable graphical interface, I created the graphical interface of the tractography method which is the DT-MRI module also developed at the LMI. As its name says it, this module groups all the tools to work on the diffusion tensor magnetic resonance imaging. It already contained a tractography tool and it was natural to put it there. The last module used is the ITK module. ITK stands for Insight Segmentation and Registration ToolKit. It is a library that contains several classes to work on medical images, data and signal analysis. Therefore, it offers numerous filter, segmentation and registration classes. Even if this library is not officially required by the 3D-Slicer to compile, many modules use it through one way or another. I used the BSpline filter class of this library but, since neither the DTMRI or the Tensor Util modules uses ITK directly, I had to put it in the ITK module which is mainly a VTK interface to the ITK filters.

5.2 Implementation of the filtering method

Since the DT-MRI data is saved in a known file format, several functions to read it into other software like Matlab were written to be able to test and implement new features. Therefore, it is not necessary to implement directly in the 3D Slicer the algorithms that would require libraries that are not available as part of the developing environment of the 3D Slicer. This is the case for the filtering technique where no wavelet transform is available inside ITK. However, a wavelet toolbox is available inside matlab and many other features, allowing more effective testing, since no compiling is needed to run a Matlab function. Moreover, since the data need to be filtered only once, it could be saved to disk once and read back into the slicer for later use. Therefore, Matlab was a good choice to implement the filtering technique with the time constraint of this internship.

5.3 The Tractography Module

As described above, a tractography tool was already available in the DT-MRI module. However, this tool used a second order Runge-Kutta method with only a linear interpolation of the tensor only and did not allow any higher order to be used. It was also using the eigenvalue of the propagation direction as a termination value for the tractography which might be really noisy. Moreover, finding a good termination eigenvalue for a tensor might be very difficult as it might be different depending on the trace of the tensor and may lie on a range around 10^{-6}. Therefore, too much analysis of the data was needed to set a good termination eigenvalue and another termination value was needed. Since all this was written inside a class available as part of the main vtk library, I had to write new classes to perform Tractography. We can see on figure 5.1 the collaboration diagram of the classes I implemented. The "vtk" prefix is omitted for purpose of clarity.

As we can see on figure 5.1, there are two parts in this module. The first part is the preprocessing of the B-Spline coefficient from the tensor data before starting the Tractography. Since the ITK class could not directly handle tensor data, I had to separate it into six differents scalar components. Afterward, it could be filtered by the ITK class named BSplineDecompositionImageFilter. The result is stored in a class named vtkBSplineInterpolateImageFunction. This class can be questionned about the BSpline value

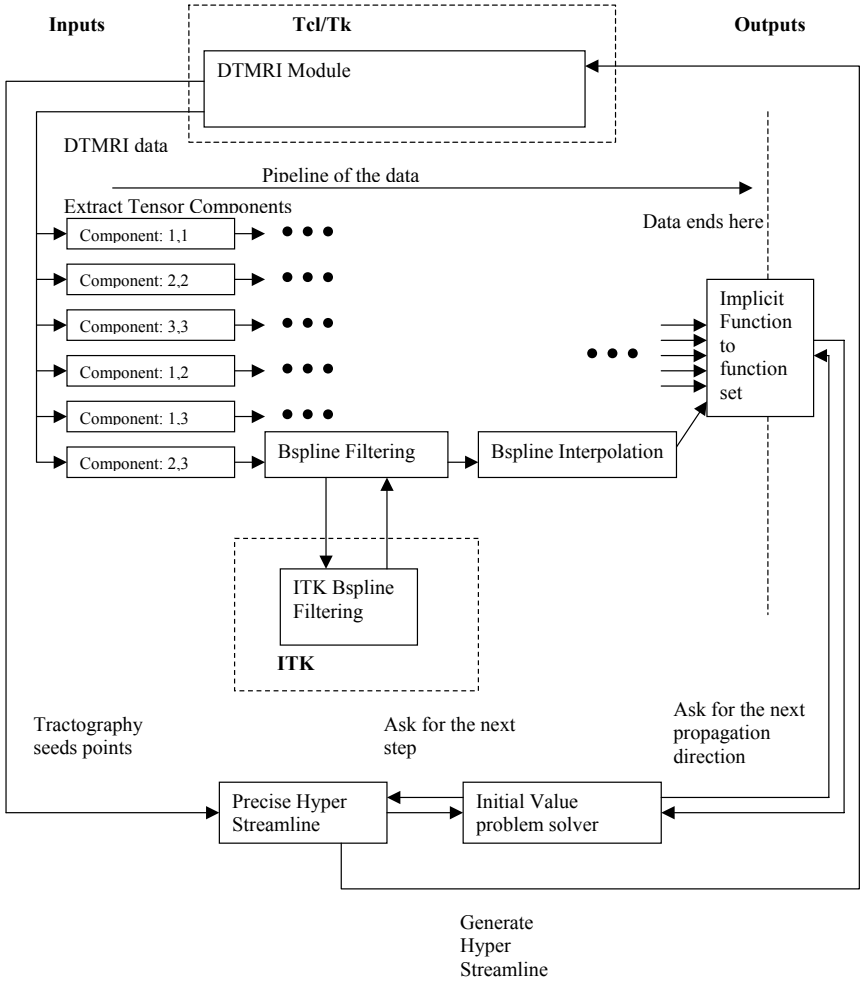

Figure 5.1: Class Diagram of the Tractography Module

at any point x and it will give the result as a scalar function. The vtkImplicitFunctionToFunctionSet does this for every scalar value and is in charge of reconstructing the tensor data, computing its fractional anisotropy and outputing the propagation direction at this given point x.

The second part of the module is the one in charge of building a stream line, or a fiber tract, from a starting point x. In the chapter on tractography, we defined a way to solve an ordinary differential equation like 4.4. The method described is implemented in a VTK class named vtkRungeKutta45 which inherits from a more general class to solve ordinary differential equations named vtkInitialValueProblemSolver. Therefore, I wrote a class that could use this class to solve the problem 4.4 and which would stop whenever a termination value is reached.

We can see the graphical interface of the DTMRI module on figure 5.2. It shows a tensor volume with glyphs showing the principal diffusion direction. The color coding corresponds to the fractional anisotropy where red is the highest and blue is the lowest. To do tractography, the user must first load a volume inside the slicer. Then he must convert it from scalar data to tensor data. If he specified that he wants to use a B-Spline interpolation, the B-Spline coefficients are computed at this time. As stated in the previous chapter, since drawing continuous fibers does not automatically mean drawing accurate fibers, many options was included. Therefore, the user can choose the order of the tractography method and the order of the B-Spline. Then he can set the tractography option. Those options are shown on the left window on figure 5.2. He can start a stream line by pointing with the mouse on the picture where he wants to start it and by hitting the 's' button. However, segmenting the whole white matter by doing tractography can get quite long. Therefore, some tools to do automatic tractography were included inside the 3D Slicer. The first thing to do is to define a region of interest, which specifies the fibers we would like to extract. As an example, we will segment the corpus callosum. Afterward, the user can easily select start autotracking which seeds a stream line in every voxel of the region of interest. A result of this tractography can be seen on picture 5.3.

There exist different way of improving this implementation. The first one might be to use a different order for the BSpline along the different dimensions. For example, in the example used in figure 5.3, the volume dimension were 256 x 256 x 24 and the voxel size was 0.859375 x 0.859375 x 4.0 mm. Therefore, the first two dimensions have a much more higher resolution than the third dimension. Using a fifth order B-Spline along the first two dimen-

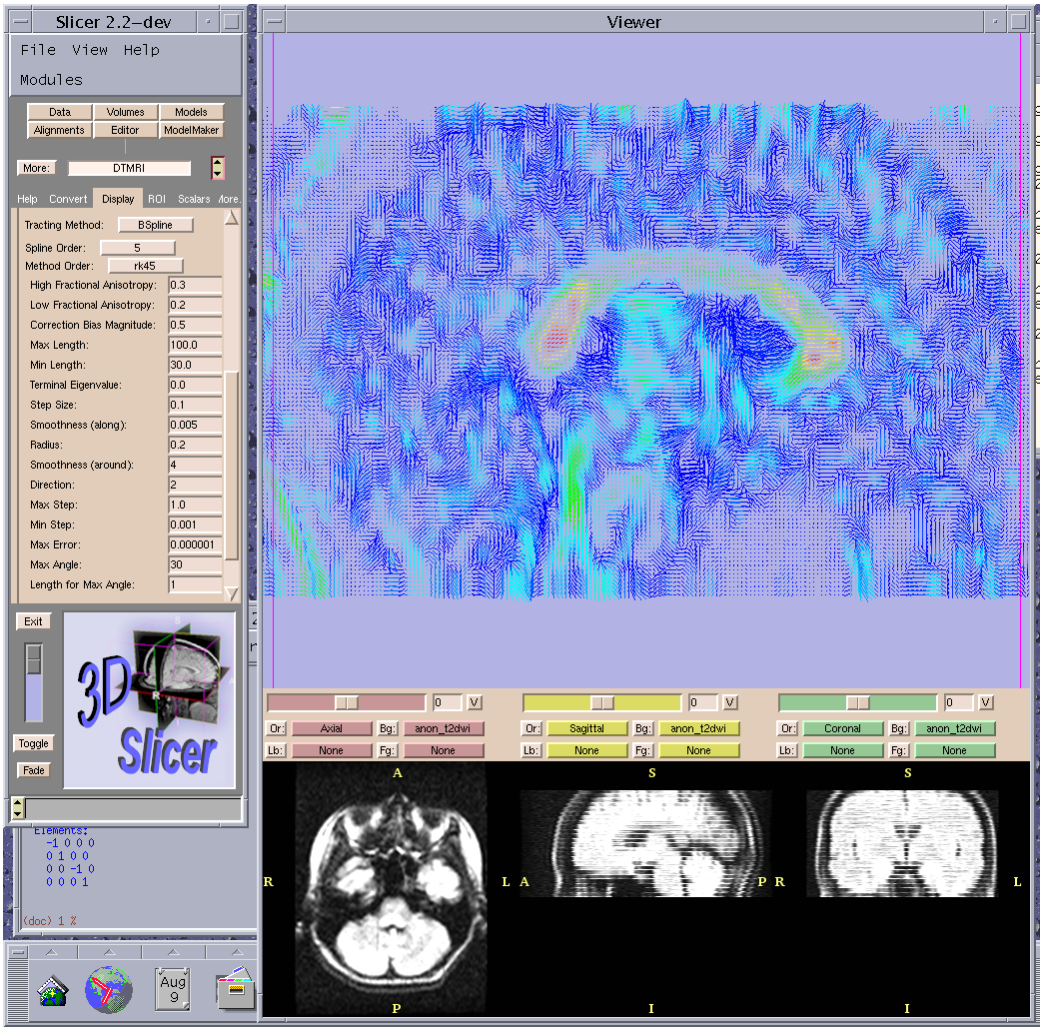

Figure 5.2: Graphical Interface of the DTMRI module

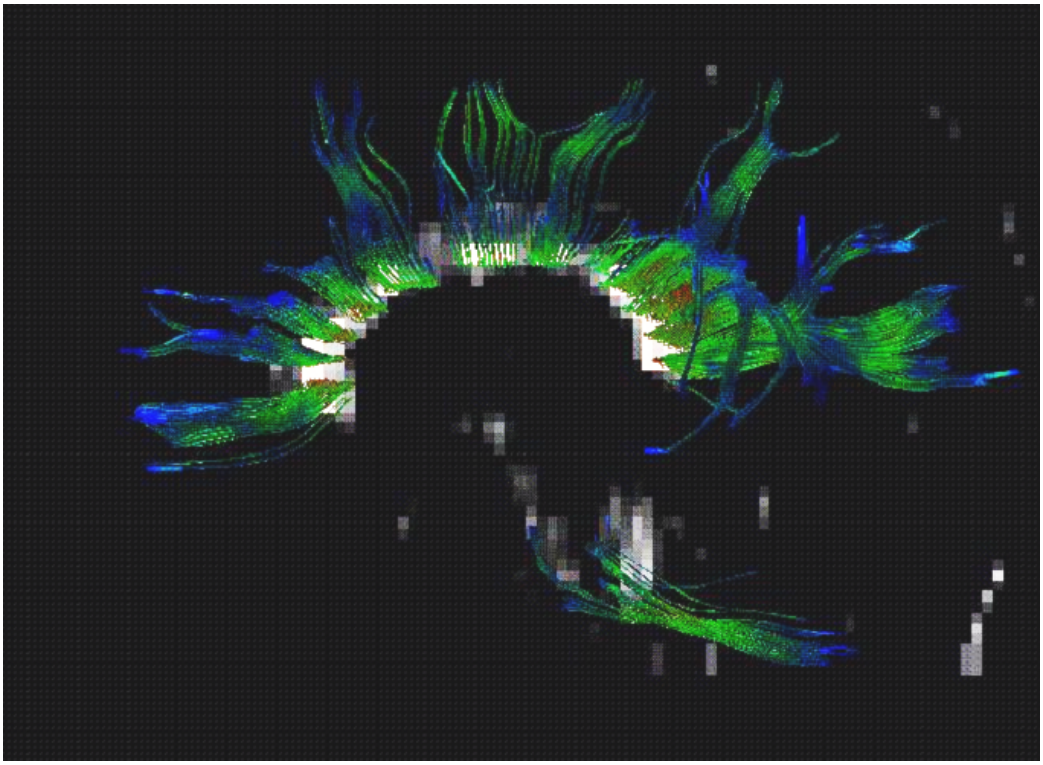

Figure 5.3: A reconstruction of the corpus callossum using fiber tractography

sions could still be accurate but using it along the third dimension mean that the region of support of the BSpline kernel will be $5 \cdot 4.0$ mm $= 20.0$ mm wide which is quite large. The approximation could be improved by using only a linear interpolation or first order B-Spline along this dimension. The overall kernel size of the B-Spline approximation would have been of $(5 \cdot 0.859375 = 4.296875)$ x 4.296875 x 4.0 mm. However, the ITK class do not allow to use different order for the BSpline along particular dimension. It could have been an interesting feature to develop.

Chapter 6

Discussion

More than studying the regularization of tractography, one of the goals of this internship was to apply my knowledge inside a research environment.

As for the first part of this goal, I think this internship was beneficial on many points. Of course, I had the chance to study a problem that had to be treated on an applied mathematic and a computer science point of view but I learned a lot from the environment I was working in. In my courses at the Ecole Polytechnique, programs were at most a few thousand lines of codes and I never had the chance to work on a project which involved a large number of developper. Hence, here I had the chance to work on a much more integrated environment, on a much bigger software where nobody except maybe a few people knows every detail of the implementation of it. Even to start working in this environment, I had to learn how to use cmake to compile my own version of the software, the basics of Tcl/Tk and VTK. Luckily, those libraries are all open source highly documented libraries with examples for every class provided. Nevertheless, understanding how everything works and just writing my first "hello world" class which just write hello world in Tcl/Tk shell took some time as I had to modify the cmakefile, understand what is needed in a vtk class. And, even if a lot of documentation is provided, I had to check several times what is really done inside a class during long debugging sessions which made me appreciate the fact that I have access to the source code of the library. I also had the chance to understand the difficulty to write software for different platforms at the same time when I received an e-mail form one of the main developers saying that my code was compiling under solaris but not under windows. I understand now the choice of using vtk, tcl/tk and cmake to help building a cross-platform environment. Complying

with there standards was sometimes painful but as their requirements insure a minimum of portability across different windowing systems, it is sure worth the hassle.

Moreover being cross platform, using VTK as the visualization library provided interesting feature like its garbage collecting capabilities. I did not had the chance to use it, but it also provide easy threading functionality where using thread or not is only defined as a flag when loading a module. However, all those features come at a cost. Since it is all implemented using the polymorphism capabilities of c++, it is very easy to use but you never know which class you are really working on. I had a lot of difficulty to figure out how the data is represented in memory and this brought a lot of difficulties when trying to create a pipeline for the data from VTK to ITK. However, I understood the difficulty of writing software which his much more difficult than writing a little program that implements a few thousand lines of code.

More than working on an interesting project, working in a team of researchers was also really interesting. Since I plan to continue during my fourth year of study of the Ecole Polytechnique in biomedical imaging, this internship was a great opportunity to discover the field and have an introduction to its major problems and paradigm. Hence, I had the chance to attend conferences given by researchers about their work in this subject and, just by speaking with people around me, obtain a good introduction about what could be done inside the different subfields. Since I planned to make my internship last more than the recommended three months, I was also given the chance to explore a problem in a more complex way. Finally, the Surgical Planning Laboratories offers facilities where developers of the 3D-slicer software work with those who use it. Therefore, I also spoke with medical students performing research and using the software that was developped. I could then see what their point of view is and learn more about the medical research. The working environment therefore offered a stimulating and interesting research environment.

Chapter 7

Conclusion

In this internship, I had the chance to study the regularization of fiber tractography, an application of DTI, which permit a full reconstruction of the inner structure of the white matter by following fiber tracts in a water diffusion tensor map. I implemented a fully working fiber tractography module with many regularization schemes ranging from noise filtering to augmenting the continuity of the reconstructed fibers. I also implemented a novel noise filtering technique for regularizing information in magnetic resonance signals. In the end, the overall tractography module helped to extract continuous fibers from the white matter and allowed full reconstruction of the connective architecture of the human brain.

This internship also offered me a great opportunity to have an overall introduction to the biomedical imaging field and to see what are the opportunities in this field. The module I developped gave me a good idea of what it is possible to do using in-vivo imaging instruments and what kind of information could be extracted. However, a lot of question is still left unanswered in this report before the fiber tractography module could be used as a diagnostic tool on real patient. For example, it could be interesting to find a criteria on when we can "merge" two fiber tracts tips if they represent two fibers which seems to be connected. It could be interesting to see if it was possible to measure the connection strength between two different region inside the white matter instead of just counting the number of fiber tract we were able to draw from those two regions. These are example of interesting problems that could help improve the fiber tractography module. It shows of how exciting is the problem that was discussed in this report and how promising are every develpment inside this field.

Bibliography

[1] Peter J. Basser, Sinisa Pajevic, Carlo Pierpaoli, Jeffrey Duda, and Akram Aldroubi. In vivo fiber tractography using dt-mri data. *Magnetic Resonance in Medicine*, 44:625–632, 2000.

[2] Peter J. Basser and Carlo Pierpaoli. Microstructural and physiological features of tissues elucidated by quantitative-diffusion-tensor mri. *Journal of Magnetic Resonance*, 111:209–219, 1996.

[3] M. Björnemo, A. Brun, R. Kikinis, and C.-F. Westin. Regularized stochastic white matter tractography using diffusion tensor mri. In *Fifth International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'02)*, pages 435–442, Tokyo, Japan, 2002.

[4] J. R. Cash and Alan H. Karp. A variable order Runge-Kutta method for initial value problems with rapidly varying right-hand sides. *ACM Transactions on Mathematical Software*, 16(3):201–222, September 1990.

[5] Le Bihan D, Breton E, Lallemand D, Grenier P, Cabanis E, and Laval-Jeantet M. MR imaging of intravoxel incoherent motions: application to diffusion and perfusion in neurologic disorders. *Radiology*, 161(2):401–407, 1986.

[6] D. L. Donoho and I. M. Johnstone. Ideal denoising in an orthonormal basis chosen from a library of bases. Technical report, Stanford University, 1994.

[7] LePennec E. and Mallat S. Sparse geometrical image approximation with bandelets. *IEEE transation on Signal Processing*, 2003. accepted.

[8] Stejskal E.O. and Tanner J.E. Spin diffusion measurements: spin echoes in the presence of a time-dependent field gradient. *J. Chem. Phys.*, 42:288–292, 1965.

[9] Press W. H. et al. Numerical recipes in fortran, second edition. *Cambridge University Press*, 1990.

[10] Gudbjartsson H, Maier S, Mulkern R, Mórocz I.A., Patz S, and Jolesz F. Line scan diffusion imaging. *Magnetic Resonance Journal*, 36:509–516, 1996.

[11] Catani M., Howard RJ, Pajevic S, and Jones DK. Virtual in vivo interactive dissection of white matter fasciculi in the human brain. *Neuroimage*, 17:77–94, 2002.

[12] L. O'Donnell, S. Haker, and C.-F. Westin. New approaches to estimation of white matter connectivity in diffusion tensor mri: Elliptic pdes and geodesics in a tensor-warped space. In *Fifth International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'02)*, pages 459–466, Tokyo, Japan, 2002.

[13] Sinisa Pajevic and Peter J. Basser. Parametric and non-parametric statistical analysis of dt-mri data. *Journal of Magnetic Resonance*, 161:1–14, 2003.

[14] Basser PJ, Mattiello J, and LeBihan D. MR diffusion tensor spectroscopy and imaging. *Biophys J*, 66(1):259–267, 1994.

[15] Maier S and Gudbjartsson H. Line scan diffusion imaging. *USA patent 5,786692*, 1998.

[16] Mallat S. A wavelet tour of signal processing, 2nd edition. *Academic Press*, 1999.

[17] M. Unser. Splines: A perfect fit for signal and image processing. *IEEE Signal Processing Magazine*, 16(6):22–38, November 1999.

[18] C.-F. Westin, S. E. Maier, H. Mamata, A. Nabavi, F. A. Jolesz, and R. Kikinis. Processing and visualization of diffusion tensor mri. *Medical Image Analysis*, 6(2):93–108, 2002.