T.C.
FATIH UNIVERSITY
INSTITUTE OF BIOMEDICAL ENGINEERING

DESIGN A HYPERRESOLUTION FIBER TRACKING SYSTEM FOR OPTIMAL NAVIGATION BY THE NEUROSURGEONS IN DEEP BRAIN STIMULATION

RUHUNUR ÖZDEMİR

MSc THESIS

BIOMEDICAL ENGINEERING PROGRAMME

İSTANBUL, AUGUST / 2013

T.C.
FATIH UNIVERSITY
INSTITUTE OF BIOMEDICAL ENGINEERING

DESIGN A HYPERRESOLUTION FIBER TRACKING SYSTEM FOR OPTIMAL NAVIGATION BY THE NEUROSURGEONS IN DEEP BRAIN STIMULATION

RUHUNUR ÖZDEMİR

MSc THESIS
BIOMEDICAL ENGINEERING PROGRAMME

THESIS ADVISOR
PROF. DR. SADIK KARA

İSTANBUL, AUGUST / 2013

T.C.
FATİH ÜNİVESİTESİ
BİYOMEDİKAL MÜHENDİSLİK ENSTİTÜSÜ

DERİN BEYİN STİMULASYONUNDA BEYİN CERRAHLARININ OPTİMAL YÖNLENDİRİLMESİ AMACI İLE BİR HYPERRESOLUTION FIBER TAKİP SİSTEMİ TASARIMI

RUHUNUR ÖZDEMİR

YÜKSEK LİSANS TEZİ
BİYOMEDİKAL MÜHENDİSLİĞİ PROGRAMI

DANIŞMAN
PROF. DR. SADIK KARA

İSTANBUL, AĞUSTOS / 2013

T.C.

FATIH UNIVERSITY

INSTITUTE OF BIOMEDICAL ENGINEERING

Ruhunur Özdemir, a **MSc** student of Fatih University **Institute of Biomedical Engineering** student ID 52011114, successfully defended the **thesis** entitled "**DESIGN A HYPERRESOLUTION FIBER TRACKING SYSTEM FOR OPTIMAL NAVIGATION BY THE NEUROSURGEONS IN DEEP BRAIN STIMULATION**", which she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : Prof. Dr. Sadık KARA.....

Fatih University

Jury Members : Prof. Dr. Sadık KARA.....

Fatih University

Assoc. Prof. Dr. Olcay KURŞUN.....

Istanbul University

Ast. Prof. Dr. Haşim Özgür TABAKOĞLU.....

Fatih University

Date of Submission : 25 July 2013

Date of Defense : 27 August 2013

To my mother Selma ÖZDEMİR and my father Ali Kemal ÖZDEMİR…

ACKNOWLEDGEMENTS

I would like to express my thanks to my supervisor, Prof. Dr. Sadık Kara, Head of Institute of Biomedical Engineering and my thanks are also extended to Prof. Dr. Bart Ter Haar Romeny, Chair Biomedical Image Analysis, Eindhoven University of Technology and ir. Birgit Plantinga, PhD Canditate for their guidance, support and giving their time so generously.

My deepest gratitude are also due to my beloved family, without whose understanding, motivation, endless support, encouragement and love I would not have overcame.

I would also like to express my gratitude to Maria Vourmans for her kind hospitality, warm friendship, behaving like a mother to me, through the duration of living in Netherlands.

August 2013 Ruhunur ÖZDEMİR

TABLE OF CONTENTS

LIST OF SYMBOLS

P_i	Predictor
Δt	Time changing
h	step size
E(r)	normalize vector field

ABBREVIATIONS

DBS : Deep Brain Stimulation
DTI : Diffusion Tensor Imaging
FACT : Fiber Assignment by Continuous Tracking
MRI : Magnetic Resonance Imaging
ODE : Ordinary Differential Equation
PD : Parkinson Disease
RK4 : 4th Runge-Kutta
ROI : Region of Interest
STN : Subthalamic nucleus

LIST OF FIGURES

Page

LIST OF TABLES

SUMMARY

DESIGN A HYPERRESOLUTION FIBER TRACKING SYSTEM FOR OPTIMAL NAVIGATION BY THE NEUROSURGEONS IN DEEP BRAIN STIMULATION

Ruhunur ÖZDEMİR

Biomedical Engineering Programme

MSc

Advisor: Prof. Dr. Sadık KARA

The thalamus is one of the most critical brain structure to provide strong connection between subcortical and cortical areas of the brain. The subthalamic nucleus is located ventral to the thalamus. Deep brain stimulation is a technique which commonly practices for advenced patients who suffer from Parkinson's disease to effectively allay the patients' typical motor symptoms such as tremor, disorientaion on long term. In this technique, the subthalamic nucleus is stimulated, but can cause to cognitive and psychiatric disadvantegeous, therefore, the STN's scale is 6 mm, which is too small to reveal. By this motivation point, in the thesis, I propose an approach that is to gain spatial resolution using post-processing method to disclose the substructures of biological regions in the brain.

Diffusion Tensor Imaging shows random motion of water molecules in biological interest. On conventional MRI, each voxel has an principle direction and value (main eigenvector and eigenvalue) which obtains from diffusion (water molecules motion). DTI measures a symmetric diffusion tensor matrix in each voxel, herewith the principle direction and value can be calculated from diffusion tensor matrix. In the thesis proposes a method to visualize all voxels as an Ellipsoid considering the main direction of the voxel and to reconstruct fiber tracks by incorporating extra information from seedpoints that are generated by randomly seeding throughout the brain and region of

interest. The fiber reconstruction method shows global information of biological interest. Last, the thesis propeses a method to calculate track density for further track density mapping.

Keywords: Deep Brain Stimulation, Fiber Reconstruction, Track Density, Hyperresolution, Parkinson Disease, Diffusion Tensor Imaging

FATIH UNIVERSITY - INSTITUTE OF BIOMEDICAL ENGINEERING

ÖZET

DERİN BEYİN STİMULASYONUNDA BEYİN CERRAHLARININ OPTİMAL YÖNLENDİRİLMESİ AMACI İLE BİR HYPERRESOLUTION FIBER TAKİP SİSTEMİ TASARIMI

Ruhunur ÖZDEMİR

Biyomedikal Mühendisliği Programı

Yüksek Lisans

Danışman: Prof. Dr. Sadık KARA

Talamus, beynin subkortikal ve kortikal bölgeleri arasında bağlantıyı sağlamak için en önemli beyin yapılarından biridir. Subtalamik nucleus talamusa ventral olarak yerleştirilmiştir. Derin beyin uyarma tekniği ileri düzeydeki Parkinson hastalarında tremör ve oryantasyon bozukluklarını etkili bir şekilde azaltmak için yaygın olarak kullanılan bir yöntemdir. Bu teknikte subtalamik nükleus uyarılır, STN bölgesinin 6 mm lik ortaya çıkarmak için küçük bir alan olmasından dolayı, bazı cognitif ve piskolojik bozukluklara neden olabilir. Bu motivasyon noktası sayesinde, tezde, beyindeki biyolojik bölgelerin altyapılarının ortaya çıkabilmesi için post-processing yöntem kullanarak, uzaysal resolasyon kazanan bir yöntem ileri sürüyorum.

Diffusion Tensor Imaging, biyolojik yapılardaki su moleküllerinin rastgele hareketlerini gösterir. Konvansiyonel MRI' da, her bir voksel difüzyondan elde edilen ana bir yöne ve değere sahiptir (ana özvekör ve özdeğer). DTI her bir vokselde bir difüzyon tensörü olçer, böylelikle difüzyon tensor matrisinden ana vector ve değer hesaplanabilir. Bu tezde, bütün vokselllerin ana yönleri dikkate alınarak, vokseller Ellipsoid olarak görüntüleyebilmek için ve bütün beyin boyunca rastgele oluşturulmuş seedpointlardan ve belirlenen bölgelerden ekstra bilgiler sayesinde fiber tracklarin olşturulması için bir metod ileri sürülüyor. Fiber track reconstruction metod, biyolojik bölgenin global

bilgisini gösterir. Son olarak, ileriki trak density mapping için bir track density method öneriliyor.

Anahtar kelimeler: . Deep Brain Stimulation, Fiber Reconstruction, Track Density, Hyperresolution, Parkinson Disease, Diffusion Tensor Imaging

CHAPTER 1

LITERATURE SURVEY

Diffusion Tensor imaging and diffusion weighted imaging are complementary techniques to utilize the diffusion characteristics of water movement in biological interest. Diffusion tensor imaging can be used to disclose the substructures of biological interest espically, white matter in the brain. Water movement can be expressed as tractography which means based on the voxel main (principal) direction as the fiber tracking direction [1]. Accourding to my researches, the vector directions can be vizualized diffirent types of glyphes such as cuboid, ellipsoid, etc [1]. The ellipsoid is used to effectively determine the direction of each voxels. In order to fully characterize the diffusion tensor matrix in each voxels, needs six parameters that can be maintained by ellipsiod in the 3D space inspection[2]. Using trilinear interpolation, discerete MRI data is turned into continuous [3]. The tracktography approach roughly distinguishes track propogation and energy minimization [3], in this study, track propogation method is perefered. According to former approach that is deterministic is not literally enough to reconstruct fibers, so track propogation can be based on 4th Runge-Kutta method [4] associated with the Fiber Assignment by Continuous Tracking (FACT) algorithm to increase achieved voxel information[5]. The stopping criteria which is fractional anisotropy, vector angle between voxels, the whole image dimensions is determined to terminate tracktography [2]. In order to calculate number of track, the grid element is smaller than the accuired voxel size and therefore the spatial resolution of the final demonstration is much higher than original MRI data by tracktography method [6].

PURPOSE OF THE THESIS

Roughly speaking, I aim to improve an approach which gains hyperresolution to reveal substruction of biological interest in the brain for optimal navigation by the neurosurgeons in deep brain stimulation. By clarifying, the thesis focuses on diffusion tensor to define a principal nature of directions for each voxel through its eigenvectors. The diffusion displacement can be represented as ellipsoids which fully characterize with the principal axes lengths related to the tensor eigenvalues and the directions which are given by the tensor eigenvectors. The main eigenvector is corresponding to the direction of main diffusivity is presumed to give the fiber direction in white matter regions to estimate white matter connectivity. Thus, I concentrate on demonstration of main directions in each voxel as ellipsoids to enoughly obtain information facilitating characterization of MRI data. The method also considers of anisotropy indices, espically fractional anisotropy that brings into prominence for accurating demonstration of fibers in the white matter. The approach provides following the whole brain or a specified area of the brain through main direction of the voxels under the hyperresolution post-processing technique. For futher studies, the thesis permits track density mapping using directional encoded color.

CHAPTER 2

INTRODUCTION

2.1 Deep Brain Stimulation and the STN

Deep Brain Stimulation (DBS) has enabled notable benefits for people who suffer from neurological diseases. The tremor can dramatically be cured by stimulation of the intermediate nucleus of the thalamus or subthalamic nucleus [7] . The surgery which is stereotactic surgery entails the implantation of a medical device which called brain pacemaker. The specific regions of the brain is electrically stimulated in this technique. Electrodes that are connected to a stimulator or battery device are put deeply in the brain. The brain pacemaker is similar to the heart pacemaker. Both use a neurostimulator that employs electric pulses to help regulate brain activity. To manage some of Parkinson's Disease symptoms, DBS has lately been shown for patients who do not answer to drug treatments, and in order to improve the patient's quality of life [8]. DBS method performs for the purpose of reduce the PD's symptoms, such as, tremor, rigidity, and slowness of movement which are arised from stopping producing of dopamine in nerve cells that are responsible for transfering messages which manage body movement. While performing, electrodes that are connected by an extension wire to a stimulator are located deeply in the brain via small holes that are in the skull. The left side of the brain manages the right side of the body and vice versa, so, DBS is usually implemented on both sides of the brain [9].

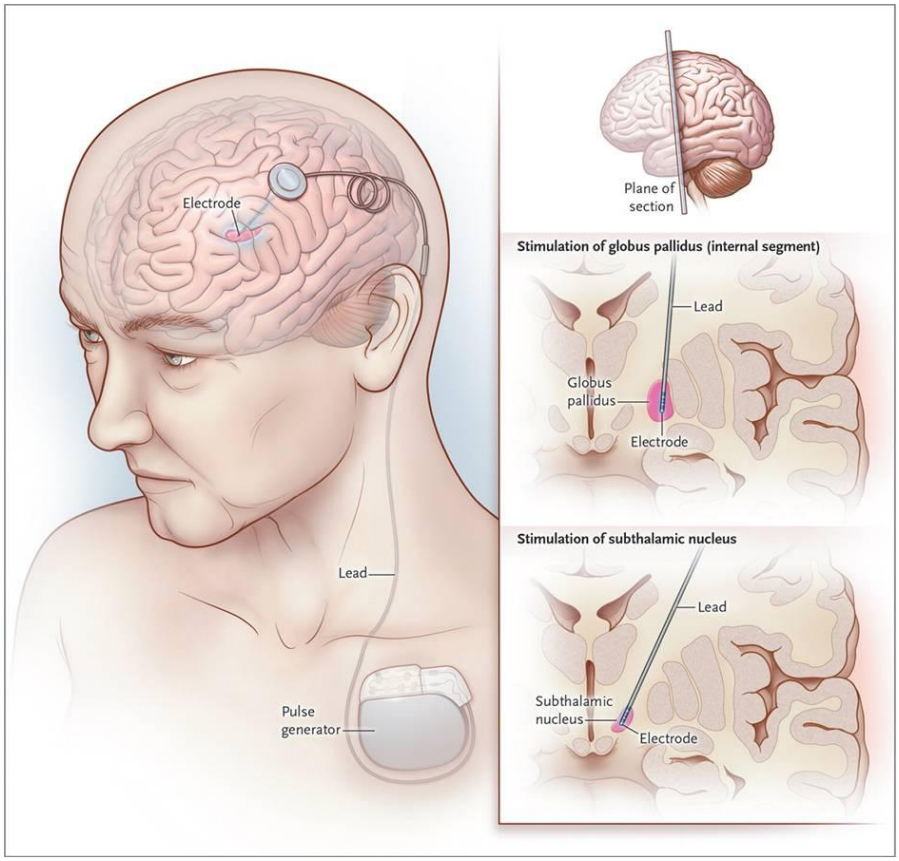

Figure 2.1 Deep Brain Stimulation [9]

It is a treatment choise for people who have suffer from Parkinson's disease and whose symptoms can not well managed with medication. Parts of the basal ganglia that plays a main role in neurological conditions are either below or above stimulated by stimulator. Natural movements are replaced by tremor, rigidity and stiffness (involuntary movements) in PD. DBS of specific ganglia modifys the irregular electrical impulse and helps for stabilizing the feedback loops, hereby the symptoms are reduced. With Parkinson’s Disease patients, the research area has narrowed for the focus of movement disorder surgery to 3 essential gray matter structures as a result of contiuned enhancement of the knowledge of basal ganglia circuitry: the thalamus, the globus pallidus, and the subthalamic nucleus. STN is most important part of basal ganglia in patients, because of displays a hyper-activity that causes inhibition of motor circut target structures.

The STN has been distinguished into three parts that are functionally different. Based upon former researches, the largest part is the motor area, associative and last one is a limbic area. After DBS, the serious adverses can be occured on cognition and behavior.

According the researches, the behavioral change is observed in 50 % of the cases, in the patients and also the change is intense in 14 % of the cases. Therefore, determining the motor part the STN is crucial to obtain the best possible surgery result on the STN and to minimize the enervating effects on cognition and behavior. However, current MRI techniques do not yet simplify to accuratly identify a substructure of biological interest such as the STN.

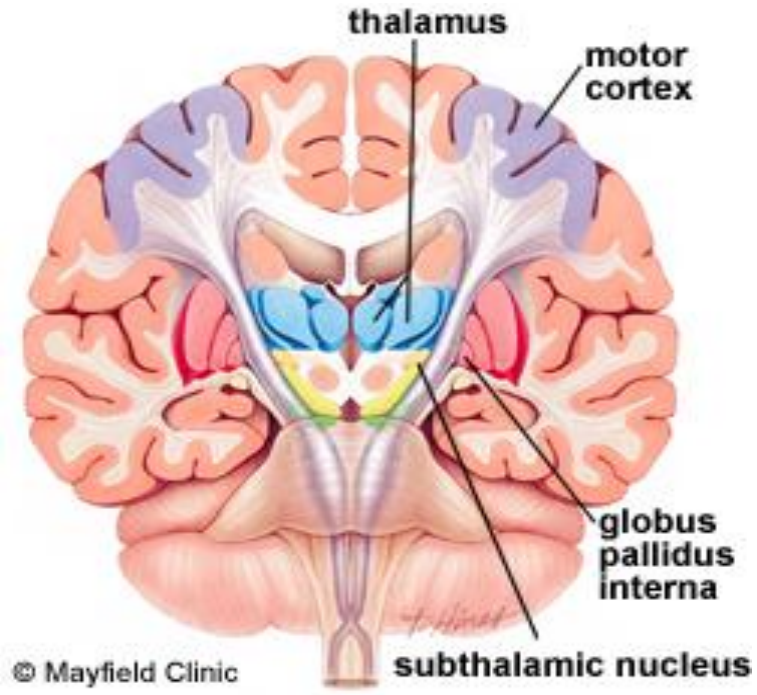

Figure 2.2 Subthalamic nucleus [9]

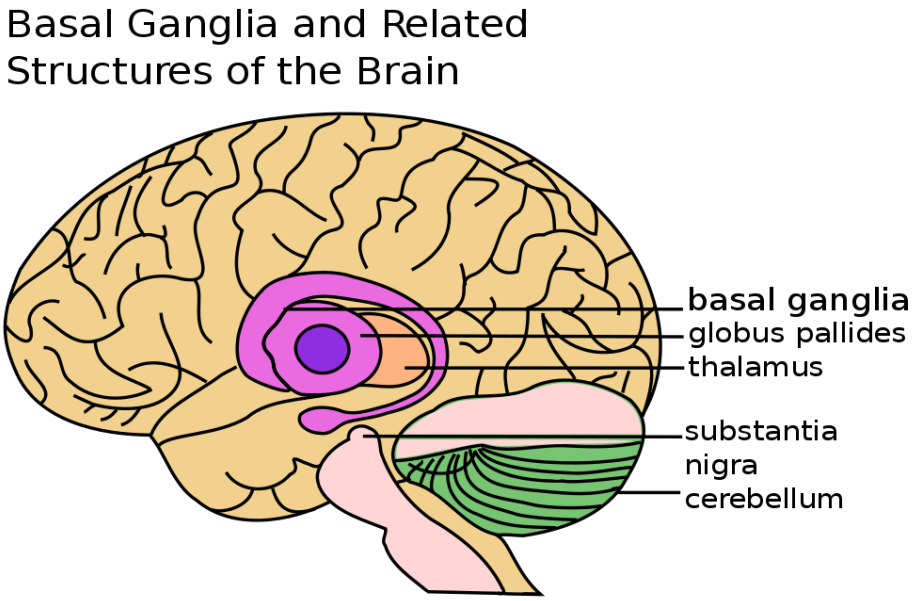

Figure 2.3 Important and related parts of brain for Parkinson's Disease in DBS [10]

The above Figure 2.3, electrodes can be placed and which adverses are in which interets by DBS. Subthalamic nucleus (STN) is effective for tremor, slowness, rigidity, dystonia and dyskinesia, thalamus (VIM) is often used to alleviate essential tremor and last one,Globus pallidus (GPi) is effective for tremor, slowness, rigidity, dystonia and dyskinesia. [10]

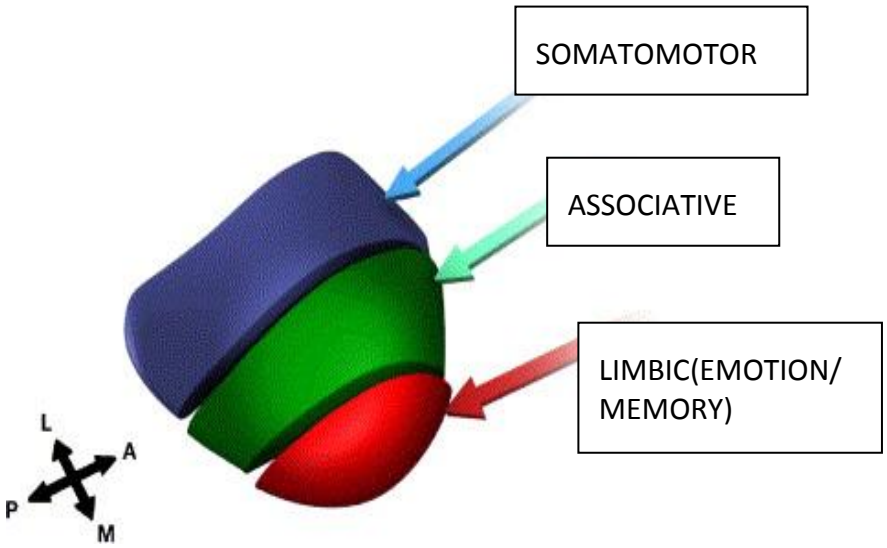

Figure 2.4 Parts of the STN [10]

CHAPTER 3

MATERIAL AND METHOD

3.1 Data and Software

It is preffered to use phantom data which was created by the program Camino that was developed by UCL(University College London) Microstructure Imaging and also artifical data was computed by myself. In order to improve method, Mathematica 9.0 student edition interface is used.

The phantom data is a Matlab (Matrix Laboratory) (.mat) file. It is a 128×128×52×6 matrix. The first tree dimensions are the spatial dimensions and the last dimension consists of the six unique elements is as follows : $D_{xx}, D_{xy}, D_{xz}, D_{yy}, D_{yz}, D_{zz}$.

The artifical data was computed as a real symmetric tensor and the dimensions can be sized. Processing of phantom data and artifical data are done by Wolfram Mathematica was preffered by reason of easily implementing and determining mathematical functions. And also MathVision tools that is developed by BMIA (BioMedical Image Analysis), Depertmant of Biomedical Engineering in Eindhoven University of Technology (TU/e) is implemented.

3.2 Diffusion MRI

Diffusion MRI performs non-invasively for mapping of the diffusion process of molecules which is essentially water molecules in vivo. Diffusion MRI is a kind of Magnetic Resonance Imaging (MRI) method. Diffusion MRI rapidly becomes a standard procedure for white matter disorders, as Diffusion Tensor Imaging (DTI) that can be exposed malformations in white matter of the brain and can be provided determined models of brain network. Molecular diffusion entitles the random transitive motion of molecules, also titled Brownian motion that proceeds from the thermal energy carried by these molecules. Water is the most convenient molecule to study with diffusion MRI that moves in the brain. The water molecules move forward averaged distances approximately 10 mm, bouncing, crossing, or interacting with many bological interest such as cell membranes, fibers, or macromolecules, etc. [11].

The diffusion that is water molecule mobility is not same in all directions in tissues is precisely a three dimensional process, the mobility is anisotropic in the brain's white matter. With diffusion tensor imaging (DTI), diffusion anisotropy's results can be fully extracted, characterized, and utilised, providing even more critical details on substructure of biological interest. In the existence of isotropy which means the water molecules can be measured same amount of all directions, Diffusion's results can be characterized by a scalar parameter. It called the diffusion coefficient, D. The result of diffusion is an attenuation, A, which based on the diffusion coefficient D and the b factor, b factor defines the gradient pulses such as timing, amplitute, shape, used in the MRI sequence on the MRI signal [12]. In the meantime, diffusion constant can be achieved from the amount of the signal loss of one image, but can not be achieved from the signal intensity in the MRI :

$$A=\exp(-bD) \tag{3.1}$$

In the existence of anisotropy, henceforth, diffusion cannot any more be described by a single scalar coefficient. A tensor, $\underline{D}$, which fully describes molecular displacement through each direction and correlation between molecules' directions requires to characterize the displacement [12]:

$$\underline{D} = \begin{bmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{bmatrix} \quad (3.2)$$

The tensor is calculated as real symmetric tensor

the attenuation, A , is represented by a new equation as the following :

$$A = \exp(-b_{xx}D_{xx} - b_{yy}D_{yy} - b_{zz}D_{zz}) \quad (3.3)$$

where b_{ii} are the elements of the b matrix which is replaced the b factor symbolized in the coordinates of this reference pattern.

Diffusivity is along x, y, and z axes, shown in D_{xx} (a), D_{yy} (d), and D_{zz} (f) images that are in the given order, Diffusivity is apparently different in white matter in the brain, especially in the corpus callosum. D_{xy} (b), D_{xz} (c), and D_{yz} (e) which are nondiagonal images are not noise images because the x, y, z reference pattern of the MRI scanner does not happen with the diffusion reference pattern of biological interest in most voxels at the same time [12]. The new equation must be considered the coupling of the non-diogonal elements, D_{ij} ($i \neq j$) which reflect interrelation between the molecular movement in perpendicular directions. The new equation can be illustrated as follow :

$$A = \exp(-b_{xx}D_{xx} - b_{yy}D_{yy} - b_{zz}D_{zz} - 2b_{xy}D_{xy} - 2b_{xz}D_{xz} - 2b_{yz}D_{yz}) \quad (3.4)$$

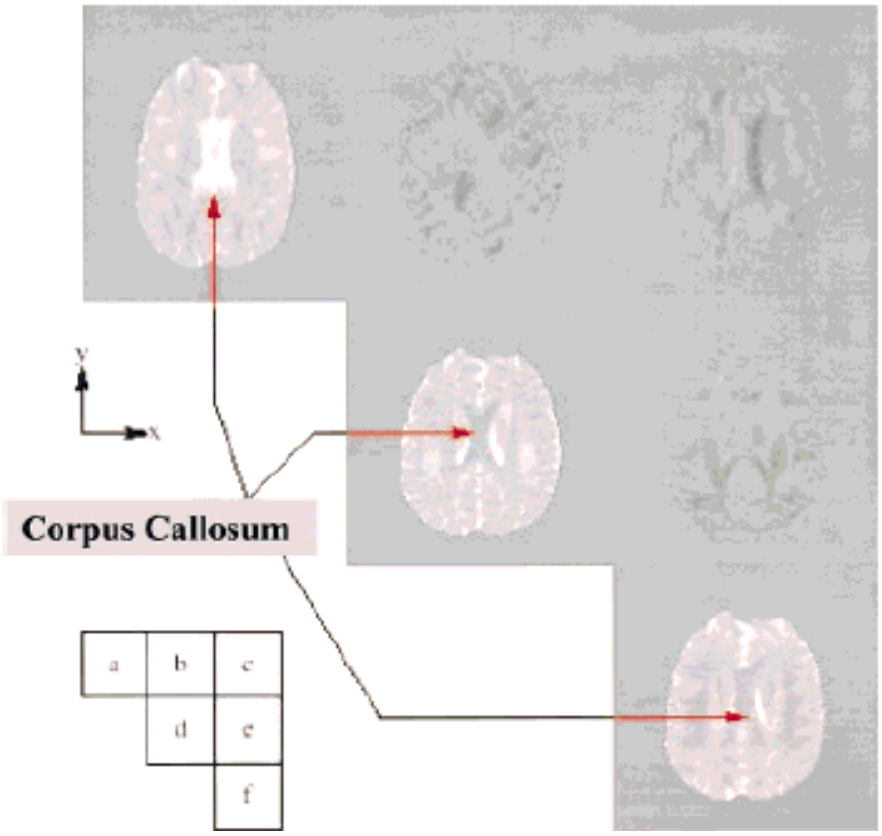

Figure 3.1 The diffusion tensor is in axial slices. Shows in D_{xx} (a), D_{yy} (d), and D_{zz} (f) [12]

3.3 Diffusion and Directionality

One of the most essential aspect of the diffusion measurement by MRI measures diffusion throughout one predetermined axis. For instance, applying a gradient along the horizontal axis, the signal of MR becomes sensitive just to horizontal movement If directionalty is random and inconsistent motion of water molecules in which water diffuses with the same amount in all directions, It is called Isotropic. In this case, We need to only one scalar which is the diffusion constant (D) as like what is mentioned that before and It also called Apperent Diffusion Coefficient (ADC) to describe the diffusion. D represents the diameter of the sphere. A sphere needs only one parameter to be determined. The other case is restricted diffusion, which means the diffusion is not the same all directions because of the physical barriers which are membranes and organels in a biological tissue.

In the brain, bodies of nerve cells are established in the grey matter which is essentially scattered surface of the brain, but axons are located in the white matter which consists of bundles of myelinated axons which are responsible for anisotropy. However, Diffusion represents as an oval in a 2D plane or ellipsoid in 3D and It is called anisotropic diffusion, and the ellipsoid is called Diffusion Ellipsoid. In anisotropic case, diffusion is larger than in one direction (principal direction) than in the others.

D is symbolized a symmetric ($D_{ij} = D_{ji}$, with i,j = x,y,z) 3x3 diffusion tensor $\underline{D}$ and It must be positive :

$$\underline{D} = \begin{bmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{bmatrix} \tag{3.4}$$

The diagonal elements D_{xx}, D_{yy}, D_{zz} present the ADC along the x,y,z axes.The off-diagonal elemets present the correlation between the diffusion which is in perpendicular directions. In isotropic case, there isn't correlation between the perpendicular directions.

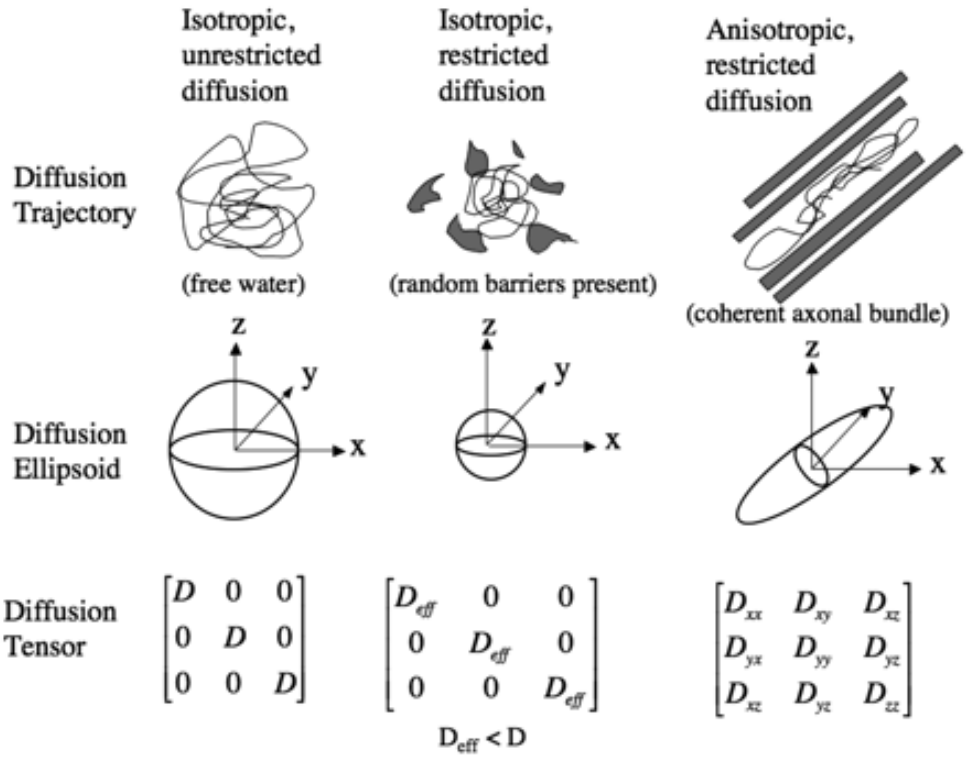

Figure 3.2 Diffusion trajectory, Diffusion Ellipsoid,Diffusion Tensor

3.4 Analyse of Diffusion Tensor and Eigensystems

To characterize the diffusion process with directionality in the presence of anisotropy, an ellipsoid is in need of six parameters to is defined which are three lengths for the longest, shortest, and middle axes and these three lengths must be perpendicular to each other. In order to identify the principle direction of every voxel, the diffusion tensor which is in each voxel can be diagonalized. The diagonalization provides the lengths which can be illustrated as, $\boldsymbol{\lambda}_1$, $\boldsymbol{\lambda}_2$, and $\boldsymbol{\lambda}_3$, and called eigenvalues and also, provides to define the direction of the principal axes, three unit vectors which are $\overrightarrow{\mathbf{e}_1}$, $\overrightarrow{\mathbf{e}_2}$, $\overrightarrow{\mathbf{e}_3}$ are called eigenvectors with three corresponding eigenvalues. The eigenvector which is corresponding to the largest eigenvalue represents in the direction of the largest diffusion coefficient (principal direction). The diagonalization of the diffusion tensor can mathematically be described as follows [8][1]:

$$\underline{D}\,\vec{e}_i = \lambda_i \vec{e}_i \qquad \vec{e}_i \neq 0 \tag{3.5}$$

The solution λ_i are the eigenvalues of $\underline{D}$. The vectors $\vec{e}_i$ which are associated with each eigenvalue are the eigenvectors of $\underline{D}$. We can reconstruct new equation to calculate λ_i from equation (3.5) . The new equation is $\left(\underline{D} - \lambda_i \underline{I}\right)\vec{e}_i = 0.$

$\underline{I}$ represents the Identity matrix. This mentions that the matrix $\underline{D} - \lambda_i \underline{I}$ is singular and its determinant is zero.The new equation is as follows:

$$\left|\underline{D} - \lambda_i \underline{I}\right| = 0 \tag{3.6}$$

For each eigenvalue λ_i the correspoinding eigenvector $\vec{e}_i$ can be found as follows:

$$\left(\underline{D} - \lambda_i \underline{I}\right)\vec{e}_i = 0 \tag{3.7}$$

For instance T is a Tensor. We can calculate the eigenvalues and eigenvectors for tensor T according to the equations:

$$T = \begin{bmatrix} 163/34 & -61/34 & 18/17 \\ -61/34 & 163/34 & -18/17 \\ 18/17 & -18/17 & 109/17 \end{bmatrix}$$

To calculate the eigenvalues in Mathematica:

Solutions = Solve[Det[(T- λ IdentityMatrix[3])] == 0, λ] (3.7)

The results are as follow : $\{\{\lambda \to 3\}, \{\lambda \to 5\}, \{\lambda \to 8\}\}$

To find the largest eigenvectors, sorting eigenvalues:

eigenvalues = ((λ/.#) &@Solutions (3.8)

eigenvalues = Sort[N@eigenvalues,Abs[#1]>Abs[#2]&] (3.9)

The results are as follow : {8.,5.,3.}

To obtain largest eigenvalue :

Eigenvalues[[1]] (3.10)

To obtain principle eigenvector :

Evsolution1 = Solve[(T-eigenvalues[[1]]* IdentityMatrix[3]).{e1,e2,e3} = = {0,0,0}]

The result is as follow: {e1→0.+0.75 e3, e2→0.- 0.75 e3}

To plot the eigenvalues :

Plot[Det[T-λ IdentityMatrix[3]],{λ,0,10}] (3.11)

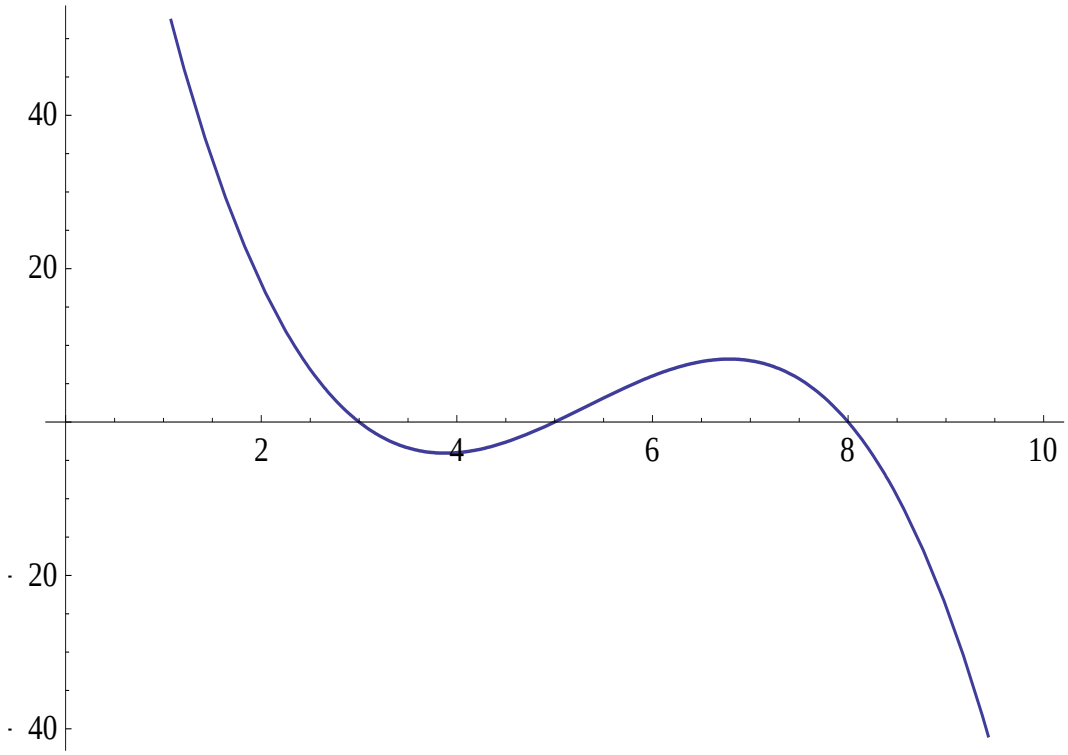

Table 3.1 Eigenvalues

Applying modules in mathematica:

CreateSymRandomRealTensor[row_,column_]:=Module[{r=row,c=column,tensor,symtensor}] :

ArtificalSymmetricTensor3D[xdim_,ydim_,zdim_]:=Module[{x=xdim,y=ydim,z=zdim}] ;

VisualizationEigVec[tensor_,point_]:=Module[{evec,eval}] ;

Graphics3D[MapIndexed[VisualizationEigVec,ardata,{3}],FaceGrids→All];

CalMainEigenvectorAllardata3D[phantomdata_]:=Map[Eigenvectors[#]⟦1⟧&,phantomdata,{3}] ;

Graphics3D[MapIndexed[{Red,Point[#2],Gray,Arrow[{#2,#1+#2}]}&,mainevec,{3}],Axes→True,AxesLabel→{x,y,z},FaceGrids→All];

ardata is a variable which is named for artifical data and its dimensions are 3x3x3.

The artifical data is created and the principal eigenvectors are calculated. And also, the principle vectors corresponding to each voxel are visualised. In the following figure, the principle vectors is shown in each voxel or can be called grid.

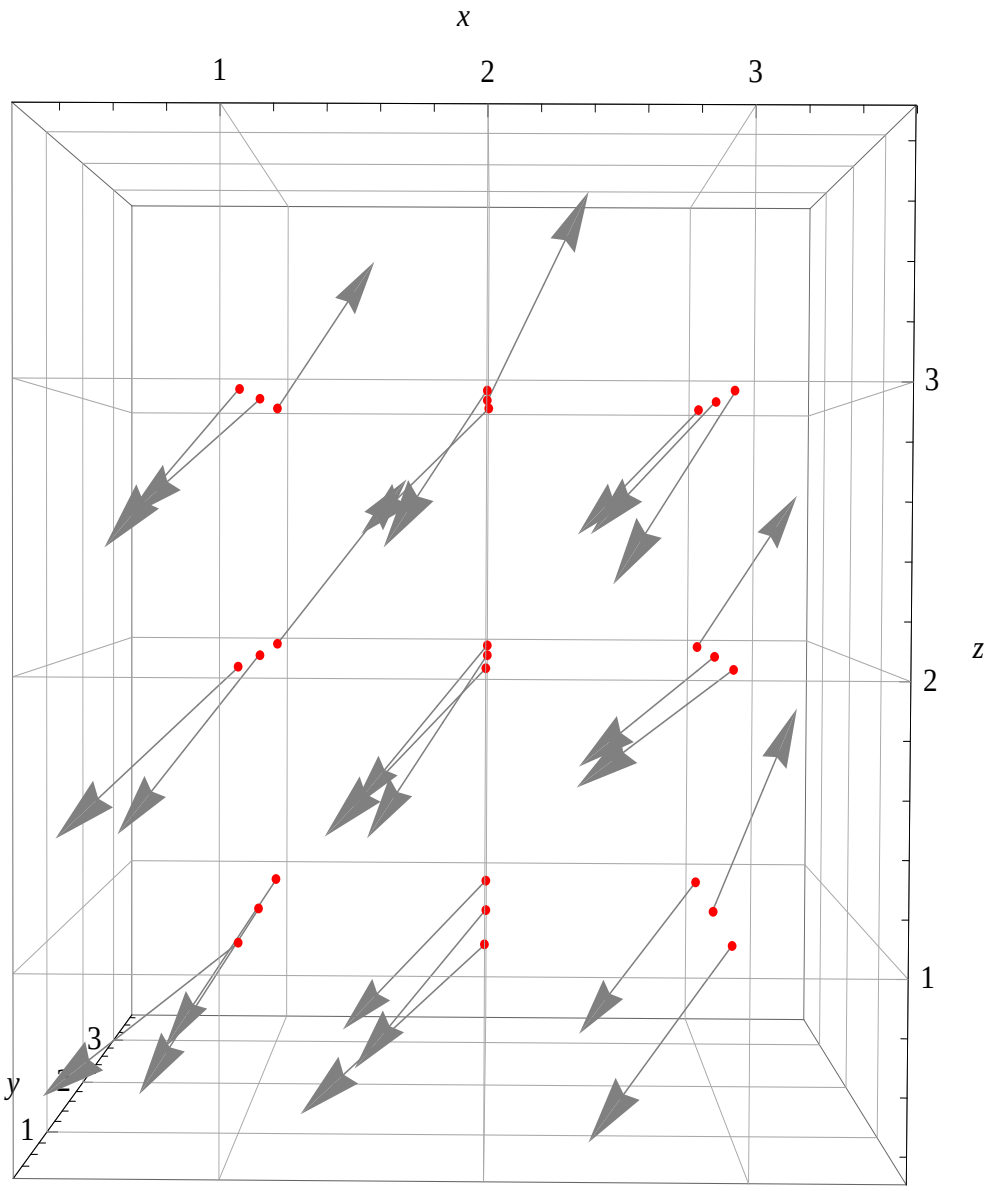

Figure 3.3 Grids, Eigenvectors and Directions of Eigenvectors in 3D space

In the following, ListEllipsoidPlot3D that is a package from MathVisionTools shows Ellipsoids corresponding with eigenvalues and eigenvectors in each voxel.

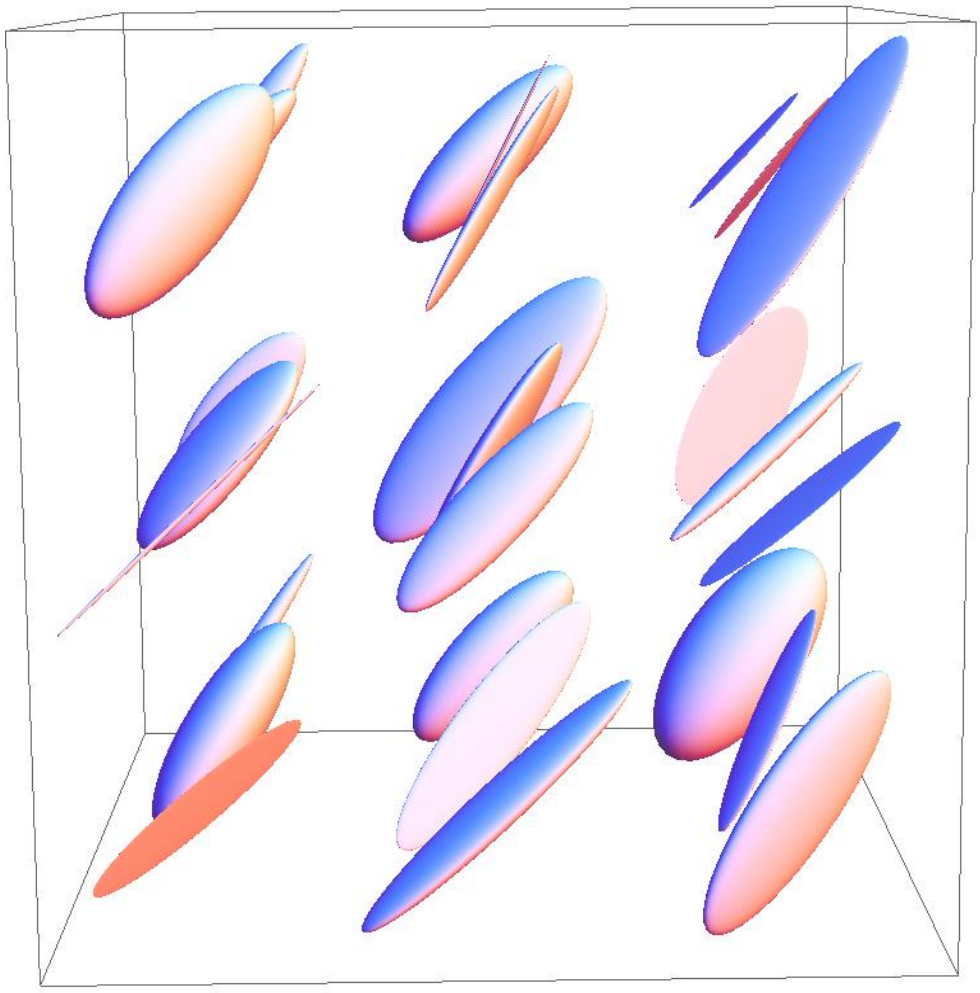

Figure 3.4 Ellipsoid visualization of eigenvectors in 3D

3.5 Anisotropy Indices

In order to indicate the amount of diffusion in presence of anisotropy, several indices can be identified for the diffusion tensor. In this thesis, the indice is based on the eigenvalues of the diffusion tensor. In some approaches, indices are divided into isotropic and anisotropic. The presence of anisotropy, indice is divided into two description which are planar (pancake shape) and linear (cigar shape). According to studies, there are two anisotropy indeces which are frequently used, fractional anisotropy (FA) and relative anisotropy (RA). The range of these indices is between zero which is isotropic and one which is anisotropic. Diffusion anisotropy was divided into three basic forms which are depended on the rank of the diffusion tensor. The diffusion which is related to the principle direction which is corresponding to the largest eigenvalue is called linearly anisotropic C_l, where the diffusion which is restricted to a plane spanned by the two eigenvectors which are corresponding to the two largest eigenvalues is called planarly anisotropic, C_p and the last one is C_s which is actually isotropic $C_s = 1$ only when all the eigenvalues can equally be calculated.

Where C_l is high, the eqution and shape are as follows:

$$\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3}$$

Figure 3.5 Equation of Linearly anisotropic indice, C_l and shape [20,21]

Where C_p is high, the equation and shape are as follows :

$$\frac{2(\lambda_2 - \lambda_3)}{\lambda_1 + \lambda_2 + \lambda_3}$$

Figure 3.6 Equation of Linearly anisotropic indice, C_p and shape [20,21]

Where C_s = 1, the equation and shape are as follows :

$$\frac{3\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}$$

Figure 3.7 Equation of Linearly anisotropic indice, C_s and shape [20,21]

It can be shown that all the values fall in the range [0,1], and that they sum to unity:

$$c_l + c_p + c_s = 1 \tag{3.12}$$

The barycentric coordinates can be used to depict the space of possible anisotropies. In the triangle every point has a corresponding ellipsoid for which the anisotropy measures (C_l ,C_p and C_s) appraise to the point's barycentric coordinates. At each vertex of the triangle, one of the anisotropy measures is one, while the two others are both zero. Along the sides of the triangle, one side of the anisotropy measures is zero, and the other two measures sum to one as follows [20]:

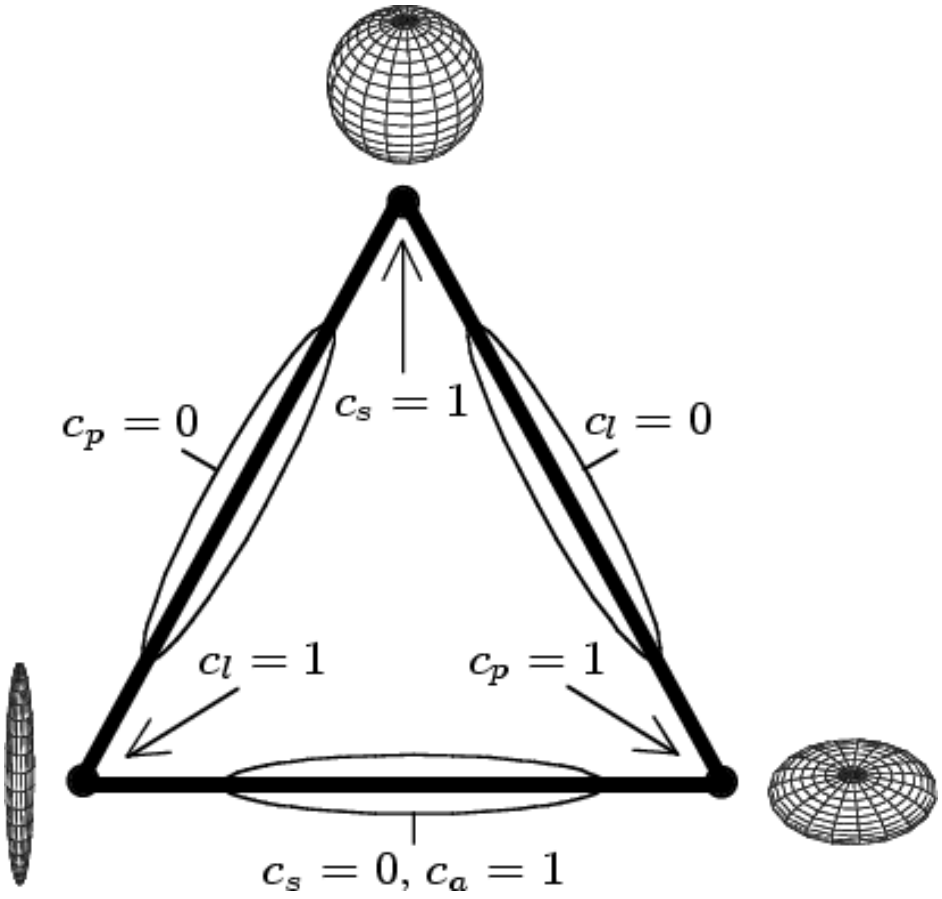

Figure 3.8 Barycentric space of anisotropies. [20,21]

3.6 Tractography

Based on primate studies, DTI gives two type information which are the extention of diffusion anisotropy and direciton of diffusion anisotropy about water moleculs motion in the brain. The largest principal axis of the diffusion tensor shows fiber orientation in a MRI voxel. The several different approaches which can be roughly divided into two types have been proposed to reconstruct tracts. One is based on line propagation algorithms that to reveal direction using local tensor information for each step. The main differences among techniques in line propagation are how to define smooth trajectories from neighboring voxels and the way of reducing noise. The second approach of reconstruction fiber is based on global energy minimization to find the energetically most notable path between two predetermined voxels. Line propogation

techniques are invetigated in the thesis. One of the line propogation techniques is simple linear tract propagation. The line can be started from a randomly chosen seedpoint, then accourding to stopping criteria, the line is terminated. Herewith, simple linear propogation technique do not facilitate to reconstruct more smoother fiber line and to reduce noise. Therefore, a different approach is proposed 4th Runge-Kutta associated FACT by based on simple line propogation technique.

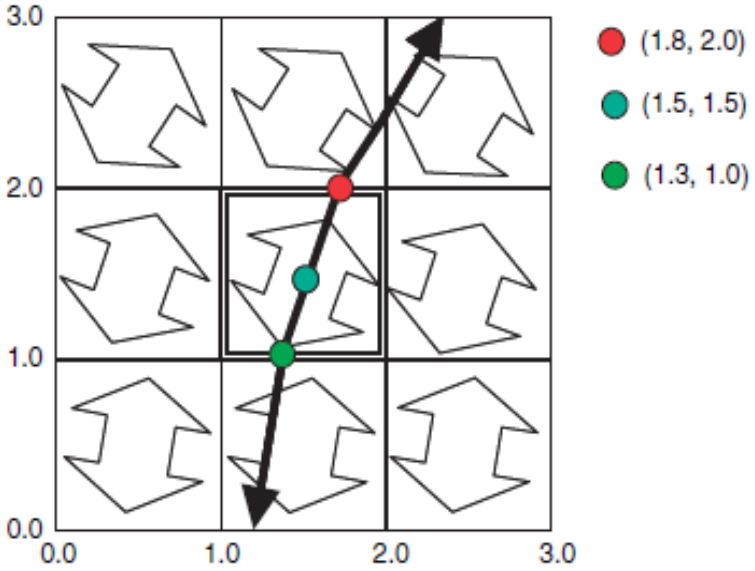

Figure 3.9 Simple Linear Propogation in 2D space [8]

In process of tractography, we perceive the directions of in each voxel at every point in the brain (interpolation- the data that are discrete become continuous). The directions can be connect (principal vector) each others to reconstruct entire pathway and therefore the brain connectivity can be visualized. In its basic form, this consists of starting at a seed location and following the preferred direction until a new measurementis reached. Thus, the new direction determines accourding to calculated seed location or applies new seed location and proceed until the entire pathway is traced.

3.6.1 Interpolation

There are several kind of interpolation techniques e.g. linear, bilinear and trilinear interpolation. However, trilinear interpolation technique is performed in the study. Trilinear interpolation is a kind of linear interpolation of a point within a box (3D) given values at vertices of the box. The trilinear interpolation technique purvey taking a value at any point in the voxel [3].

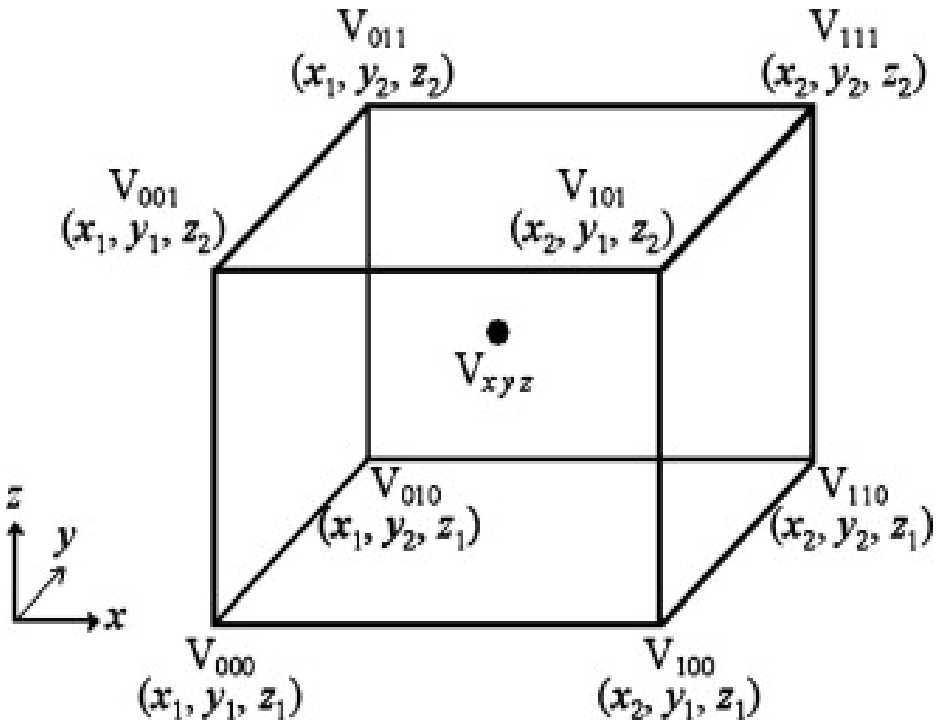

Figure 3.10 Rectangular box for the trilinear interpolation [3]

$$V_{xyz} = V_{000}(1-x'')(1-y'')(1-z'') + V_{100}x''(1-y'')(1-z'') + V_{010}(1 - x'')y''(1-z'') + V_{001}(1-x'')(1-y'')z'' + V_{101}x''(1 - y'')z'' + V_{011}(1-x'')y''z'' + V_{110}x''y''(1-z'') + V_{111}x''y''z''$$

Where

$$x^{"} = (x - x_1)/(x_2 - x_1)\,, y^{"} = (y - y_1)/(y_2 - y_1)\,, z^{"} = (z - z_1)/(z_2 - z_1)$$

In this study, the data is interpolated by Mathematica library, Interpolation[], as follows:

Giving position to determine voxels:

positions=Flatten[Table[{x,y,z},{x,1,3},{y,1,3},{z,1,3}],2]

Tensor List: tensors = Flatten[ardata,2];

Determining the Tensor for each voxel:

PosAndTen=Table[{positions[[z]],tensors[[z]]},{z,1,(3*3*3)}];

Positions and Tensors are interpolated together:

interpolated=Interpolation[PosAndTen,InterpolationOrder→1]

Result is an interpolation function:

InterpolatingFunction[{{1.,3.},{1.,3.},{1.,3.}},"<>"]

Anymore, the discrete data are continuous. Taking intermediate value (Tensor) e.g. [1.2,1.5,1.7].

exintdata=interpolated[1.2,1.5,1.7]//MatrixForm

as follows, result is :

$$\begin{pmatrix} 0.9584179718667934 & 1.1035918900320523 & 1.1054594579112698 \\ 1.1035918900320523 & 0.6453006290891963 & 0.9340447384317935 \\ 1.1054594579112698 & 0.9340447384317935 & 1.0453286664192396 \end{pmatrix}$$

3.6.2 Runge-Kutta Method

The fourth-order Runge-Kutta is mathematically an iterative methods for the approximation of the results that obtain from ordinary differential equation (ODE). Besides, the method resembles with the improved Euler method, but 4th Runge-Kutta method provides more effective results then improved Euler and Euler by utilized predictor and corrector steps. By following figure, assuming that Δt equals 1, the predictor values are illustrated :

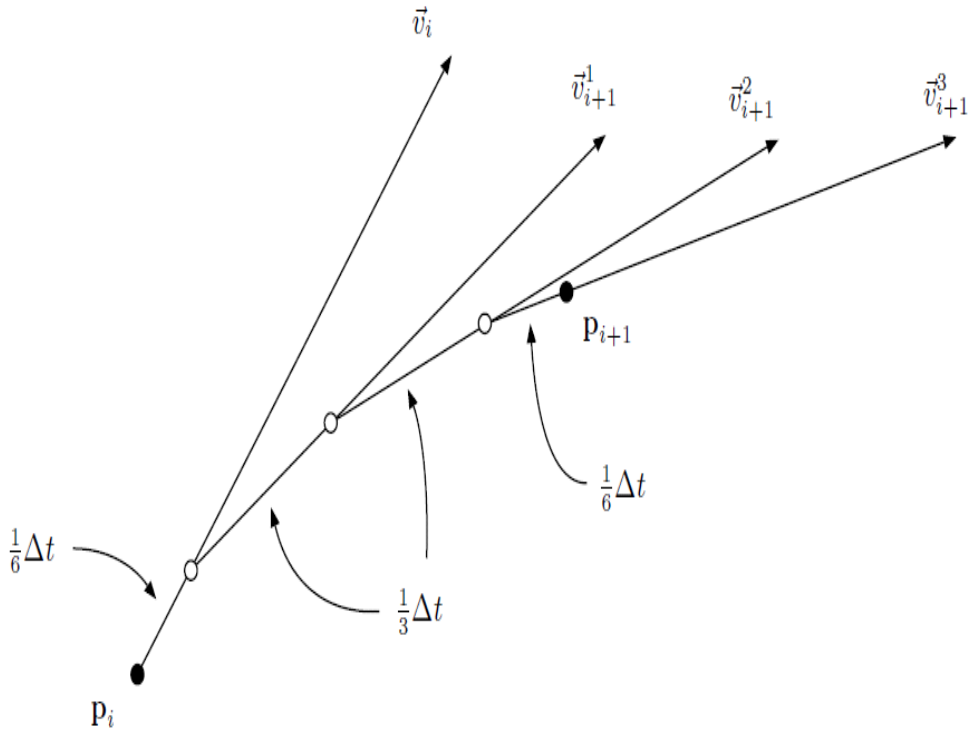

Figure 3.11 Runge-Kutta predictor values [7]

In the following equation, The final predictor value can be calculated :

$$p_{i+1} = p_i + h\frac{(\vec{v}_i + 2\vec{v}^1_{i+1} + 2\vec{v}^2_{i+1} + \vec{v}^3_{i+1})}{6} \qquad (3.13)$$

In order to reveal an image for mapping tracks, the RK4 method might be improved accourding to voxel directipns. So, from an initail point r_0 that is obtained from random seedpoints or ROI to a final point r_N is calculated by all intermediate points r_n by following equations:

$$r_{n+1} = r_n + hV_{n+1} \qquad (3.14)$$

h is step size, V_n is calculated from the identified major diffusion vector that is as a continuous due to trilinear interpolation using RK4. The normalized vector field E(r) is ODE for $n \geq 0$.

$$V_{n+1} = \frac{(k_1 + 2k_2 + 2k_3 + k_4)}{6} \qquad (3.15)$$

Illustrating the equation as the following figure :

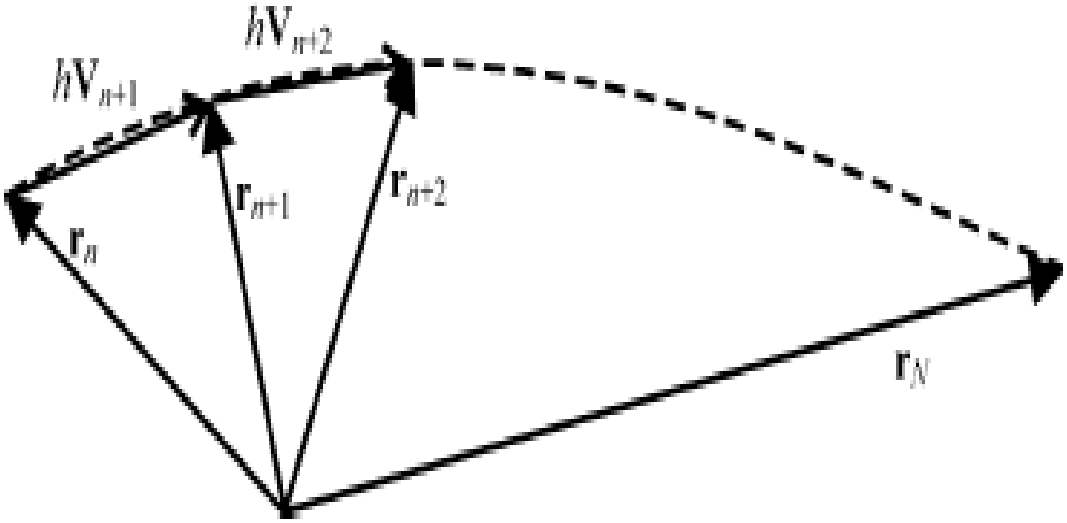

Figure 3.12 Illustration of intermediate points [4]

The step size is used such that convergent results are achieved, so that h is asigned as 1 mm which is half the smallest linear dimensions of a voxel (smaller grid size, as well) [4] :

k_1, k_2, k_3 and k_4 are new predictor values and the values can be calculated by the following equations [4] :

$$\vec{k_1} = \frac{\vec{V_0} \cdot \vec{E}(\vec{r_0})}{\left|\vec{V_0} \cdot E(\vec{r_0})\right|} E(\vec{r_0})$$

$$\vec{k_2} = \frac{\vec{V_0} \cdot E(\vec{r_0} + \frac{h}{2} * \vec{k_1})}{\left|\vec{V_0} \cdot E(\vec{r_0} + \frac{h}{2} * \vec{k_1})\right|} E(\vec{r_0} + \frac{h}{2} * \vec{k_1})$$

$$\vec{k_3} = \frac{\vec{V_0} \cdot E(\vec{r_0} + \frac{h}{2} * \vec{k_2})}{\left|\vec{V_0} \cdot E(\vec{r_0} + \frac{h}{2} * \vec{k_2})\right|} E(\vec{r_0} + \frac{h}{2} * \vec{k_2})$$

$$\vec{k_4} = \frac{\vec{V_0} \cdot E(\vec{r_0} + h * \vec{k_3})}{\left|\vec{V_0} \cdot E(\vec{r_0} + h * \vec{k_3})\right|} E(\vec{r_0} + h * \vec{k_3})$$

Figure 3.13 Calculation of intermediate vectors for n $\geq$ 0 [4]

After the calculation, V_n . $V_{n-1} \geq 0$ must be certain to assign main fiber direction from E(r).

3.6.3 Stopping Criteria

In the thesis, 3 stopping criteria are applied by reconstruction fibers method . One of them is FA(fractional anisotropy). The reconstruction of fiber must be terminated when the next fibers are in regoins of low anisotropy where the main eigenvector is not clearly defined (such as gray matter). Thus, the track sign must be changed. In this case, FA be used for terminating reconstruction. FA is a scalar value between 0 and 1 that defines the degree of anisotropy of a diffusion process. A value of zero shows that diffusion is isotropic, i.e. it is unrestricted or evenly restricted in all directions.

A value of one shows that diffusion happens solely throughout one axis and is fully restricted throughout all other directions. We use the following equation to calculate FA [4] [9] :

$$FA = \sqrt{\frac{1}{2}} \sqrt{\frac{(\lambda_1-\lambda_2)^2-(\lambda_2-\lambda_3)^2-(\lambda_3-\lambda_1)^2}{\sqrt{(\lambda_1^2+\lambda_2^2+\lambda_3^2)}}} \qquad (3.16)$$

The other stopping criteria is Vector Angle that is the maximum reacheable angle between the voxels. When the angle that is between two voxel is bigger than the threshold, this is caused by noise or a partial volume effect (such as crossing fibers), so angle treshold is necessary to stop the reconstruction to take accurate reconstruction results at the proper point [9]. Last one is Image boundary: When the fiber hits the boundary of image, the reconstruction algorithm will be stopped.

3.6.4 Fiber Reconstruction Algorithm

The fiber reconstruction algorithm is implemented stepwise. The module gives the result as a fiber list. The function takes seedpoints (x0,y0,z0), interpolated data and data dimensions(xd,yd,zd)

```
recfiber[x0_,y0_, z0_, data_,xd_, yd_, zd_]: =
Module[{x, y, z, vec, k1, k2, k3, k4, i, tlist, list1, list2, vectsign, tensor,
        fa, tlistlength, h, end, end1, dt = data}]
```

In order to try this algorithm over the artifical data, ardata is interpolated. Seedpoint and seedpoints must be determined, The fiber reconstuction algorithm is started by one seed point according to the module e.g. {1.2,1.5,1.7} as follows:

$$\text{fiblist} = \text{recfiber}[1.2,1.5,1.7, \text{interpolated}, 3,3,3]$$

The list can be visualized as the following:

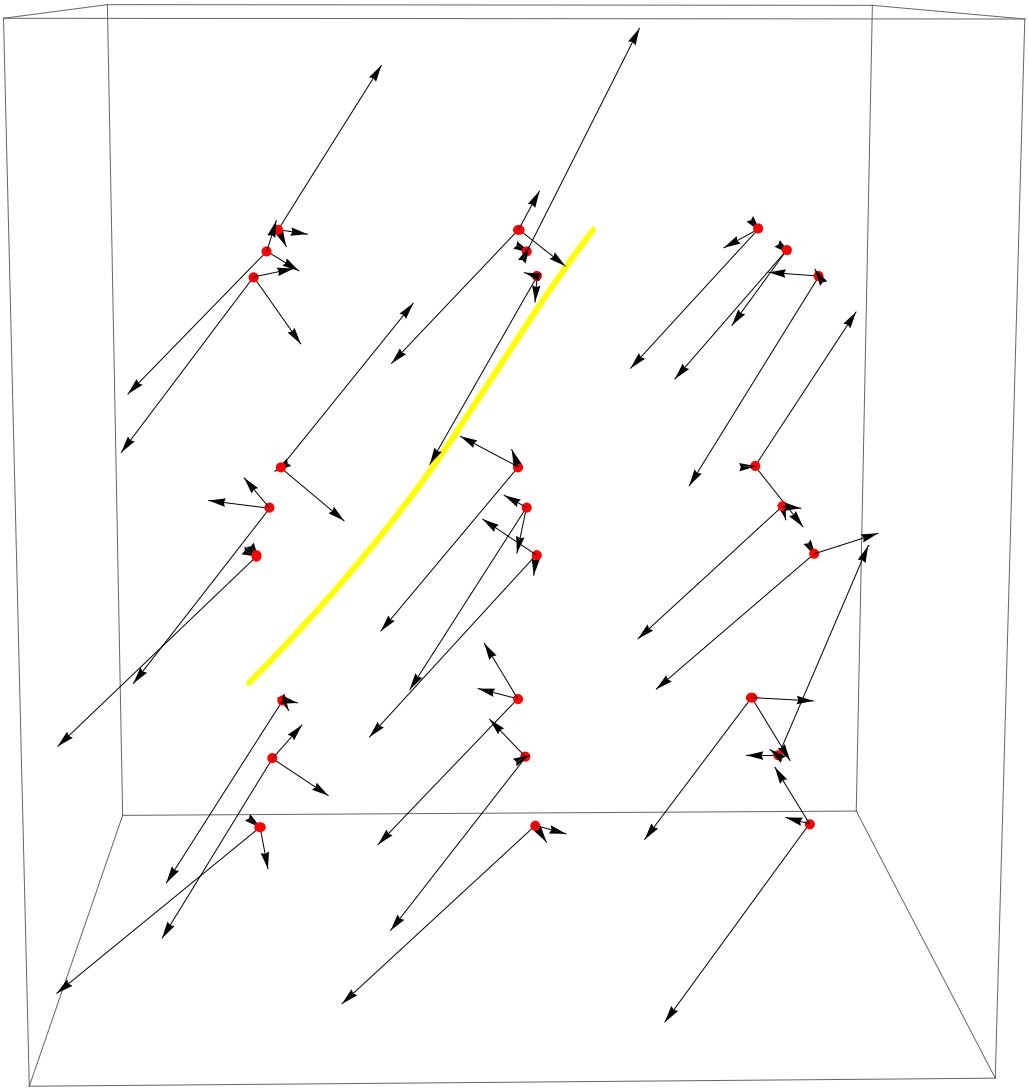

Figure 3.13 The fiber track from one seedpoint due to eigenvectors for artificail data

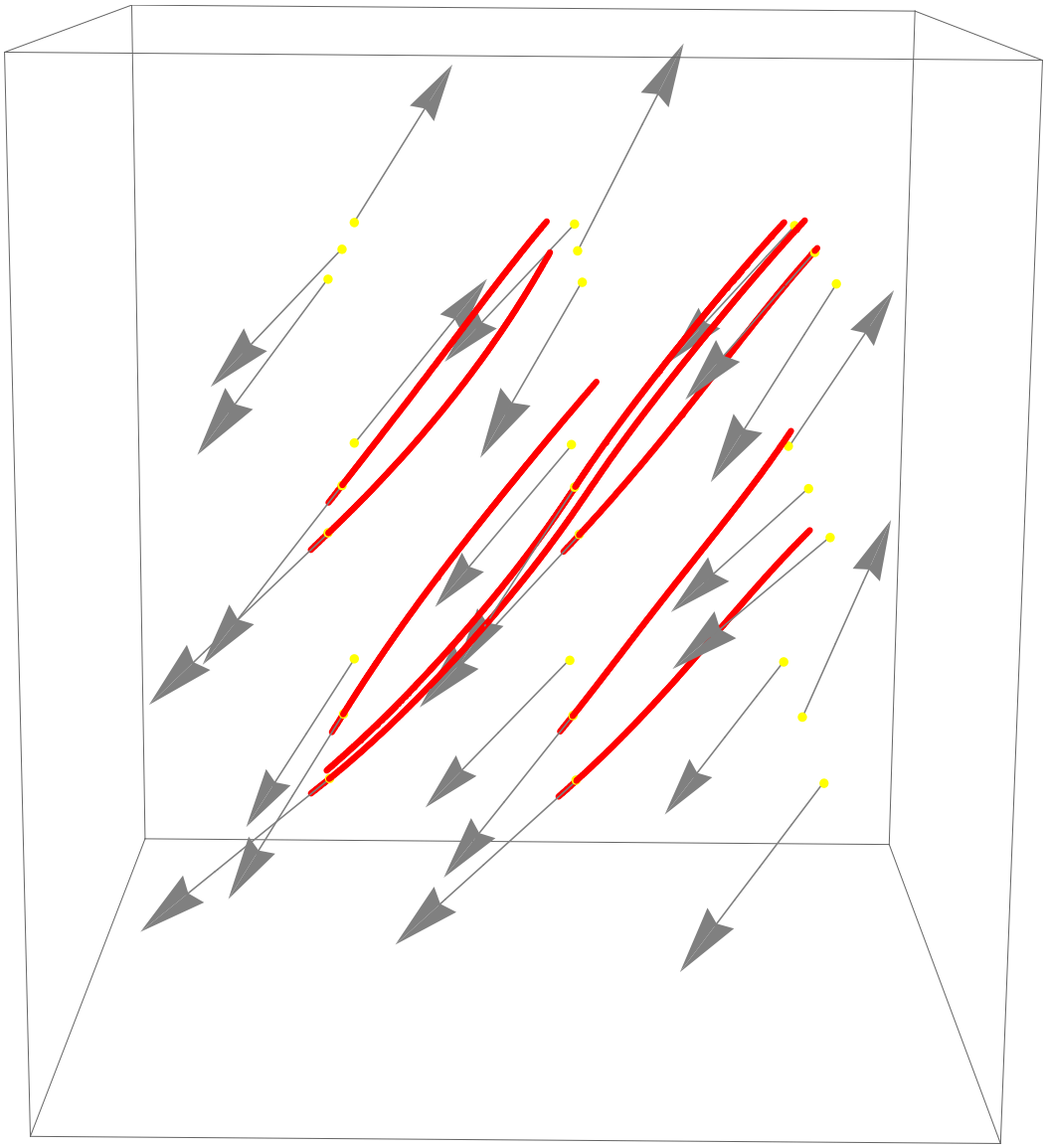

Figure 3.14 For more than one seed points

3.6.5 Track Density Function

The specified areas and deep structures are difficult to visualize and to understand what the underlying brain structures are in the brain. Hence, the new post-processing method, which is called Track Density Imaging is developed in order to enhance spatial resolution. Essentially, The method presents the total number of tracks in each element of a grid and It also helps to solve the crossing and kissing tracks problem of track reconstruction.

To constitute a TDI map, whole-brain track reconstruction is performed (fiber tracking can be performed generated by randomly seed points as initial step). The total number of tracks may be calculated in each element of a grid that should be smaller than the acquired voxel size.

To Create all fibers in specific area accourding to given seedpoints, If the results must be effective, I need to create fiber as much as possible. This method is called Targeted tracktography [12,13]. In particularly, tracking was performed from selected area (ROI). ROI is selected to display the connectivity network [11]. I can give ROI for example start from (1,1,1) to (2,2,2) e.g. a grid. The CreateFib algorithm turns a fiber list according to given seedpoints in specific area. For example, I assume that STN is from (57,57,25) to (58,58,26) or I can examine much smaller or bigger area. The Target Fiber Reconstruction is performed by in a positioned small ROIs. In the thesis, assuming that one voxel size is 1mm. Existing fiber reconstruction algorithm is performed without considering crossing and kissing fibers, under these circumstances to obtain accurate results CreateFib algorithm can be perform grid by grid [11,12,13,14].

CHAPTER 4

RESULTS AND DISCUSSION

As result of the thesis, the tractography algorithm is effectively improved and the technique facilitate accurate identification of the brain to navigate of neurosurgon. How ever, it does not yet run to reconstruct massive amount of fibers. Diffusion Tensor, Eigensystem of Tensor, Principal Direction, visualization of principal direction, reconstruction fibers corresponding to principal directions, reconstruction fibers method (using 4^{th} Runge-Kutta), interpolation (using Trilinear interpolation) and finally new Track Density function are thoroughly focused on as the result of the thesis.

From fiber reconstruction algorithm, fiber tracks can be reconstruted throughout all image from starting given seedpoint. Desceret MRI data canconverted to continuous by trilinear interpolation. While determining a specified region, the tractography algorithm can be run, thus using BsplineCurve, fiber track can be visualized.

Importing the phantomdata (dtnew) :

$$\mathrm{dt} = 10^{10}\mathrm{Import}[\text{"dtnew.mat"}][\![1]\!];$$

Slice of dataset:

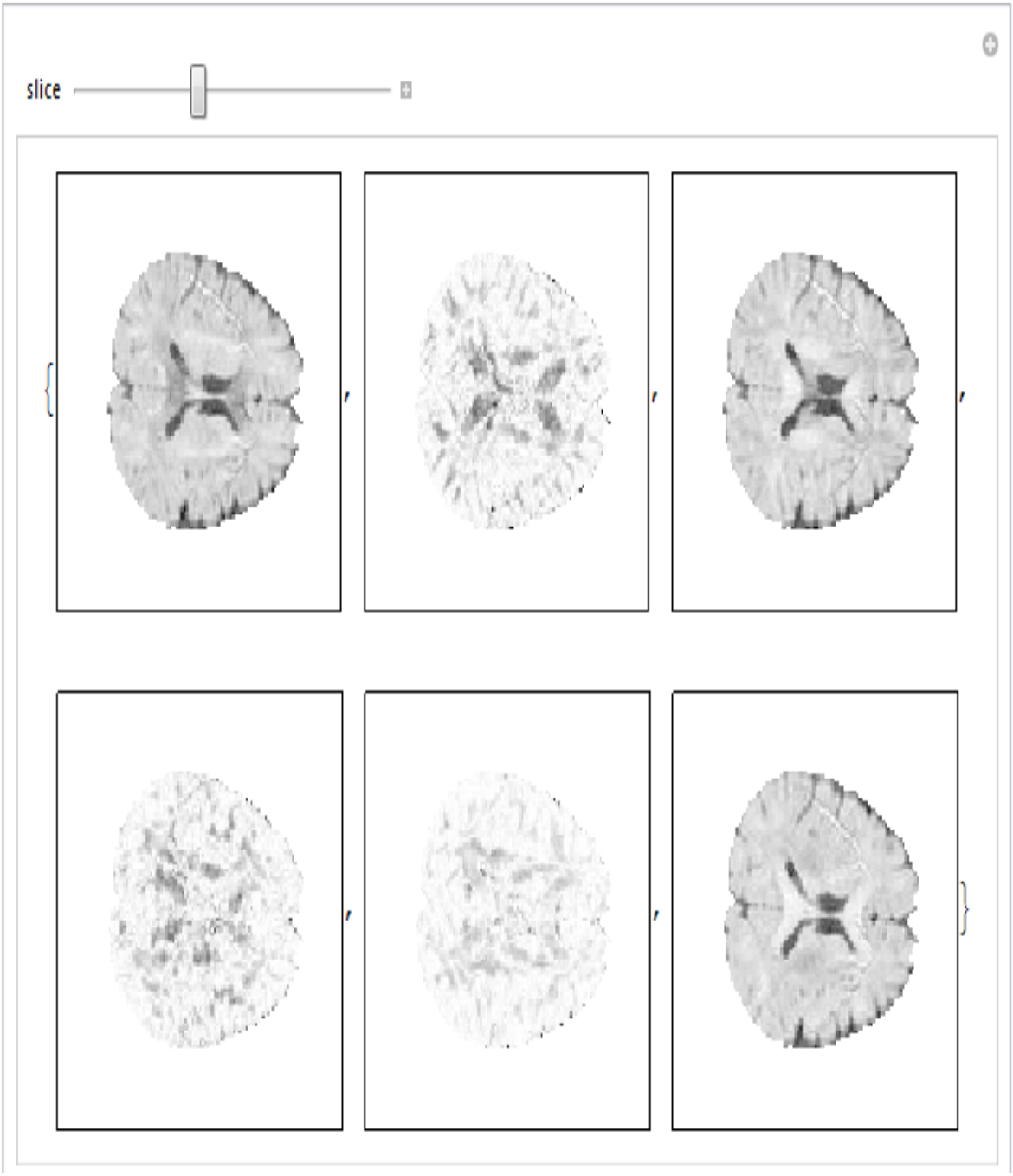

Figure 4.1 Phantom data in slice 25

By written module, The main diffusion direction can be demonstrate as Ellipsiod before converting into continuous field.

ListEllipsoidPlot3D[Transpose[dtdata, {2,3,1}][[24]], ScaleFactor → 3]

Dtdata is discrete phantomdata.

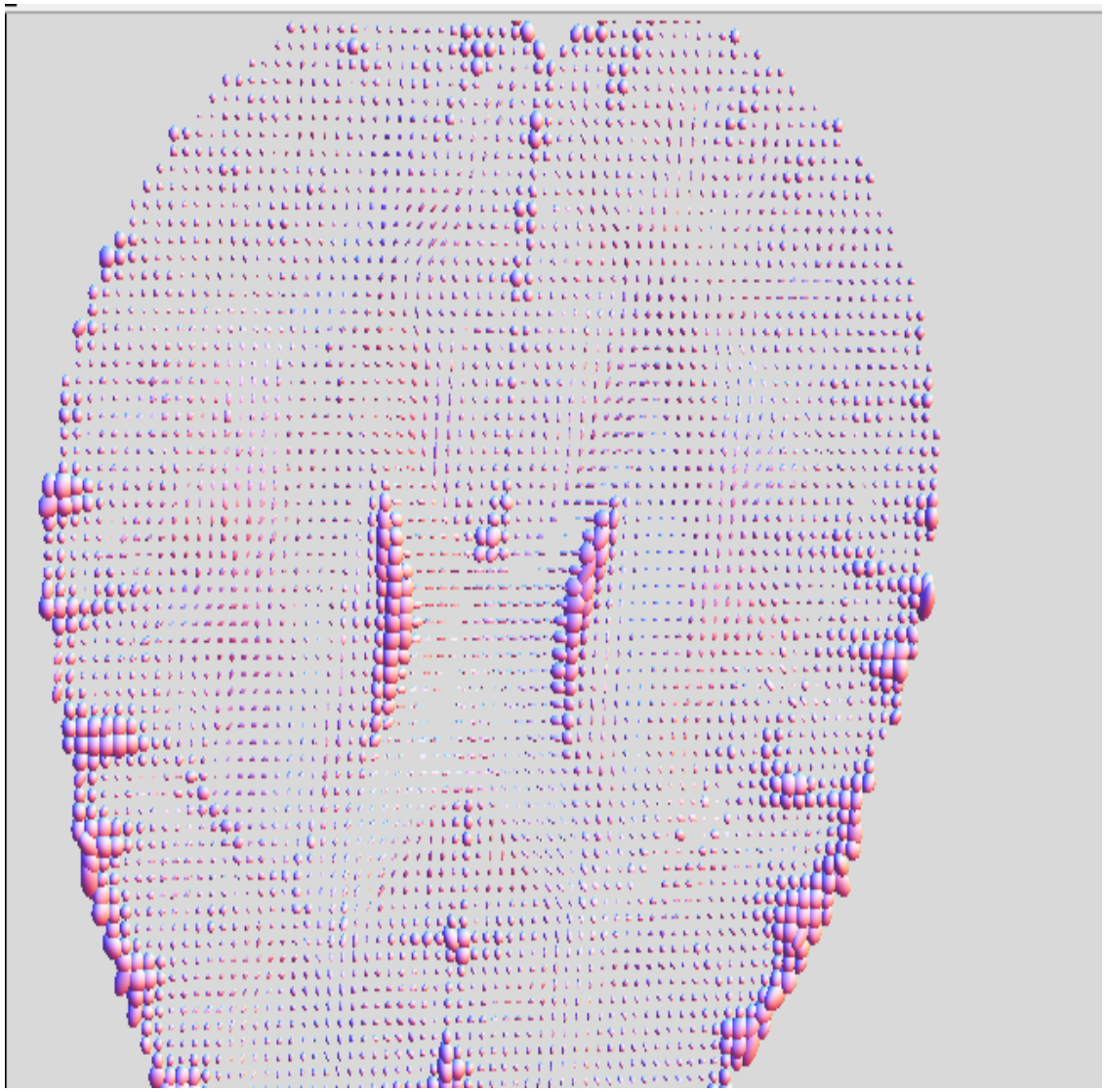

Figure 4.2 Ellipsoid visualization of phantomdata

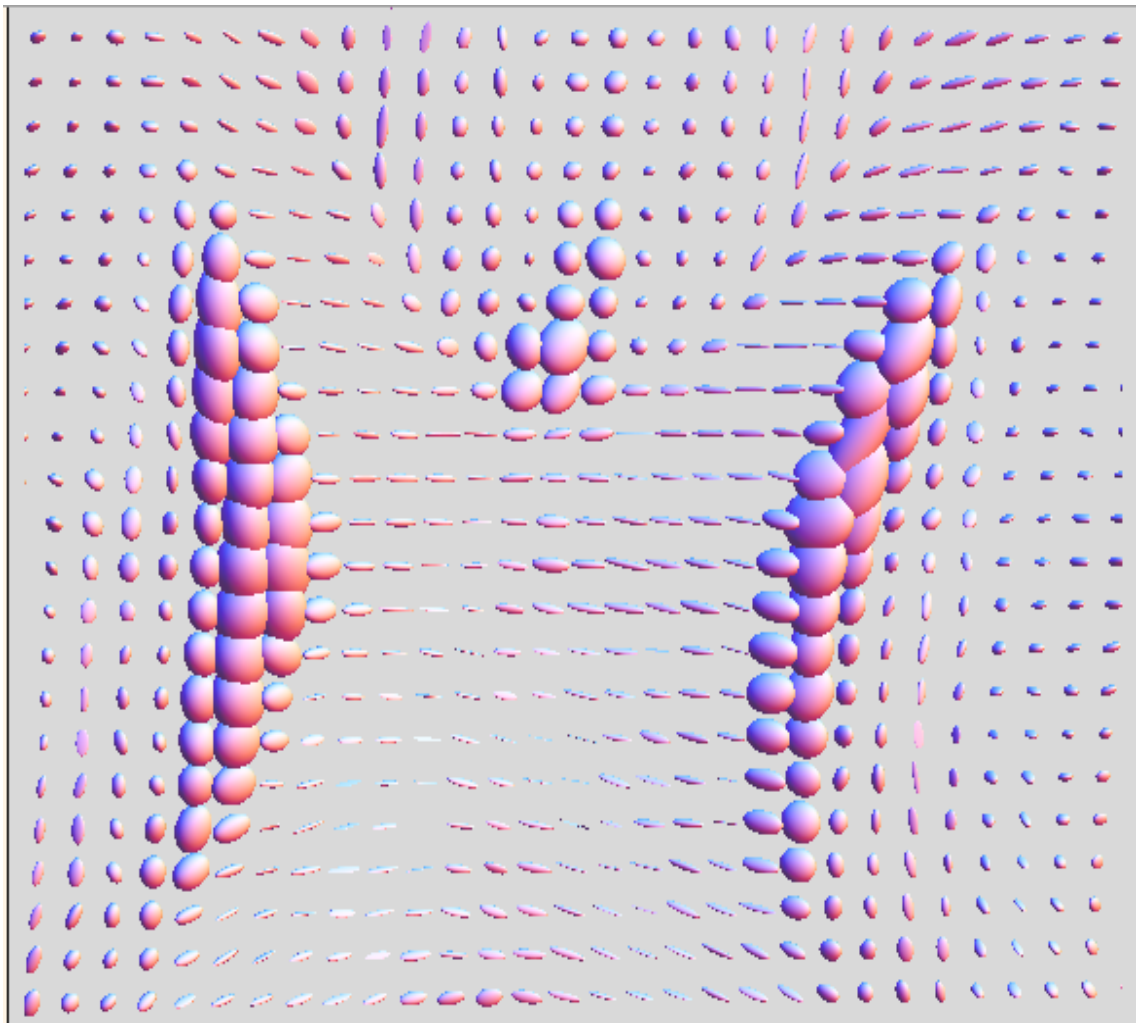

Figure 4.3 Zoom in for closed inspection.

One fiber is reconstructed corresponding to a given seedpoint. Blue point is the seedpoint which is started from { 65.5,62.4,26.4 }. The fiber surrounds the corpus callosum.

Figure 4.4 A fiber from one seedpoint

Giving seed points between specific ranges as knowing Region of Interest. The area starts from {50,50,24} to {53,53,24} as follows :

$$\text{seedpoints} = \text{Flatten}[\text{Table}[\{x, y, z\}, \{x, 50{,}53\}, \{y, 50{,}53\}, \{z, 24{,}24\}], 2];$$

Figure 4.5 the corpus callosum due to human brain atlas

```
Show[Graphics3D[{Thickness[Large], Red, BSplineCurve[#]}]&
      @recfiber[#1, #2, #3, intlocatedTensor, 90,90,50]&"@@@" seedpoints]
```

Figure 4.6 Targetting the thalamus

The fiber list is created by CreateFib module corresponding to spesific area. The number of fibers can be calculated by CreateandCountfib module which counts each element that is list of fibers which are created throughout image and are started from the grid which is within specified ROI.

$$\text{CreateandCountfib}[sx\_, sy\_, sz\_, data\_, vsx\_, vsy\_, vsz\_, vfx\_, vfy\_, vfz\_, xd\_, yd\_, zd\_]:$$
$$= \text{Module}[\{dt$$
$$= \text{data}, \text{tensor}, \text{fa}, i, \text{trackstartpoints}, \text{fiber}, \text{fiberlist}, x, y, z\}]$$

To determine seedpoints which is random numbers :

$$\text{areapoint1} = \text{Flatten}[\text{Table}[\{\text{RandomReal}[\{57,58\}], \text{RandomReal}[\{57,58\}],$$

$$\text{RandomReal}[\{25,26\}]\}, \{x, 1,3\}, \{y, 1,3\}, \{z, 1,3\}],2];$$

$$\text{CountofFibersAllImage}$$
$$= \text{CreateFib}[\#1, \#2, \#3, \text{intpdata}, 57,57,25,58,58,26,112,112,50]\&@@@\text{areapoint1};$$

The lengths are calculated by fibersLengthList module. Because short fibers affects the directionality in a voxel, the short fibers must be ignored for accurate spatial resolution. If they do not ignore, at this time the density will be reduced. The module must be modified considering to fibersLengthLis in next step. To compansate for the reduced intensity involved with length constraint, a much larger number of tracks need to be generated to maintain a decent constrast-to-noise ratio after eliminating short fibers [12-16]as follows;

FibersLengthList[sx_,sy_,sz_,data_,vsx_,vsy_,vsz_,vfx_,vfy_,vfz_,xd_,yd_,zd_]:=

Module[{ListofFibers,NumberofFibers},ListofFibers=CreateFib[sx,sy,sz,data,vsx,vsy,vsz,vfx,vfy,vfz,xd,yd,zd];NumberofFibers=Countfib[ListofFibers];NumberofFibers]
List1=FibersLengthList[#1,#2,#3,intpdata,57,57,25,58,58,26,112,112,50]

&@@@areapoint1

RECOMMENDATIONS

To enhance the TDI method, a significant point of the TDI mapping is the grid size that is made smaller than the acquired vozel size, so, final map is generated at much higher resolution than the original DWI [12-16]. Therefore, I will reveal much smaller grid size from real voxel size. Due to directional encoded color (DEC), TDI maps might be generated. One of the important points, short tracks, TDI method will be modified to eleminate short fibers. Because, the studies show that this aproach increases directional information as compared with the standard technique, so that better visualization will be realized [12-16]. The another improvement step is a new method that is to mapping the average path length of tracktogram (TDI). In this technique, known as average pathlength mapping (APM), a map of the mean length of all streamlines going through a voxel is calculated[12-16].

After improving current fiber reconstruction algorithm, reconstruct the whole-brain fiber tracking, The TDI algorthim can be modified finding points that are along specified grid.

The interface will be written using the Microsoft Visual Studio(MVS) *and C# language*.To write interface, first of all, needs connection to mathematica kernel, Wolfram has efficient dll (dynamic link library) for connection, using Wolfram.NETLink.dll. This library allow for using Mathematica code in MVS or vise versa. In this way, the written Mathematica code, as well and using characteristics of these two programming language will reveal a more efficient study.

REFERENCES

[1] Guus Berenschot, (2003). “Visualization of Diffusion Tensor Imaging”, IBM Journal of Research and Development, master thesis.

[2] Yaniv Assaf & Ofer Pasternak,(2007), “Diffusion Tensor Imaging (DTI)-based White Matter Mapping in Brain Research: A Review”

[3] Rick Wagner,”Multi Interpolation”

[4] Susumu Mori, Peter C.V. van Zijl, (2002) “Fiber tracking:Pricipal and strategy: A technical review”, NMR Biomed. 2002;15:468–480

[5] Hangyi Jiang, Peter C.M. van Zijl, Jinsuh Kim, Godfrey D. Pearlsonc,d, Susumu Mori, (2006) ,” DtiStudio: Resource program for diffusion tensor

computation and fiber bundle tracking”, 8 1 (2 0 0 6) 106–116

[6] Kenneth I. Joy,” Numerical Methods for Particle Tracing in Vector Fields”, Visualization and Graphics Research Laboratory, University of California, Davis

[7] Murat Aksoy,Sancar Adalı, Hasan Ayaz, “Fiber Tractography Using Brain DT-MRI Data and VTK Visualization”

[8] Deep Brain Stimulation http://blogs.nejm.org/now/index.php/deep-brain-stimulation-for-parkinsons-disease/2012/10/19/

[9] Anatomy of the brain, http://www.mayfieldclinic.com/PE-AnatBrain.htm

[10] Joel S. Perlmutter and JonathanW. Mink,(2006),”Deep Brain Stimulation”, 2006. 29:229–57

[11] Susumu Mori,(2007)”Introduction to Diffusion Tensor Imaging” Elsevier Book,

[12] Ellen Brunenberg (2011),”Hitting the right target,noninvasive localization of the subthalamic nucleus motor part for specific deep brain stimulation”, PhD thesis.

[13] Sarah C. Mang, Ania Busza, Susanne Reiterer, Wolfgang Grodd, and Uwe Klose, (2011),” Thalamus Segmentation Based on the Local Diffusion Direction: A Group Study”, 000:000–000 (2011)

[14] Denis Le Bihan, Jean-Franc¸ois Mangin, Cyril Poupon, Chris A. Clark, Sabina Pappata, Nicolas Molko, Hughes Chabriat, (2001),” Diffusion Tensor Imaging: Concepts and Applications”, 13:534–546 (2001)

[15] Bram Stieltjes, Walter E. Kaufmann,Peter C. M. van Zijl,Kim Fredericksen, Godfrey D. Pearlson, Meiyappan Solaiyappan, and Susumu Mori (2001), “Diffusion Tensor Imaging and Axonal Tracking in the Human Brainstem”, 723–735 (2001).

[16] Fernando Calamante, Se-Hong Oh,3,4 Jacques-Donald Tournier, Sung-Yeon Park, Young-Don Son, Jun-Young Chung,3,5 Je-Geun Chi, Graeme D. Jackson, Chan-Woong Park, Young-Bo Kim, Alan Connelly and Zang-Hee

Cho3, (2012) ," Super-Resolution Track-Density Imaging of Thalamic Substructures: Comparison With High-Resolution Anatomical Magnetic Resonance Imaging at 7.0T ", 000:000–000 (2012)

[17] Fernando Calamante, Jacques-Donald Tournier, Nyoman D. Kurniawan, Zhengyi Yang, Erika Gyengesi, Graham J. Galloway, David C. Reutens, Alan Connelly, (2011), "Super-resolution track-density imaging studies of mouse brain: Comparison to histology ", 59 (2012) 286–296

[18] Fernando Calamante, Jacques-Donald Tournier, Graeme D. Jackson, Alan Connelly, (2010), "Track-density imaging (TDI): Super-resolution white matter imaging using whole-brain track-density mapping", 53 (2010) 1233–1243

[19] Fernando Calamante, Jacques-Donald Tournier, Robin M. Heidemann , Alfred Anwander, GraemeD.Jackson,AllenConnelly (2011)," Track density imaging (TDI): Validation of superresolution property".

[20] Barycentric space of anisotropies,http://www.cs.utah./~gk/papers/tvcg00/node4.html

[21] S. Pajevic and C. Pierpaoli, "Color schemes too represent orientation of anisotropic tissue from difuusion tensor data : Application to white matter fiber tract mapping in the human brain.", Magnetic resonance in Medicine, 42:526-540

APPENDICES

APPENDIX A

A-1 Some Mathematica Modules

Fractional Anisotropy module:

$$\text{FAvalue[diftensor\_]} := \text{Module[\{ev\}ev} = \text{Eigenvalues[diftensor]; If[Plus@@ev} = = 0,0.\,, \sqrt{\frac{3\text{Variance[ev]}}{2\text{Plus@@ev}^2}}]]$$

The following module represents the tractography algorithm:

recfiber[x0_,y0_,z0_,data_,xd_,yd_,zd_]:=Module[{x,y,z,vec,k1,k2,k3,k4,i,tlist,list1,list2,vectsign,tensor,fa,tlistlength,h,end,end1,dt=data},

i=0;

h=0.1;

{ [x, 0], [y, 0], [z, 0]}={x0,y0,z0};

[tensor, 0]=dt[[x, 0], [y, 0], [z, 0]];

[fa, 0]=FAvalue[[tensor, 0]];

If[[fa, 0]>= 0.15,

Evaluate[

list1={{ [x, 0], [y, 0], [z, 0]}};

list2={{ [x, 0], [y, 0], [z, 0]}};

While[1<= [x, i]<x d&&1<= [y, i] < yd&&1<= [z, i]<zd,

Evaluate[

```
 [vec, 0]=Eigenvectors[dt[ [x, 0], [y, 0], [z, 0]]][1];


k1 =Normalize[ [vec, i].Eigenvectors[dt[ [x, i], [y, i], [z, i]]][1]]* Eigenvectors[dt[ [x, i], [y, i], [z, i]]][1];

k2 =Normalize[ [vec, i].Eigenvectors[dt[ [x, i]+h/2* k1[1], [y, i]+h/2* k1[2], [z, i]+h/2* k1[3]]][1]* Eigenvectors[dt[ [x, i]+h/2* k[1], [y, i]+h/2* k1[2], [z, i]+h/2* k1[3]]][1];

k3 =Normalize[ [vec, i].Eigenvectors[dt[ [x, i]+h/2* k2[1], [y, i]+h/2* k2[2], [z, i]+h/2* k2[3]]][1]]*  Eigenvectors[dt[ [x, i]+h/2* k2[1], [y, i]+h/2* k2[2], [z, i]+h/2* k2[3]]][1];

k4 = Normalize[ [vec, i].Eigenvectors[dt[ [x, i]+h*k3[1], [y, i]+h* k3[2], [z, i]+h* k3[3]]][1]]* Eigenvectors[dt[ [x, i]+h*k3[1], [y, i]+h* k3[2], [z, i]+h* k3[3]]][1];

 [vec, i+1]=(k1+2k2+2k3+k4)/6;

 [vec, i+1]=If[ [vec, i]. [vec, i+1]<0,(-1)* [vec, i+1], [vec, i+1]];

{ [x, i+1], [y, i+1], [z, i+1]}={ [x, i], [y, i], [z, i]}+h* [vec, i+1];

 [tensor, i+1]=dt[ [x, i+1], [y, i+1], [z, i+1]];

 [fa, i+1]=FAvalue[ [tensor, i+1]];

If[ [fa, i+1]>= 0.015,

If[(0<(VectorAngle[N[ [vec, i+1]],N[ [vec, i]]]*180/[Pi])<70)||(290<=( VectorAngle[N[ [vec, i+1]],N[ [vec, i]]]*180/[Pi])<360),

list1=AppendTo[list1,{ [x, i+1], [y, i+1], [z, i+1]}],Goto[end]],Goto[end]];

i=i+1];

Label[end];

Remove[vec,x,y,z,k1,k2,k3,k4,i];

{ [x, 0], [y, 0], [z, 0]}={x0,y0,z0};

i=0;

While[1<= [x, i ]< xd&&1<= [y, i] <yd&&1<= [z, i]<zd,

Evaluate[

 [vec, 0]=(-1)*Eigenvectors[dt[ [x, 0], [y, 0], [z, 0]]][1];
```

```
k1 =Normalize[ [vec, i].Eigenvectors[dt[ [x, i], [y, i], [z, i]]][1]]* Eigenvectors[dt[ [x, i], [y, i], [z, i]]][1];

k2 =Normalize[ [vec, i].Eigenvectors[dt[ [x, i]+h/2* k1[1], [y, i]+h/2* k1[2], [z, i]+h/2* k1[3]]][1]]* Eigenvectors[dt[ [x, i]+h/2* k1[1], [y, i]+h/2* k1[2], [z, i]+h/2* k1[3]]][1];

k3 =Normalize[ [vec, i].Eigenvectors[dt[ [x, i]+h/2* k2[1], [y, i]+h/2* k2[2], [z, i]+h/2* k2[3]]][1]]*  Eigenvectors[dt[ [x, i]+h/2* k2[1], [y, i]+h/2* k2[2], [z, i]+h/2* k2[3]]][1];

k4 = Normalize[ [vec, i].Eigenvectors[dt[ [x, i]+h*k3[1], [y, i]+h* k3[2], [z, i]+h* k3[3]]][1]]* Eigenvectors[dt[ [x, i]+h*k3[1], [y, i]+h* k3[2], [z, i]+h* k3[3]]][1];

[vec, i+1]=(k1+2k2+2k3+k4)/6;

[vec, i+1]=If[ [vec, i]. [vec, i+1]<0,(-1)* [vec, i+1],[vec, i+1]];

{[x, i+1], [y, i+1], [z, i+1]}={ [x, i], [y, i],[z, i]}+h*[vec, i+1];

[tensor, i+1]=dt[ [x, i+1],[y, i+1], [z, i+1]];

[fa, i+1]=FAvalue[ [tensor, i+1]];

If[[fa, i+1]>= 0.15,

If[(0<(VectorAngle[N[ [vec, i+1]],N[ [vec, i]]]*180/[Pi])<70)||(290<=( VectorAngle[N[ [vec, i+1]],N[ [vec, i]]]*180/[Pi])<360),

list2=AppendTo[list2,{ [x, i+1], [y, i+1], [z, i+1]}],Goto[end1]],Goto[end1]]

];i=i+1];

Label[end1];list1=Reverse[list1];list2=Drop[list2,1];tlist=Join[list1,list2]],{{0,0,0},{0,0,0});

tlistlength=(Length[tlist]-1)*h;

If[tlistlength>=1,tlist,{{0,0,0},{0,0,0}}]]
```

Visualization Module:

```
VisualizationEigVec[tensor_,point_]:=

Module[{evec,eval},{eval,evec}=Eigensystem[tensor];

{Tooltip[Arrow[{point,point+#1*#2}],Row[{#1,MatrixForm[#2]}]]&

@@@Transpose[{Normalize[eval],Normalize/@evec}],Red,PointSize[0.01],Point[point}

];
```

```
Graphics3D[MapIndexed[VisualizationEigVec, ardata, {3}], FaceGrids → All]
```

To Create RandomReal Tensor:

Function takes row and column numbers then return a RandomReal Tensor

```
CreateRandomRealTensor[row_,column_]:= Module[{r = row,c =
column,tensor},If[r==c, tensor = Table[RandomReal[],{r},{c}],False]];
```

To Create RandomReal and Symmetric Tensor:

Function takes row and column numbers then return a RandomReal Symmetric Tensor:

```
CreateSymRandomRealTensor[row_,column_]:=
Module[{r = row,c = column,tensor,symtensor},
If[r==c, tensor = Table[RandomReal[],{r},{c}],False];symtensor =
tensor+Transpose[tensor];If[SymmetricQ[symtensor],
Return [symtensor],False]];

PhantomSymmetricTensor2D[xdim_,ydim_]:=
Module[{x=xdim,y=ydim},Table[CreateSymRandomRealTensor[3,3],{xd,1,x},{yd,1,y
}]];
```

To create 3D phantomdata x,y and z by Tensor 3x3
Functions takes dimension x ,y and z then return 3D phantomdata by Tensor:

```
PhantomSymmetricTensor3D[xdim_,ydim_,zdim_]:=Module[{x=xdim,y=ydim,z=zdim
},Table[CreateSymRandomRealTensor[3,3],{xd,1,x},{yd,1,y},{zd,1,z}]];
```

To calculate All Eigenvectors :
Function calculate each pixel or voxel Eigenvectors

```
CalEigenvectorsAllpdata[phantomdata_]:=
Module[{phd= phantomdata,ev},ev =N[Map[Eigenvectors[#]&,phd,{2}]];Return[ev]];
```

```
CreateFib[sx_, sy_, sz_, data_, vsx_, vsy_, vsz_, vfx_, vfy_, vfz_, xd_, yd_, zd_]:
= Module[{dt = data, tensor, fa, i, trackstartpoints, fiber, fiberlist, x, y, z}, i
= 0; {x0, y0, z0} = {sx, sy, sz}; tensor0 = dt[x0, y0, z0]; fa0
= FAvalue[tensor0]; fiberlist = {{0,0,0}}; If[fa0
≥ 0.15, Evaluate[trackstartpoints{{x0, y0, z0}}; fiber0
= recfiber[x0, y0, z0, data, xd, yd, zd]; fiberlist = If[Length[fiber0] ≥ 1, fiberlist
= AppendTo[fiberlist, fiber0], fiberlist = AppendTo[fiberlist, {{0,0,0}}]]; While[vsx
≤ xi ≤ vfx&&vsy ≤ yi ≤ vfy&&vsz ≤ zi
≤ vfz, Evaluate[If[recfiber[xi, yi, zi, data, xd, yd, zd][[1]][[1]] == 0; &fiberlist
= AppendTo[fiberlist, fiberi = recfiber[xi, yi, zi, data, xd, yd, zd]]]; ]; i
= i + 1]; fiberlist
= Drop[fiberlist, {1}]; ](*Evaluationend*), {{0}, {0}}](*Ifend*); fiberlist
= Drop[fiberlist, {1}]; fiberlist]


CreateTensorMatrix[x0_, y0_, z0_, importedData_] := Module[{x1 = x0, y1 = y0, z1
= z0, data = importedData}, Table[{{#[[x, y]]&@data[[1,1, z]],
#[[x, y]]&@data[[1,2, z]], #[[x, y]]&@data[[1,4, z]]}, {#[[x, y]]&@data[[1,2, z]], #[[x, y]]
&@data[[1,3, z]],
#[[x, y]]&@data[[1,5, z]]}, {#[[x, y]]&@data[[1,4, z]], #[[x, y]]&@data[[1,5, z]], #[[x, y]]
&@data[[1,6, z]]}},{x, 1, x1}, {y, 1, y1}, {z, 1, z1}]]


TensorLocation[x0_, y0_, z0_, data_]:= Module[{dt = data, x = x0, y = y0, z
= z0, points, points3D, fdata, pointandtensor}, points
= Table[{i, j, t}, {i, 1, x}, {j, 1, y}, {t, 1, z}]; points3D
= Flatten[points, 2]; fdata = Flatten[dt, 2]; pointandtensor
= Table[{points3D[[z]], fdata[[z]]}, {z, 1, (112 * 112 * 50)}]]
```

```
BilinearInterpolationFun[va0_, vb0_, vc0_, vd0_, seedpoint0_, pa0_, pb0_, pc0_, pd0_]:
        = Module[{va = va0, vb = vb0, vc = vc0, vd = vd0, sdp
        = seedpoint0, pa = pa0, pb = pb0, pc = pc0, pd
        = pd0, vecAB, vecCD, vecSdp, resultvec}, vecAB
        = (pb[[1]] − sdp[[1]]) ∗ va + (sdp[[1]] − pa[[1]]) ∗ vb; vecCD
        = (pc[[1]] − sdp[[1]]) ∗ vd + (sdp[[1]] − pd[[1]]) ∗ vc; vecSdp
        = (pc[[2]] − sdp[[2]]) ∗ vecAB + (sdp[[2]] − pb[[2]])
        ∗ vecCD; resultvec = Flatten[vecSdp, 1]]


FuncTrilinearInterpolationDeneme[{x0_, y0_, z0_, v01_, v02_, v03_, v04_, v05_, v06_, v07_, v08_}]:
= Module[{x = x0, y = y0, z = z0, v1 = v01, v2 = v02, v3 = v03, v4 = v04, v5
= v05, v6 = v06, v7 = v07, v8 = v08, V1, V2}, V1
= (v1(1 − x)(1 − y)(1 − z)) + (v2x(1 − y)(1 − z)) + (v3(1 − x)y(1 − z))
+ (v4(1 − x)(1 − y)z); V2
= (v5x(1 − y)z) + (v6(1 − x)yz) + (v7xy(1 − z)) + (v8xyz); Return[V1 + V2]];

AlRungeKutta[xs_, ys_, zs_, data_, stepsize_] := Module[{dt = data, x = xs, y = ys, z
= zs, eval, evec, l1, l2, l3, v1, v2, v3, stps, k1, sdp2, x2, y2, z2, k2, sdp3, x3, y3, z3, k3,
sdp4, x4, y4, z4, k4, finalpoint, xf, yf, zf, evec2, evec3, evec4, evec5}, {eval, evec}
        = Eigensystem[dt[x, y, z]]; {l1, l2, l3} = eval; {v1, v2, v3} = evec; k1
        = stps ∗ v1; sdp2 = {x, y, z} + 1/2 ∗ k1; {x2, y2, z2} = sdp2; evec2
        = Eigenvectors[dt[x2, y2, z2]][[1]]; k2 = stps ∗ evec2; sdp3
        = {x, y, z} + 1/2 ∗ k2; {x3, y3, z3} = sdp3; evec3
        = Eigenvectors[dt[x3, y3, z3]][[1]]; k3 = stps ∗ evec3; sdp4
        = {x, y, z} + k3; {x4, y4, z4} = sdp4; evec4
        = Eigenvectors[dt[x4, y4, z4]][[1]]; k4 = stps ∗ evec4; finalpoint
        = {x, y, z} + 1/6 (k1 + k2 + k3 + k4); {xf, yf, zf} = finalpoint; evec5
        = Eigenvectors[dt[xf, yf, zf]][[1]]];
```

CURRICULUM VITAE

CONTACT

Home Adress : Akpınar mh. Açı sk. No:3
Osmangazi, Bursa, Turkey

Phone Number :+90 505 892 30 30

E-Mail : ruhunur@gmail.com

EDUCATION

Ruhunur Özdemir was born on August 19, 1984, in Giresun,Turkey. She finished high school education at Gülbahar Hatun Science and Anatolia High School as a boarder, in 2003, Trabzon. She started studying Computer Information System at Near East University, in Nicosia, Cyprus. After two years, subsequently, She started studying Computer Engineering at Bahçeşehir University from which she obtained Bachelor of Science degree in 2009. She worked as an software engineer at Provincial Health Directorate of Istanbul for two years. She started her Master of Science degree, Biomedical Engineering, at Institute of Biomedical Engineering, Fatih University, in 2011. During her master study, she performed an internship at Biomedical Image Analysis Laboratory, Eindhoven University of Technology in Eindhoven, The Netherlands where she focused on her master thesis, Design a hyperresolution fiber tracking system for optimal navigation by the neurosurgeons in deep brain stimulation.

ACTIVITIES

During high school education, She played volleyball and table tennis, She was a member of chess club to tutor juniors. In the university, she was a vocalist at Bahçeşehir University Turkish Music Society. She has performed Girl Ney. She is foil fencer and former member of Hoc Hobet Fencing Club Eindhoven.