

**FIBERWEB : VISUALISATION ET INTERACTION DANS
LE WEB EN IMAGERIE PAR RÉSONANCE
MAGNÉTIQUE DE DIFFUSION**

par

Louis-Philippe Ledoux

Mémoire présenté au Département d'informatique
en vue de l'obtention du grade de maître ès sciences (M.Sc.)

FACULTÉ DES SCIENCES
UNIVERSITÉ DE SHERBROOKE

Sherbrooke, Québec, Canada, 27 février 2017

Le 27 février 2017

*le jury a accepté le mémoire de Monsieur Louis-Philippe
Ledoux dans sa version finale.*

Membres du jury

Professeur Maxime Descoteaux
Directeur de recherche
Département d'informatique

Professeur Kevin Whittingstall
Codirecteur de recherche
Département de radiologie diagnostic

Professeur Pierre-Marc Jodoin
Membre interne
Département d'informatique

Professeur Djemel Ziou
Président-rapporteur
Département d'informatique

Sommaire

Ce mémoire présente une nouvelle application web permettant de visualiser et d'interagir avec des données d'imagerie médicale provenant principalement de l'imagerie par résonance magnétique de diffusion, une technique non-invasive permettant d'explorer la matière blanche du cerveau. Dans le premier chapitre, un historique de la diffusion, l'explication de divers concepts importants du domaine ainsi qu'un survol des méthodes de visualisation seront présentés. Le deuxième chapitre présentera le web, et plus précisément la technologie qui a rendu possible l'existence du Fiberweb : WebGL. Des choix de design du Fiberweb seront également abordés. Le troisième chapitre est la contribution principale de ce mémoire. Il consiste en un article présentant le Fiberweb, ses fonctionnalités et également une nouvelle technique de tractographie en temps réel probabiliste. Finalement, les diverses problématiques non-résolues liées à notre application, ses utilisations potentielles et ses perspectives futures seront discutées.

Mots-clés: Imagerie par résonance magnétique de diffusion, visualisation, interaction, web, tractographie, temps réel.

Remerciements

Je tiens tout d'abord à remercier mes directeurs de recherche, Maxime Descoteaux et Kevin Whittingstall. Grâce à eux, j'ai eu l'occasion de pouvoir explorer le potentiel du web et de mener à terme un superbe projet. Ce n'est pas tous les jours qu'on a la chance de se faire offrir un projet de maîtrise comme celui du Fiberweb.

J'aimerais ensuite remercier mes collègues de l'université qui m'ont aidé tant au plan personnel que scolaire : Alessandro Daducci, Alexandre Gauvin, Caroline Presseault, Clément Zotti, Eleftherios Garyfallidis, Emmanuelle Renauld, Étienne St-Onge, François Rheault, Gabriel Girard, Jasmeen Sidhu, Jean-Christophe Houde, Marc-Alexandre Côté, Maxime Chamberland, Michael Bernier, Michael Paquette, Nghi Huynh, Olivier Godin, Patrick Bellavance Marcoux, Samuel Deslauriers-Gauthier, Samuel St-Jean, Seyed Mohammad Havaei, Vincent Méthot, Yi Wang et Zhiming Luo.

J'aimerais remercier tous les membres d'Iméka. Cette maîtrise n'aurait pas été possible sans leur soutien financier, mais surtout sans leur expertise. Je tiens à remercier spécifiquement ceux qui n'ont pas été mentionné auparavant : Jean-René Bélanger, Matthieu Dumont et Nil Goyette.

J'aimerais également remercier mes parents, ma soeur et ma copine pour leur soutien tout au long de cette aventure. Ils ont contribué énormément à rendre ma maîtrise une expérience positive et ont également contribué à la correction de ce mémoire (merci maman!).

Un très grand merci à un trio très spécial : Bern, Charlie et Martin. Ces trois personnes m'ont soutenu tout au long de ma maîtrise. J'ai vécu des moments inoubliables avec eux, et j'espère pouvoir poursuivre les "expériences Tara" pour de longues années à venir.

Finalement, je réserve mon plus grand merci à Félix C. Morency. Il a été bien

REMERCIEMENTS

plus qu'un superviseur technique à l'intérieur d'Iméka. Il m'a donné la motivation de continuer et persévérer dans les moments les plus difficiles de ma maîtrise. Il a été un mentor ainsi qu'un ami sur qui j'ai pu compter en tout temps. Sans lui, cette maîtrise n'aurait jamais été menée à terme. Félix, un énorme, un gigantesque merci !

Abréviations

2D Deux dimensions

3D Trois dimensions

AF Arcuate Fasciculus

AFQ Algorithme de quantification automatique de fibres (automated fibers quantification)

AVC Accident vasculo-cérébral

API Interface de programmation applicative (application programming interface)

CC Corpus callosum

CST Corticospinal tract

DTI Imagerie du tenseur de diffusion (diffusion tensor imaging)

FA Anisotropie fractionnelle (fractional anisotropy)

fODF Fiber orientation distribution function

IRM Imagerie par résonance magnétique

IRMd Imagerie par résonance magnétique de diffusion

ODF Fonction d'orientation de distributions (orientation distribution function)

RGB Rouge, vert, bleu (red, green, blue)

RTT Tractographie en temps réel (real-time tractography)

SH Harmonique sphérique (spherical harmonic)

W3C World Wide Web Consortium

WHATWG Web Hypertext Application Technology Working Group

Table des matières

Sommaire	i
Remerciements	ii
Abréviations	iv
Table des matières	v
Liste des figures	viii
Liste des tableaux	xi
Introduction	2
1 Imagerie de diffusion et tractographie	6
1.1 Historique de la diffusion	6
1.1.1 Tenseurs	7
1.1.2 Fonction d’Orientation de Distributions (ODF)	11
1.1.3 Tractographie	13
1.2 Visualisation de la diffusion	17
1.2.1 Visualisation d’images	17
1.2.2 Visualisation de glyphes	17
1.2.3 Visualisation de streamlines	19
1.2.4 Visualisation de tractogrammes	20
1.2.5 Visualisation de faisceaux	21
1.2.6 Visualisation de connectomes et graphes	22

TABLE DES MATIÈRES

2	Visualisation dans le web	26
2.1	Visualisation dans le web	26
2.1.1	HTML5	26
2.1.2	JavaScript	27
2.1.3	Canvas3D	27
2.1.4	WebGL	28
2.1.5	Revue de différentes boîtes à outils	29
2.2	Design général	30
2.3	Logiciels pré-existants	31
2.3.1	Slice :Drop	31
2.3.2	Brain Networks	32
2.3.3	Article interactif	32
2.3.4	AFQ Browser	33
3	Fiberweb	36
3.1	Introduction	39
3.2	Methods	40
3.2.1	Visualization in the web	40
3.2.2	Interactions and processing	43
3.3	Results	49
3.3.1	Visualization	49
3.3.2	Streamline interactions	50
3.3.3	Probabilistic tractography : online vs offline	52
3.4	Discussion	53
3.4.1	New probabilistic algorithm	53
3.4.2	Applications	56
3.4.3	Future work	57
3.5	Conclusion	58
	Conclusion	61
	Contributions	61
	Problèmes ouverts et non-résolus	61
	Conclusion générale	63

TABLE DES MATIÈRES

A Fiberweb : a new web solution for medical visualization and inter-	
actions	65
B Supplemental data	68
Supplementary code	68
Weighted Dice coefficient	77

Liste des figures

1	(a) Le cerveau humain (adapté de Cephalicvein [10]). (b) Dissection du cerveau par Klingler (adapté de Türe et al. 2000 [57]) (c) Exemple d'une dissection virtuelle du faisceau "occipito-frontal fasciculus" d'un autre faisceau, le "uncinate fasciculus (adapté de Catani 2002).	3
1.1	Image pondérée de la diffusion d'un patient ayant subi un AVC, 1 heure (gauche) et 3 heures (droite) après l'accident (adapté de Moseley 1990 [44]).	7
1.2	6 images représentant des directions différentes d'une même tranche axiale du cerveau (adapté d'une présentation de Descoteaux 2014 [15]).	8
1.3	Représentation d'un tenseur. À gauche, on peut voir le mouvement brownien de l'eau (adapté de Poupon 1999 [46]). À droite, on peut observer le tenseur de diffusion et ses valeurs propres (adapté de Descoteaux 2008 [14]).	9
1.4	Représentation d'un croisement de fibres dans la matière blanche par le tenseur de diffusion, adapté de Descoteaux et Poupon 2014 [18]. . .	10
1.5	Harmoniques sphériques (seulement la partie réelle) jusqu'à leur 3ème ordre (adapté de Descoteaux 2008 [14]). L'harmonique sphérique d'ordre zéro est bleue car elle est constante sur la sphère. Le reste des couleurs illustre les valeurs maximales (rouge) et minimales (bleu).	11
1.6	Représentation d'un signal de base par une approximation faite à l'aide de la convolution sphérique, adapté de Kazhdan 2015 [35].	12

LISTE DES FIGURES

1.7	Représentation d'un croisement de fibres dans la matière blanche, adapté de Descoteaux et Poupon 2012 [18]. (a) Déplacement de l'eau (b) Directions principales (c) Tenseur de diffusion (d) ODF (e) fODF.	13
1.8	Illustration d'un faisceau de fibres issu de la tractographie.	14
1.9	Illustration de la sélection de la direction principale à partir d'un champ de tenseurs de diffusion. À gauche, la tractographie déterministe et à droite, la tractographie probabiliste (adapté de Descoteaux et Poupon 2014 [18]).	15
1.10	Vue d'une coupe axiale de la FA (gauche) et d'une carte RGB (droite)(adapté de Descoteaux 2008 [14]).	18
1.11	Anatomie représentée à la fois en 3D et en 2D selon les trois axes (axial, coronal et sagittal).	18
1.12	Différents types de glyphes.	19
1.13	Illustration simplifiée d'une "streamline" à l'aide de points (a) et le faisceau résultant, composé de multiples "streamlines".	20
1.14	Tractogramme d'un cerveau complet (66,540 fibres).	21
1.15	Exemple d'un ensemble de faisceaux du cerveau (adapté de Klein 2009).	21
1.16	Divers faisceaux du cerveau.	23
1.17	Exemple d'un graphe illustrant toutes les connections entre les zones du cerveau pour un sujet (adapté de Irimia 2012 [32]).	24
2.1	Rendu d'une théière avec Canvas3D (adapté de Vukićević 2007 [59]).	28
2.2	Diagramme représentant le design général du Fiberweb.	30
2.3	Slice :Drop.	31
2.4	Brain Networks.	32
2.5	Version interactive de l'article de Friederici.	33
2.6	AFQ Browser.	34
3.1	Anatomy visualization in 3D and 2D views inside Fiberweb.	41
3.2	Anatomy and peaks visualization in 3D and 2D views inside Fiberweb.	42
3.3	Anatomy and streamlines visualization inside Fiberweb, with the intersected streamlines in the 2D views.	43

LISTE DES FIGURES

3.4	Fiber bundle (part of the corpus callosum), result of the real-time tractography, with the initialization box (red) in the mid-sagittal slice of the CC and the following parameters : angle threshold of 50 deg, fiber length minimum of 10mm and maximum of 200mm, FA threshold of 0.1, a step size of 0.5mm, 10 seeds per axis and a puncture value (g) of 0.2.	45
3.5	Fiberweb menu and parameters.	46
3.6	Illustration of the uncertainty alpha parameter at 95%, 75% and 35% of the maxima. The script to compute these alpha values for each fODF peak is given in the Appendix.	47
3.7	Mean framerate (in frames per second (fps)) versus number of streamlines.	50
3.8	Various bundles resulting from probabilistic tractography (real-time in the Fiberweb vs offline). Dice coefficients are : CST (0.865), CC (0.928) and AF (0.811). The yellow arrows show the location of seeding regions.	51
3.9	Various bundles resulting from probabilistic tractography, each with a different uncertainty percentage.	54
3.10	Probabilistic RTT of the CST for different puncture values.	55
3.11	fODF represented by spheres of different resolutions.	56

Liste des tableaux

3.1	Comparison between offline and online algorithms.	52
-----	---	----

Don't let your dreams be dreams. Just do it.

SHIA LABEOUF

Introduction

Le cerveau est l'un des organes les plus complexes du corps humain et c'est également l'un des plus importants (figure 1). Il est l'équivalent d'un ordinateur biologique, tout en étant plus versatile que cette machine. Composé de plusieurs milliards de neurones, le cerveau est un organe qui possède toujours de nombreux mystères. Il était donc tout naturel que l'être humain se soit décidé à en connaître plus à son sujet. L'une des méthodes employée afin d'explorer le cerveau est la dissection (figure 1). Cependant, c'est une technique qui ne peut être pratiquée sur une personne vivante. Grâce à l'imagerie par résonance magnétique (IRM), il est possible de le disséquer virtuellement. On prend tout d'abord des données d'IRM de diffusion (IRMd). Ces données nous permettent de faire de la tractographie afin de recréer un tractogramme, c'est-à-dire une reconstruction des fibres du cerveau. Par la suite, on peut utiliser des boîtes de sélection et d'exclusion de fibres afin de faire la dissection virtuelle. On peut en voir un exemple tiré de Catani et al. [8] à la figure 1. Il existe plusieurs logiciels permettant de faire ce type de dissection, comme le FiberNavigator [58] ou MI-Brain [31].

Au cours des dernières années, l'intérêt envers les technologies du web et du mobile a bondi. Généralement, on aime avoir accès à l'information que l'on soit à son bureau, à la maison, ou même lors d'un déplacement entre les deux. Cependant, dans le cas de l'imagerie médicale, peu de logiciels sont disponibles permettant de tirer profit de ces technologies (quelques exemples seront présentés au deuxième chapitre). De plus, ces logiciels sont seulement faits pour de la visualisation de données. Si l'on veut faire de la dissection virtuelle tel que mentionné précédemment, ce n'est tout simplement pas possible. En effet, pour pouvoir effectuer cette dissection, on a besoin de faire des calculs d'intersection et des tests logiques sur nos données, ce qui n'est tout simple-

INTRODUCTION

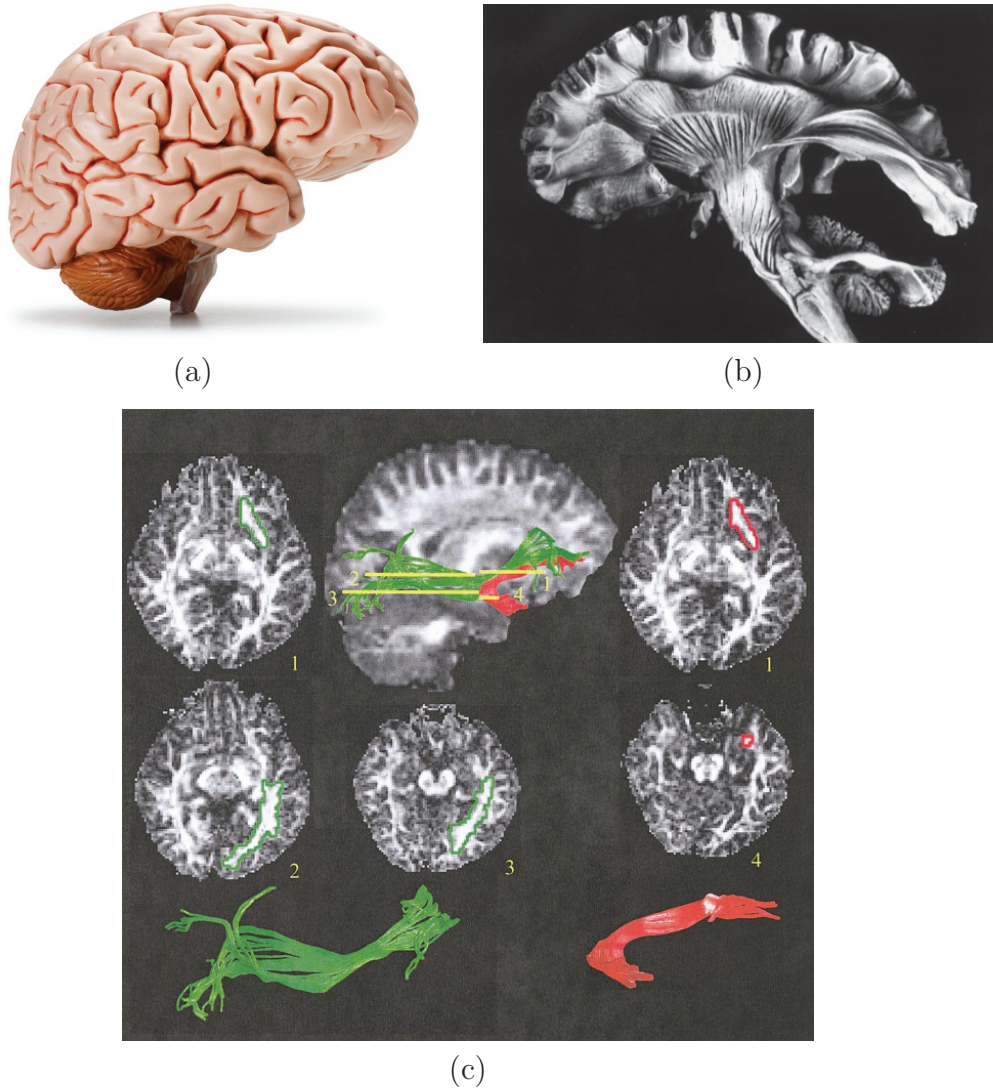


figure 1 – (a) Le cerveau humain (adapté de Cephalicvein [10]). (b) Dissection du cerveau par Klingler (adapté de Türe et al. 2000 [57]) (c) Exemple d'une dissection virtuelle du faisceau "occipito-frontal fasciculus" d'un autre faisceau, le "uncinatus fasciculus" (adapté de Catani 2002).

ment pas permis par ces applications. Nous avons donc décidé de combler cette lacune en développant notre propre application de visualisation d'imagerie médicale (qui est

INTRODUCTION

la contribution principale de ce mémoire) : le Fiberweb. Cette application utilise le potentiel du web afin de visualiser, mais aussi d'interagir avec des données. L'une des interactions possibles du Fiberweb est un nouvel algorithme de tractographie en temps réel probabiliste.

Dans ce mémoire, le premier chapitre survolera l'IRMd en faisant un petit historique de ce domaine. On parlera tout d'abord de divers concepts nécessaires pour comprendre le concept de tractographie en imagerie de diffusion. On partira de données qu'on peut obtenir grâce à l'IRMd, pour ensuite voir quelle information on peut en tirer, et finalement comment en arriver à reconstruire la matière blanche du cerveau, que la méthode soit en temps réel ou non. De plus, une fois l'historique terminé, on verra ces divers concepts en images et comment ils sont généralement visualisés.

Dans le second chapitre, il y aura un résumé de ce qu'est la visualisation dans le web, de ses débuts jusqu'à aujourd'hui. On verra entre autres d'où vient l'idée de faire de la visualisation en trois dimensions dans un navigateur internet et ce qui a permis à la mise en oeuvre de cette idée. Pour poursuivre, on en apprendra davantage sur la technologie derrière le Fiberweb : WebGL. Finalement, certaines décisions par rapport au développement de l'application, ainsi que le design général seront discutés.

Dans le troisième chapitre, on trouvera la contribution principale de ce mémoire. C'est un article qui sera soumis en vue d'une publication dans la revue "Frontiers : NeuroInformatics". Cet article présente entre autre une nouvelle application web permettant non seulement de visualiser des données issues de l'IRMd et de la tractographie, mais également d'interagir avec elles. De plus, un nouvel algorithme de tractographie probabiliste en temps réel sera introduit. Par la suite, il y aura une discussion sur la validité des résultats obtenus et l'impact qu'aura cette nouvelle application de visualisation.

Pour conclure ce mémoire, il y aura d'abord un retour sur les diverses contributions qui ont été apportées par cette maîtrise. On verra ensuite les divers problèmes non-résolus ainsi que les nouveaux qui ont été mis en lumière tout au long de ce projet. On finira par une discussion sur l'utilisation typique du Fiberweb, son intérêt par rapport aux autres logiciels, ainsi que du futur qui attend l'application.

*The important thing is to not stop questioning. Curiosity has
its own reason for existing.*

ALBERT EINSTEIN

Chapitre 1

Imagerie de diffusion et tractographie

Ce chapitre survolera d'abord l'histoire de l'IRMd. On y verra entre autres plusieurs concepts de ce domaine permettant éventuellement de faire de la tractographie. Puis, on verra des exemples de visualisation de ces différents concepts, puisqu'ils permettent de mieux les comprendre.

1.1 Historique de la diffusion

L'IRM est une technique inventée par PC Lauterbur [37] et qui permet de se servir de la résonance magnétique nucléaire, afin de créer des images en deux dimensions (2D). Au début des années 1900, Albert Einstein a énoncé la théorie du mouvement brownien [20], qui décrit le mouvement aléatoire des particules dans un fluide. En combinaison avec l'IRM, cette théorie a permis à plusieurs chercheurs [38] [42] [52] de réussir à générer les premières cartes de diffusion d'objets synthétiques (fantômes) et de tissus biologiques. En 1986, Le Bihan a réussi à obtenir les premières images provenant de la diffusion dans un cerveau humain [39]. Au départ, l'IRM de diffusion était avant tout une imagerie scalaire, ce qui permettait de voir rapidement et efficacement les accidents vasculo-cérébraux (AVC) (voir la figure 1.1 tirée de Moseley et al. 1990 [44]). Cependant, puisque cette imagerie est scalaire, on ne mesurait qu'une

1.1. HISTORIQUE DE LA DIFFUSION

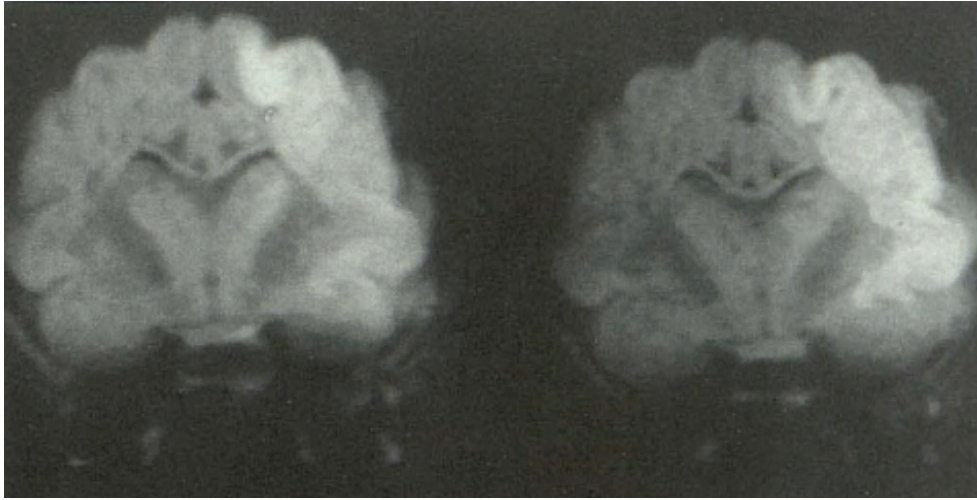


figure 1.1 – Image pondérée de la diffusion d’un patient ayant subi un AVC, 1 heure (gauche) et 3 heures (droite) après l’accident (adapté de Moseley 1990 [44]).

seule direction (ou plutôt la moyenne de x,y,z). Donc, si le cerveau n’était pas aligné en X,Y,Z , les images obtenues changeaient. Il fallait donc trouver une imagerie de diffusion et une reconstruction invariante à la rotation. C’est alors que Basser a proposé le modèle du tenseur de diffusion (DTI) [3], nécessitant un minimum de 6 mesures de diffusion (figure 1.2).

1.1.1 Tenseurs

Un tenseur est un outil mathématique permettant de représenter la diffusion sous la forme d’un système d’axes orthogonaux invariant à la rotation. En 1992, Basser et Le Bihan ont proposé une méthode afin d’estimer les composantes de la matrice D , représentant le tenseur de diffusion [5]. Il est important de noter que 6 mesures sont nécessaires pour estimer la matrice D , composée de 6 composantes indépendantes (car D est symétrique) ainsi qu’une image de référence où $b = 0$.

1.1. HISTORIQUE DE LA DIFFUSION

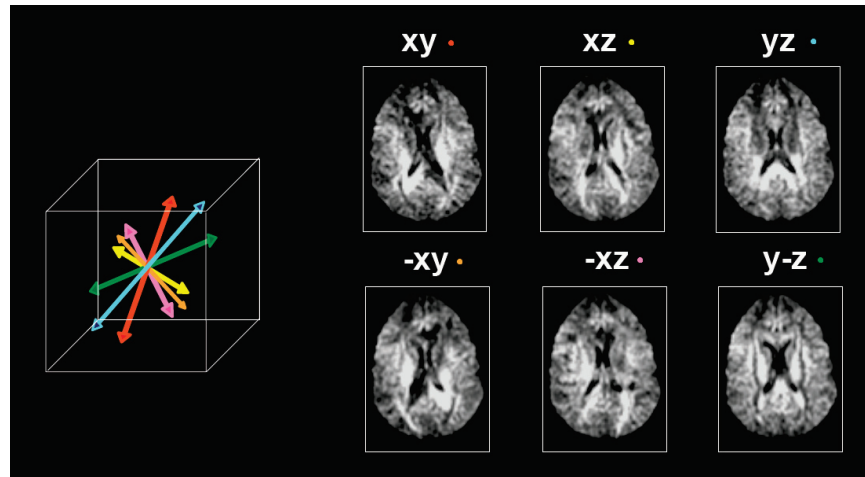


figure 1.2 – 6 images représentant des directions différentes d’une même tranche axiale du cerveau (adapté d’une présentation de Descoteaux 2014 [15]).

$$\mathbf{D} = \begin{bmatrix} d_{11} & d_{21} & d_{31} \\ d_{21} & d_{22} & d_{32} \\ d_{31} & d_{32} & d_{33} \end{bmatrix}.$$

Les 6 coefficients de cette matrice sont estimés à l’aide de cette équation :

$$\mathbf{S}(\mathbf{g}, t) = \mathbf{S}_0 \exp(-b\mathbf{g}^T \mathbf{D} \mathbf{g}),$$

où \mathbf{g} est le vecteur du gradient de diffusion qui est appliqué, b est le terme de pondération regroupant le temps effectif de diffusion et la puissance des gradients, et S_0 correspond à l’image de base ($b=0$). À l’aide de la matrice \mathbf{D} , on peut ensuite construire un ellipsoïde représentant la diffusion à l’intérieur d’un voxel [2], l’équivalent d’un pixel pour une volume 3D.

On peut ensuite faire de l’imagerie du tenseur de diffusion [3]. Pour chaque matrice \mathbf{D} que l’on estime, on peut construire un système local de coordonnées orthogonales. Puisque \mathbf{D} est symétrique et définie positive, ses trois vecteurs propres sont orthogonaux. De plus, chacun de ces vecteurs a une valeur propre associée (figure 1.3). On

1.1. HISTORIQUE DE LA DIFFUSION

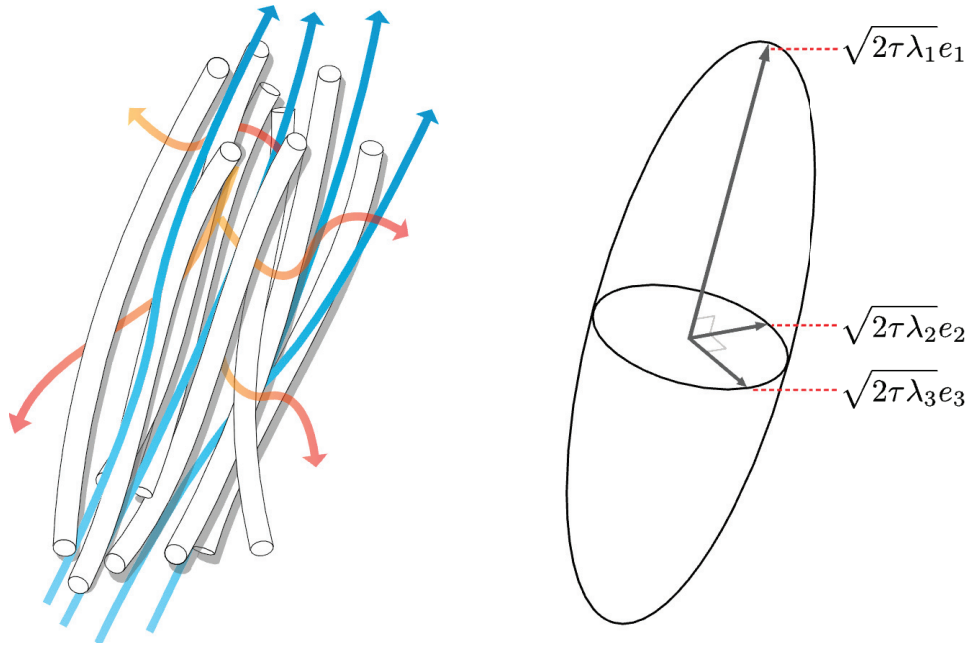


figure 1.3 – Représentation d’un tenseur. À gauche, on peut voir le mouvement brownien de l’eau (adapté de Poupon 1999 [46]). À droite, on peut observer le tenseur de diffusion et ses valeurs propres (adapté de Descoteaux 2008 [14]).

définit le tout avec l’équation de matrice suivante :

$$DE = E\Lambda \text{ avec } E = (\epsilon_1|\epsilon_2|\epsilon_3) \text{ et } \Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

Dans un tissu fibreux anisotrope, les directions des vecteurs propres correspondent aux directions orthogonales de ce tissu. La plus grande valeur propre représente la diffusivité du tissu, et le vecteur propre qui lui est associé représente la direction principale du faisceau de fibres sous-jacent. En répétant ce processus, on obtient l’information nécessaire pour définir un ellipsoïde représentant le tenseur pour chaque voxel. On peut également déduire de l’information par rapport à l’anisotropie du voxel grâce aux vecteurs propres puisqu’ils représentent la diffusion. Par exemple,

1.1. HISTORIQUE DE LA DIFFUSION

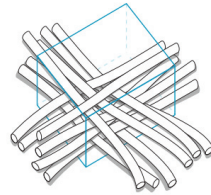
un tenseur en forme de sphère implique que la diffusion se fait également dans toutes les directions, alors qu'un tenseur de forme ellipsoïdale indique une forte diffusion selon un axe. En plus de l'information sur l'orientation des fibres, on peut également déduire des scalaires qui sont invariants à la rotation à partir de D et de ses valeurs propres :

$$I_1 = \lambda_1 + \lambda_2 + \lambda_3 = \text{Tr}(D) = \text{Tr}(\Lambda)$$

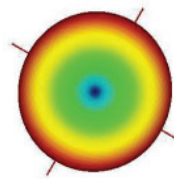
$$I_2 = \lambda_1\lambda_2 + \lambda_3\lambda_1 + \lambda_2\lambda_3$$

$$I_3 = \lambda_1\lambda_2\lambda_3 = |D| = |\Lambda|.$$

I_n représente des propriétés qui nous permettent de caractériser leur milieu, et dans notre cas, un tissu du cerveau.



(a) Déplacement de l'eau (b) Directions principales



(c) Tenseur de diffusion

figure 1.4 – Représentation d'un croisement de fibres dans la matière blanche par le tenseur de diffusion, adapté de Descoteaux et Poupon 2014 [18].

Cependant, puisque le modèle du tenseur suppose que la diffusion suit une distribution gaussienne, elle limite l'information qu'on peut obtenir, surtout lorsqu'il y

1.1. HISTORIQUE DE LA DIFFUSION

a des croisements de fibres. Par exemple, si deux faisceaux se croisent perpendiculairement (figure 1.4), le tenseur représentant le tout serait en forme de disque. Il est donc impossible de savoir dans quelle direction ces deux faisceaux vont à l'aide du tenseur. Ainsi, il a fallu développer des techniques plus avancées afin de résoudre ce problème [14].

1.1.2 Fonction d'Orientation de Distributions (ODF)

En 2000, Wedeen [64] a introduit une méthode afin de pouvoir visualiser la distribution des orientations des fibres à l'intérieur d'un voxel [64]. Cela a mené à l'introduction des méthodes HARDI par Tuch [56] et Descoteaux [16]. Celle présentée par Tuch requiert une méthode d'acquisition utilisant de très grandes valeurs b du gradient de diffusion. Contrairement au DTI, on suppose que le modèle du signal de diffusion d'un voxel est représenté par une mixture de gaussiennes.

L'ODF capture le processus de diffusion de l'eau, qui est intrinsèquement lisse et flou. L'objet d'intérêt pour la tractographie est un modèle des fibres sous-jacentes et non de l'eau qui diffuse dans ce milieu. C'est pour cette raison que la fiber ODF (fODF) fut introduite par Tournier [55] et Descoteaux [17], utilisant la technique de la déconvolution sphérique.

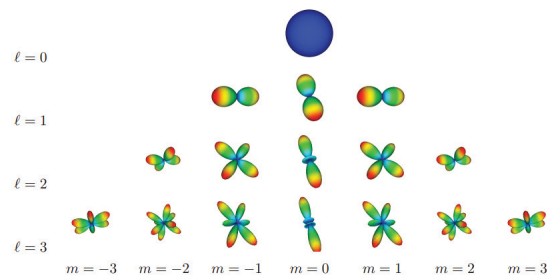


figure 1.5 – Harmoniques sphériques (seulement la partie réelle) jusqu'à leur 3ème ordre (adapté de Descoteaux 2008 [14]). L'harmonique sphérique d'ordre zéro est bleue car elle est constante sur la sphère. Le reste des couleurs illustre les valeurs maximales (rouge) et minimales (bleu).

1.1. HISTORIQUE DE LA DIFFUSION

Healy et al. se sont attaqués au problème de la déconvolution sphérique en 1998 [28]. On construit une base orthonormale de plusieurs fonctions sur la sphère (harmoniques sphériques), ce qui est très semblable aux séries de Fourier dans un espace cartésien (figure 1.5). Comme les harmoniques sphériques (SH) forment une base complète, elles peuvent être utilisées afin de représenter n'importe quelle fonction bornée par valeurs simples sur une sphère. La convolution sphérique (figure 1.6) peut être représentée comme :

$$\underline{S''} = R'' \underline{F''},$$

où F'' est un vecteur de longueur $(2n + 1)$ représentant le nième ordre de décomposition en SH de $F(\Theta, \phi)$, R'' est une matrice $(2n + 1)(2n + 1)$ représentant le nième ordre de décomposition en harmoniques rotationnelles de $R(\Theta)$, et S'' est la représentation du nième ordre en SH de $S(\Theta, \phi)$. La convolution sphérique peut donc être réduite à des multiplications de matrices puisqu'elle est constituée d'un vecteur et d'une matrice. Dans le contexte de la diffusion, cette opération est encore plus simple [55]. La déconvolution sphérique consiste à inverser la matrice R'' pour retrouver F'' .

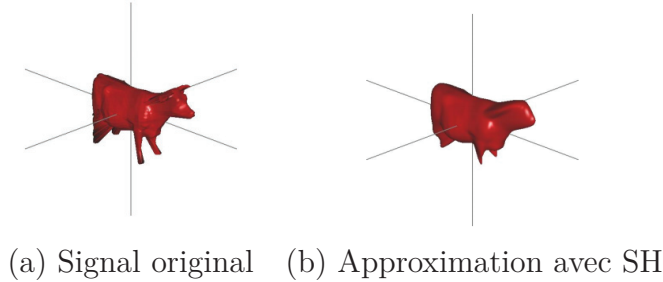


figure 1.6 – Représentation d'un signal de base par une approximation faite à l'aide de la convolution sphérique, adapté de Kazhdan 2015 [35].

À partir de données à haute résolution angulaire, il est possible d'estimer directement la distribution des orientations des fibres d'un voxel sans avoir de connaissances a priori sur le nombre de populations de fibres présentes [55]. Cette méthode repose

1.1. HISTORIQUE DE LA DIFFUSION

sur l'équation :

$$S(\Theta, \phi) = F(\Theta, \phi) * R(\Theta),$$

où S représente le signal, R la fonction de réponse des populations de fibres, et F la fonction de densité d'orientations de fibres (fODF) (figure 1.7). À priori, si l'on connaît la fonction de réponse R , il est possible d'obtenir la fODF en effectuant une déconvolution sphérique du signal S par R .

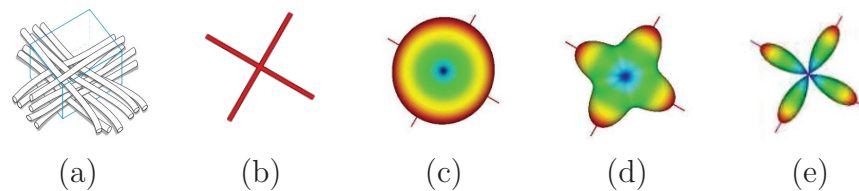


figure 1.7 – Représentation d'un croisement de fibres dans la matière blanche, adapté de Descoteaux et Poupon 2012 [18]. (a) Déplacement de l'eau (b) Directions principales (c) Tenseur de diffusion (d) ODF (e) fODF.

1.1.3 Tractographie

La tractographie est une technique permettant de recréer les fibres de la matière blanche du cerveau. Il existe deux grandes familles d'algorithmes de tractographie : ceux qui sont locaux et ceux qui sont globaux. Dans notre cas, ce sont les algorithmes locaux qui nous intéressent. À partir d'un champ de vecteurs que l'on intègre, on peut reconstruire des chemins (ou plutôt des "streamlines") [43] [12]. Le champ de vecteurs nécessaire pour que ces algorithmes fonctionnent peuvent provenir de différentes sources, comme le vecteur propre principal en DTI [4], ou encore des multiples directions de la fODF en HARDI [17]. On peut voir une illustration d'un faisceau de fibres recréé à l'aide de la tractographie à la figure 1.8.

Chaque courbe que l'on estime à l'aide d'un algorithme de tractographie représente une fibre, plus communément appelée "streamline". Les premiers algorithmes ayant été développés font partie de la famille des méthodes déterministes. Cependant, ces

1.1. HISTORIQUE DE LA DIFFUSION

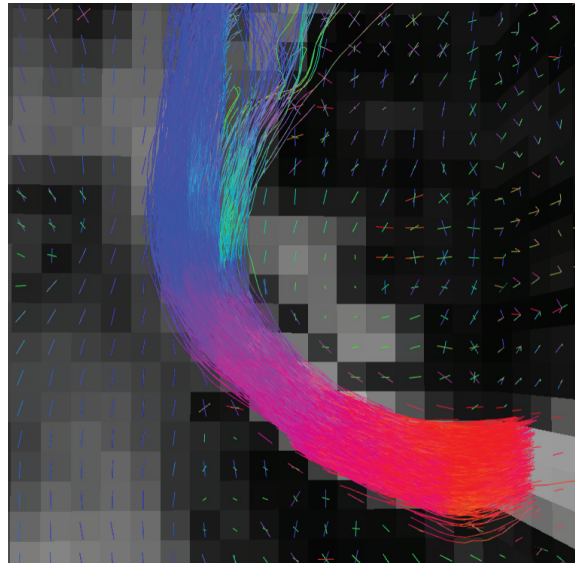


figure 1.8 – Illustration d’un faisceau de fibres issu de la tractographie.

algorithmes déterministes sont affectés par plusieurs limitations, par exemple, une sensibilité à l’estimation de la direction principale de la diffusion [17].

Afin de tenter de résoudre ces problèmes, plusieurs algorithmes probabilistes ont été développés. Par exemple, Berhens et al. [6] ont créé une méthode permettant d’estimer localement des fonctions de densité probabiliste pour des paramètres d’intérêt d’un modèle de diffusion. Ils appliquent cette technique aux tenseurs de diffusion. Une fois l’estimation faite, ils retrouvent une estimation globale de la connectivité du cerveau. De façon plus générale, les algorithmes probabilistes considèrent une incertitude dans les maxima à suivre, ou une distribution probabiliste des fibres lors de la tractographie. Nous nous intéressons particulièrement aux algorithmes utilisant les fODFs puisque celles-ci peuvent être considérées comme étant des fonctions de densité de probabilité de fibres [17]. On peut voir à la figure 1.9 la différence de concept entre la tractographie déterministe et probabiliste. Dans l’exemple probabiliste, on voit très bien que l’incertitude permet d’emprunter un chemin différent de la direction principale.

Généralement, les algorithmes probabilistes sont plus exigeants au niveau des cal-

1.1. HISTORIQUE DE LA DIFFUSION

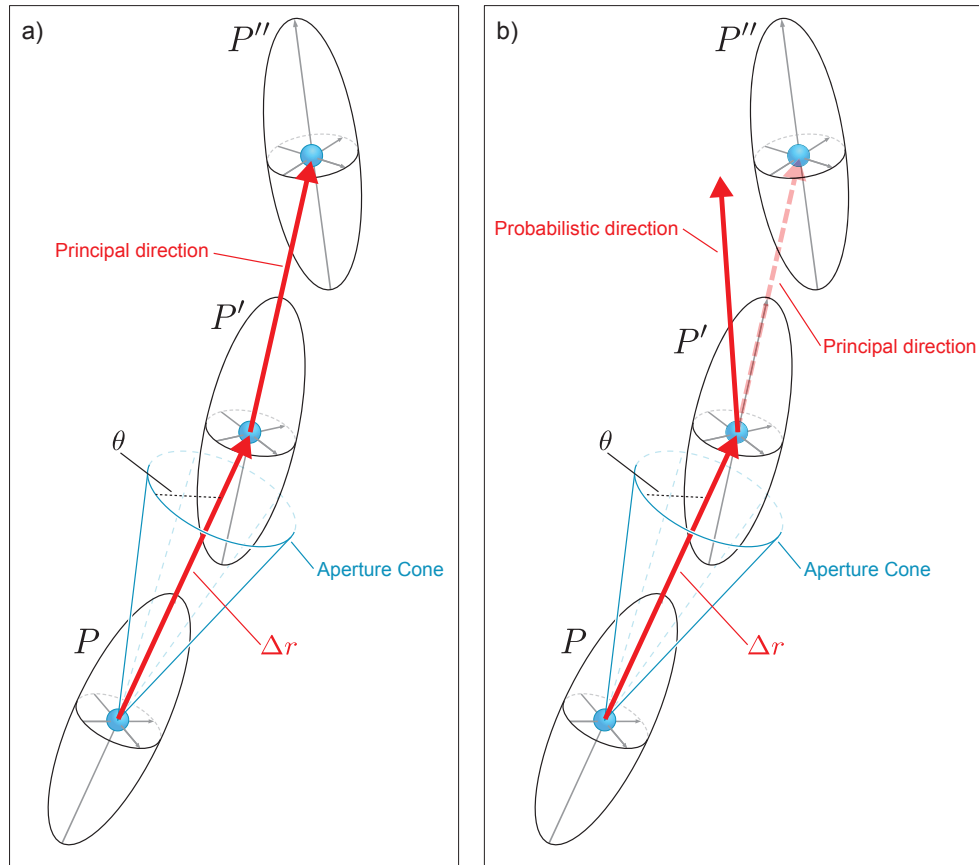


figure 1.9 – Illustration de la sélection de la direction principale à partir d'un champ de tenseurs de diffusion. À gauche, la tractographie déterministe et à droite, la tractographie probabiliste (adapté de Descoteaux et Poupon 2014 [18]).

culs, mais ils sont plus robustes par rapport aux effets de volumes partiels et au bruit dû à l'incertitude de la direction des fibres. De plus, le résultat des algorithmes probabilistes permet généralement d'obtenir une mesure de probabilité de connectivité inter-voxel.

Jusqu'à maintenant, toutes les techniques qui ont été mentionnées sont dites hors-ligne (ou "offline"), c'est-à-dire que l'on exécute les algorithmes, pour ensuite les visualiser. Cependant, en 2014, Chamberland et al. [11] proposent une nouvelle technique permettant de faire la tractographie en temps réel (RTT). L'algorithme nécessite une

1.1. HISTORIQUE DE LA DIFFUSION

carte de maximas extraite (champ de vecteurs) d'une carte de fODFs. Ces maximas sont ensuite utilisés lors de la tractographie en temps réel pour prendre des décisions quant à la direction à prendre lors du traçage, suivant cette équation :

$$\mathbf{V}_{n+1} = (f\mathbf{V}_n) + (1 - f)((1 - g)\mathbf{V}_{n-1} + g\mathbf{V}_n),$$

où les différents \mathbf{V} représentent des directions scalaires de diffusion (x,y,z), f est la valeur de la FA (cela peut être n'importe quelle autre valeur scalaire tiré d'une carte ou d'un masque) dans le voxel courant et g est la valeur du poids que l'on donne aux deux directions impliquées dans l'équation.

Plusieurs paramètres sont impliqués dans ce type d'algorithmes. Ces paramètres sont :

- **la zone d'initialisation** : C'est la zone à partir de laquelle on lance la tractographie. Cette zone peut être un masque de la matière blanche si on veut créer un tractogramme complet, ou encore une région d'intérêt de quelques voxels, si une zone ou un faisceau nous intéresse en particulier.
- **le pas de propagation** : C'est la distance par laquelle l'algorithme évolue entre chaque point d'une "streamline". Ce pas varie généralement entre $0.2mm$ et $1mm$ [14] [53] [12] en fonction de la résolution spatiale de l'image. On peut en voir un exemple sur la figure 1.9 en tant que δr .
- **le masque de tractographie** : L'algorithme de tractographie doit s'effectuer à l'intérieur de la matière blanche, puisque c'est dans ces tissus que les directions extraites des données de la diffusion sont les plus cohérentes. On utilise souvent une carte de FA, puisque la matière blanche a des valeurs plus élevées que les autres tissus du cerveau dans celle-ci. Cette carte est généralement seuillée afin de garder principalement la matière blanche.
- **l'angle d'ouverture** : C'est le maximum que la valeur de l'angle entre deux directions consécutives peut prendre lors du traçage d'une "streamline". Si cet angle est trop grand, on arrête l'exécution de l'algorithme. On peut en voir un exemple sur la figure 1.9 en tant que θ , ainsi que le cône d'ouverture maximale associé.
- **le seuil de longueur minimum et maximum** : Si une "streamline" n'atteint

1.2. VISUALISATION DE LA DIFFUSION

pas le seuil minimum de longueur, elle est rejetée, étant généralement trop courte pour être d'un quelconque intérêt. On pourrait penser à une "streamline" s'étant arrêtée prématurément à cause d'un angle d'ouverture trop élevé. On peut également rejeter celles qui dépassent le seuil maximum de longueur, car l'on suppose que celles-ci sont trop longues pour représenter un faisceau réel.

L'avantage du RTT est de pouvoir voir les effets qu'ont les paramètres de l'algorithme en temps réel. Cela permet d'obtenir des paramètres qui sont optimaux pour pouvoir lancer un algorithme hors-ligne afin d'avoir un tractogramme d'un cerveau complet. Cela permet également d'explorer des cerveaux anormaux ou pathologiques en changeant les paramètres par défaut en temps réel. Par exemple, en neurochirurgie, lorsqu'un patient a une tumeur, on veut pouvoir visualiser l'effet que celle-ci a sur les fibres avoisinantes, sans avoir à créer un tractogramme complet.

1.2 Visualisation de la diffusion

Afin de mieux comprendre les données d'imagerie médicale, il est important de pouvoir les visualiser. Au fil du temps, plusieurs techniques ont été développées afin de pouvoir observer ces différents types de données.

1.2.1 Visualisation d'images

La première technique de visualisation est de représenter les données sous forme de volumes à trois dimensions. On utilise cette technique pour montrer, entre autres, des cartes d'anisotropie fractionnelle (FA), ou encore une carte de couleurs (RGB) permettant de voir l'orientation de la diffusion, comme à la figure 1.10. De plus, on peut généralement visualiser le tout directement en trois dimensions, mais on peut également les visualiser selon les trois différents axes 2D, que l'on peut voir sur la figure 1.11 : axial, coronal et sagittal.

1.2.2 Visualisation de glyphes

Une autre technique de visualisation extrêmement fréquente dans le domaine de l'imagerie médicale est la visualisation de glyphes. Une glyphe est un petit objet

1.2. VISUALISATION DE LA DIFFUSION

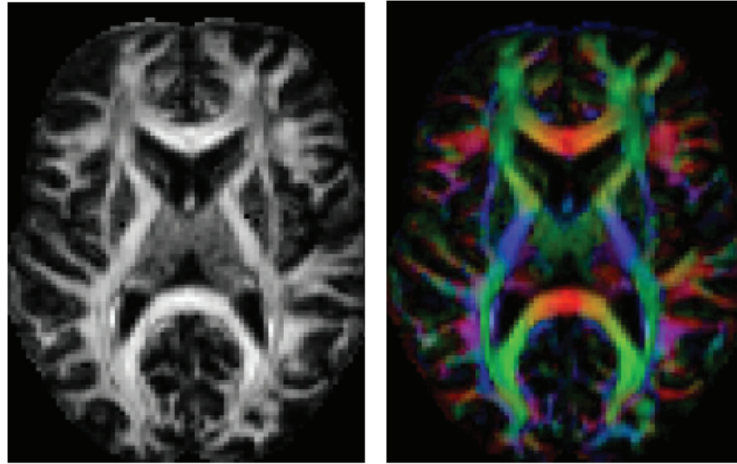


figure 1.10 – Vue d’une coupe axiale de la FA (gauche) et d’une carte RGB (droite)(adapté de Descoteaux 2008 [14]).

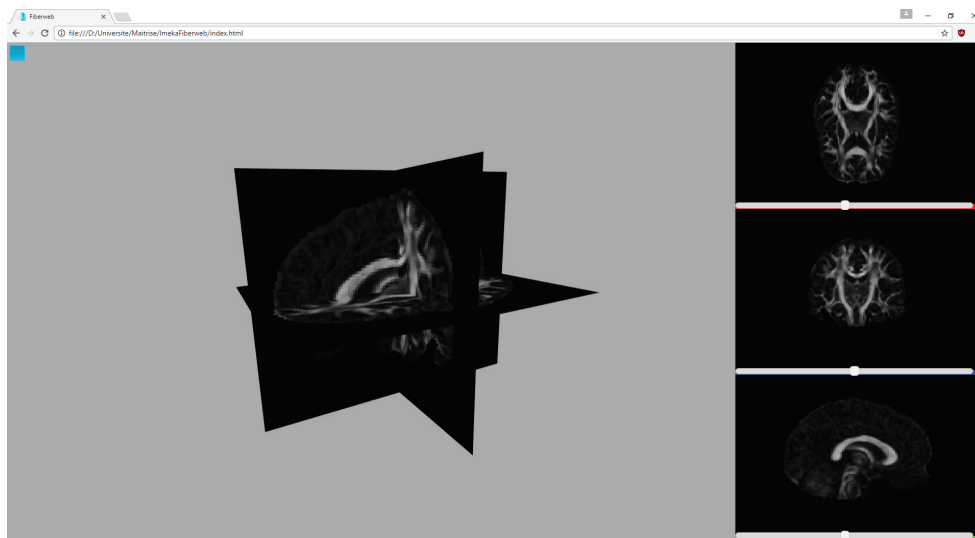


figure 1.11 – Anatomie représentée à la fois en 3D et en 2D selon les trois axes (axial, coronal et sagittal).

1.2. VISUALISATION DE LA DIFFUSION

visuel indépendant représentant plusieurs attributs d'un type de données. C'est la technique qui est utilisée afin de représenter les tenseurs, les ODFs ainsi que les maximas. Chacun de ces types de glyphes sont représentés à la figure 1.12.

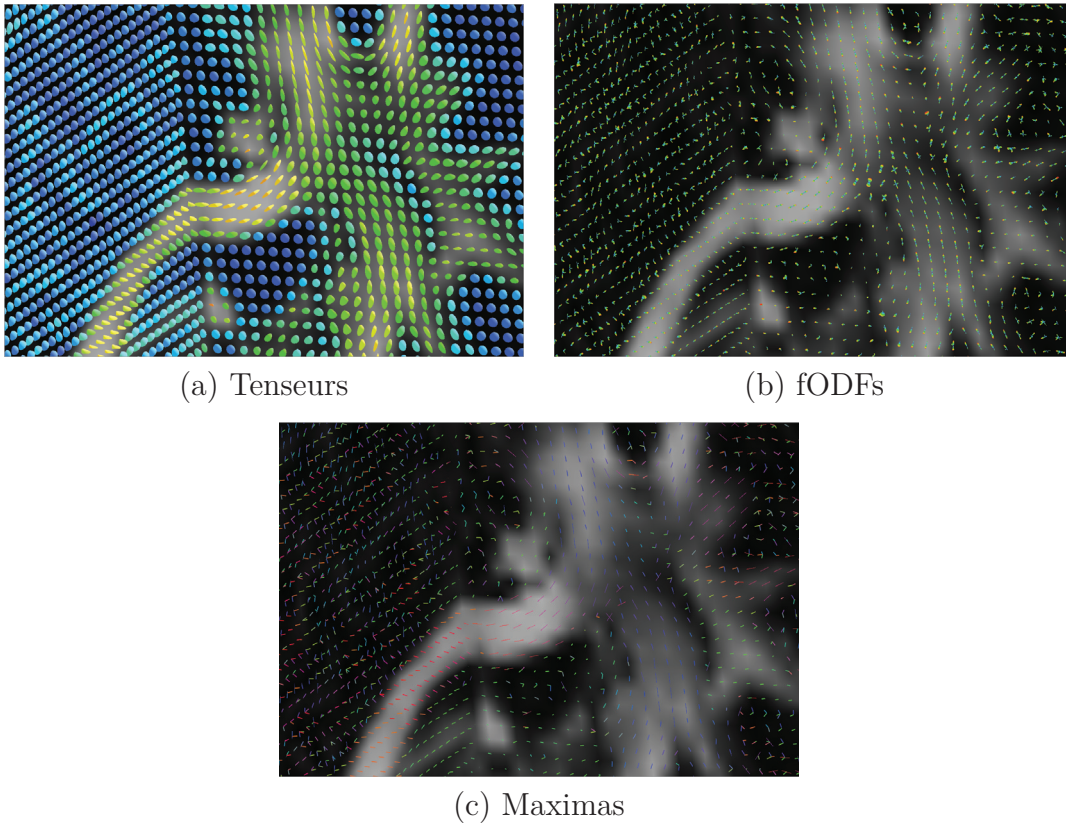


figure 1.12 – Différents types de glyphes.

1.2.3 Visualisation de streamlines

Afin de représenter les fibres du cerveau, on utilise les lignes de courants, ou plutôt les "streamlines". Une "streamline" est une suite de points représentant une courbe (illustrée à la figure 1.13). La couleur de chaque segment de cette courbe représente la direction de la ligne. Par exemple, un segment de droite parallèle à l'axe

1.2. VISUALISATION DE LA DIFFUSION

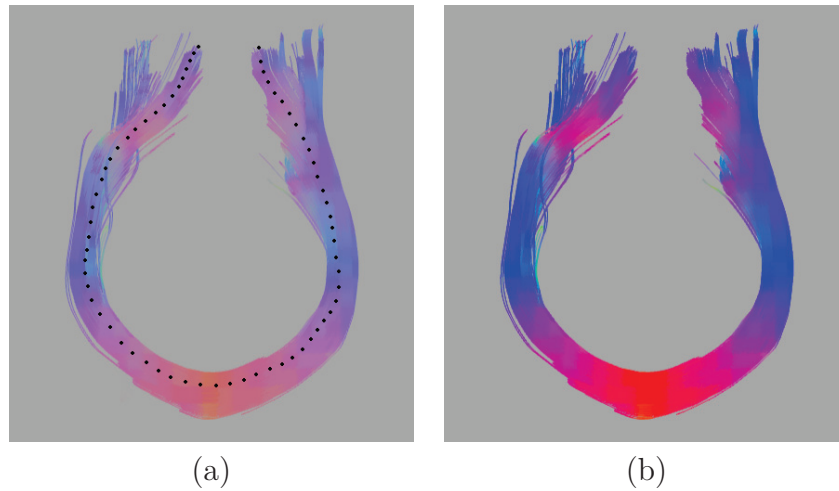


figure 1.13 – Illustration simplifiée d'une "streamline" à l'aide de points (a) et le faisceau résultant, composé de multiples "streamlines".

des x sera rouge, alors qu'un autre étant plutôt parallèle à l'axe des y sera bleu, et le vert est réservé à l'axe des z . Toutes les lignes n'étant pas parallèles à un de ces axes principaux sera un mélange de ces différentes couleurs. Un exemple de plusieurs "streamlines" formant un faisceau est présenté à la figure 1.13.

1.2.4 Visualisation de tractogrammes

Grâce à la tractographie, il est possible de générer des "streamlines" représentant les fibres du cerveau. Lorsqu'on fait de la tractographie partout à l'intérieur du cerveau, on obtient ce qu'on appelle un tractogramme. C'est un ensemble de streamlines qui représente le câblage du cerveau. On peut, par la suite, se servir de ce tractogramme afin de faire de la dissection virtuelle dans le but d'analyser l'anatomie du cerveau d'un patient, ou encore de planifier une chirurgie en fonction des déformations qu'on peut observer autour d'une tumeur. Un tractogramme (66,540 fibres) est illustré à la figure 1.14.

1.2. VISUALISATION DE LA DIFFUSION

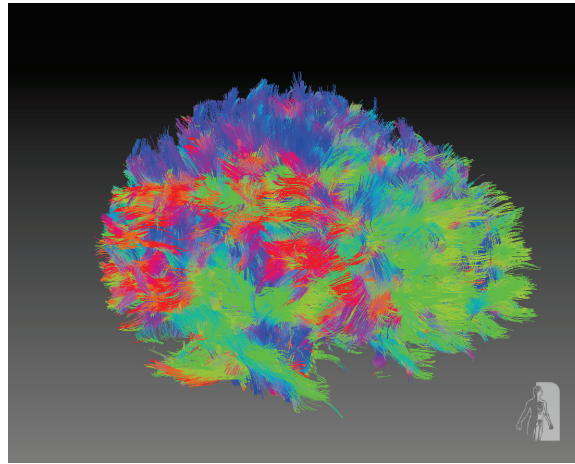


figure 1.14 – Tractogramme d'un cerveau complet (66,540 fibres).

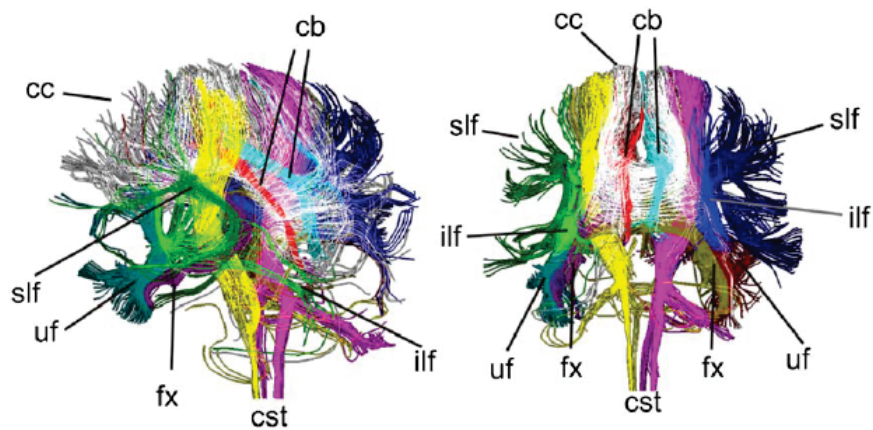


figure 1.15 – Exemple d'un ensemble de faisceaux du cerveau (adapté de Klein 2009).

1.2.5 Visualisation de faisceaux

Lorsqu'on effectue de la tractographie dans le but d'obtenir un tractogramme, on peut regrouper plusieurs fibres suivant un chemin commun en une entité qu'on appelle un faisceau. Afin de former ces faisceaux, on utilise une technique de regroupement,

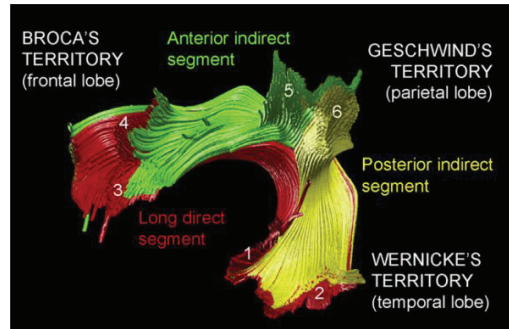
1.2. VISUALISATION DE LA DIFFUSION

par exemple QuickBundles [24], afin de rassembler les fibres. Une fois le regroupement effectué, on peut ensuite procéder à l’affichage des faisceaux. On peut par la suite les afficher sous forme d’enveloppes convexes, ou plus simplement en colorant de différentes couleurs les fibres appartenant à différents faisceaux, comme sur la figure 1.15 (Klein et al. 2009 [36]). Cela permet d’explorer plus facilement notre jeu de données, et de retrouver les faisceaux qui sont pertinents beaucoup plus rapidement, surtout pour quelqu’un connaissant un peu moins l’anatomie du cerveau. On peut ainsi présenter les faisceaux du langage (Catani et Mesulam 2008 [9], figure 1.16 (a)) ou de la vision (Yamamoto et al. 2007 [65], figure 1.16 (b)).

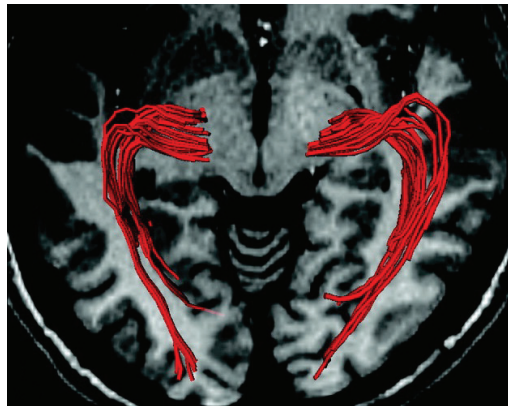
1.2.6 Visualisation de connectomes et graphes

Finalement, il est aussi intéressant de pouvoir représenter visuellement les connexions du cerveau. Un connectome permet de visualiser les connexions neuronales à l’intérieur du cerveau. De plus, à partir d’un connectome, il est possible par la suite, en définissant des zones du cerveau, de représenter ces connexions sous forme d’un graphe. On peut voir un superbe exemple tiré d’un article de Irimia et al. [32] à la figure 1.17.

1.2. VISUALISATION DE LA DIFFUSION



(a) Arcuate fasciculus (adapté de Catani 2008)



(b) Optic radiation (adapté de Yamato 2007)

figure 1.16 – Divers faisceaux du cerveau.

*The programmers of tomorrow are the wizards of the future.
You're going to look like you have magic powers compared to
everybody else.*

GABE NEWELL

Chapitre 2

Visualisation dans le web

Dans ce chapitre, nous verrons l’historique de la technologie principale derrière la création du Fiberweb. Par la suite, certaines décisions du design de notre application web seront expliquées, plus précisément le choix de la boîte à outils qui est utilisé par le Fiberweb. On verra aussi le design général de l’application.

2.1 Visualisation dans le web

2.1.1 HTML5

HTML5 est la technologie principale qui a permis à la visualisation web d’être ce quelle est aujourd’hui. En 2004, la Fondation Mozilla ainsi que la compagnie Opera Software ont créé un groupe de travail appelé le "web Hypertext Application Technology Working Group" (WHATWG) afin de réactiver le développement du format HTML. Ce groupe avait également le devoir de répondre à des besoins complètement nouveaux, tout en se modernisant et en se basant sur des technologies plus récentes et plus simples [60].

En 2007, suite à un sondage auprès d’un comité de plus d’une centaine de développeurs, la décision a été prise de reprendre les travaux du WHATWG comme point de départ afin de former la spécification HTML5 [61].

Sept ans plus tard, en octobre 2014, HTML5 devient la recommandation officielle

2.1. VISUALISATION DANS LE WEB

du World Wide Web Consortium (W3C) et devient donc le standard du web pour le développement de pages et d'applications web [62].

HTML5 amène avec lui plusieurs nouvelles fonctionnalités. Il introduit entre autres de nouveaux éléments et attributs qui sont à jour et plus utiles par rapport aux sites webs modernes. Par exemple, on remplace l'ancien élément `<object>` par `<audio>` et `<video>`. En plus de l'introduction de ces nouvelles méthodes de balisage, HTML5 amène avec lui son lot de nouvelles interfaces de programmation applicative (plus communément appelé API pour le terme anglophone Application Programming Interface). Une de ces APIs sera le point de départ de la visualisation web tel qu'on la connaît aujourd'hui : le canvas. Cette API permet à l'intérieur d'une page web de dessiner divers éléments animés ou interactifs à l'écran, en deux dimensions, lorsqu'un utilisateur visite votre page.

2.1.2 JavaScript

HTML5 ne peut pas utiliser des APIs tel que le canvas par lui-même, c'est pourquoi on doit en plus utiliser le langage de programmation Javascript. Comme le nom le laisse entendre, c'est un langage de programmation scripté. Cela signifie que le code est interprété, et dans notre cas, c'est le navigateur web qui s'en charge. Généralement, Javascript est considéré moins performant qu'un langage compilé comme C++, surtout lorsqu'un programme est exigeant, telle qu'une application de rendu graphique. Cependant, le Javascript a l'avantage d'être supporté facilement sur plusieurs plateformes différentes puisque son code est interprété.

2.1.3 Canvas3D

Durant le développement de HTML5, Vladimir Vukićević, un développeur travaillant pour la Fondation Mozilla présente une nouvelle idée, celle d'utiliser les capacités du canvas afin de pouvoir faire du rendu 3D. C'est à partir de cette prémisse que Vukićević développe un prototype qui permet de créer un contexte 3D d'OpenGL directement dans le canvas. Son prototype est tout d'abord basé sur OpenGL ES 1.1.4 et il se nomme Canvas3D [59]. Un peu plus tard, en 2007, deux extensions de Canvas 3D sont disponibles pour Firefox, l'une toujours basée sur la même version

2.1. VISUALISATION DANS LE WEB

d'OpenGL, et l'autre sur la version 2.0 . Vukićević a choisi OpenGL car c'est une API qui n'a plus besoin de prouver ses mérites inter-plateforme et qui est déjà standardisée (Vukićević 2007 [59]). Par la suite, Canvas3D sera transformé en WebGL.



figure 2.1 – Rendu d'une théière avec Canvas3D (adapté de Vukićević 2007 [59]).

2.1.4 WebGL

WebGL [40] est une technologie permettant de gérer des éléments graphiques directement à l'intérieur d'un fureteur internet. Ceci a été rendu possible grâce à l'arrivée de la technologie HTML5, et plus précisément grâce à une de ses caractéristiques appelée le canvas. L'élément canvas s'avère être une zone rectangulaire dans laquelle on peut dessiner à l'intérieur même d'une page HTML5. Cet élément peut être modifié à l'aide de Javascript. Cette technologie a été développée afin d'être plus portable qu'une librairie graphique traditionnelle, puisqu'un simple fureteur internet en est la seule exigence. WebGL est basé sur la technologie de la librairie graphique OpenGL ES (version 2.0) et le langage de shading d'OpenGL (GLSL, version 1.0). WebGL est supporté par la majorité des fureteurs internet tel que Google Chrome et Mozilla Firefox. Afin de simplifier l'utilisation de WebGL et ajouter un niveau d'abstraction, plusieurs boîtes à outils ont été créées.

2.1. VISUALISATION DANS LE WEB

2.1.5 Revue de différentes boîtes à outils

La première étape dans le processus de création du FiberWeb a été le choix de la boîte à outils qu'on utilise pour interagir avec WebGL. Une boîte à outils est un ensemble d'éléments permettant d'aider à travailler et utiliser, dans notre cas, WebGL. Les trois boîtes qui ont été testées sont : XTK, Scene.js et Three.js. Afin de tester chacune d'elles, trois prototypes permettant d'afficher un cerveau ont été créés. Chacun de ces prototypes étaient comparés principalement par rapport au temps nécessaire pour charger les données de cerveau, ainsi que la vitesse d'affichage lorsqu'on visualise celles-ci. Après avoir créé chaque prototype, les résultats ont montré que la vitesse de chargement était comparable pour chaque librairie, et la vitesse d'affichage également. Puisque les performances étaient très similaires, le choix de la librairie s'est basé sur d'autres critères, telle que la facilité d'utilisation du code.

XTK

XTK [27] est une boîte à outils pour WebGL. Elle a été créée spécifiquement pour pouvoir faire de la visualisation scientifique. C'est également la librairie qui est à la base du logiciel de visualisation Slice:Drop. Ce logiciel permet de visualiser en temps réel des données d'imagerie médicale. XTK permet entre autre d'interagir avec les formats d'imagerie médicale standards tel que Niifti ou TrackVis. Cependant, la librairie ne laisse pas autant de liberté que d'autres étant plus générale. Par exemple, il y a déjà des objets pour les anatomies, ou encore, les fibres. Il y a donc de meilleurs choix si on veut avoir une plus grande liberté pour interagir directement avec nos données.

Scene.js

Scene.js [34] est une autre boîte à outils permettant d'exploiter WebGL. La différence principale est que cette librairie utilise un graphe de scène comme structure générale de données afin de représenter la scène graphique. L'utilisation de cette structure permet notamment de pouvoir appliquer des transformations sur plusieurs objets, puisqu'elle est représentée comme un arbre. Donc, si on applique une transformation sur un parent, tous les enfants seront affectés également. De plus, contrairement à

2.2. DESIGN GÉNÉRAL

XTK, la librairie est un peu plus générale. Plutôt que d'avoir des objets représentant des données d'imagerie médicale, elle donne accès à des géométries.

Three.js

Three.js [7] est encore une fois une boîte à outils permettant d'interagir avec WebGL de façon simple. Tout comme Scene.js, cette librairie donne accès à un bon nombre de géométries simples plutôt qu'à des objets d'imagerie médicale. Elle possède aussi les outils de base pour gérer des caméras, la lumière, ainsi que des géométries plus complexes tel qu'un maillage. Par contre, elle n'utilise pas comme structure de données le graphe de scène. Des trois boîtes à outils qui ont été mentionnées, c'est de loin la plus active avec des contributions fréquentes, alors que les deux autres n'en ont eues aucune depuis 2015.

2.2 Design général

Une fois la boîte à outils choisie, il a fallu choisir le design général de l'application. Comme on peut le voir sur la figure 2.2, chaque aspect du Fiberweb est détaché du reste de l'application. Cela permet entre autre de se retrouver plus facilement dans le code et ainsi définir clairement ce que chaque élément doit faire. De plus, il sera plus facile de faire la transition vers une implémentation "multi-threader" si jamais cela semble être nécessaire.

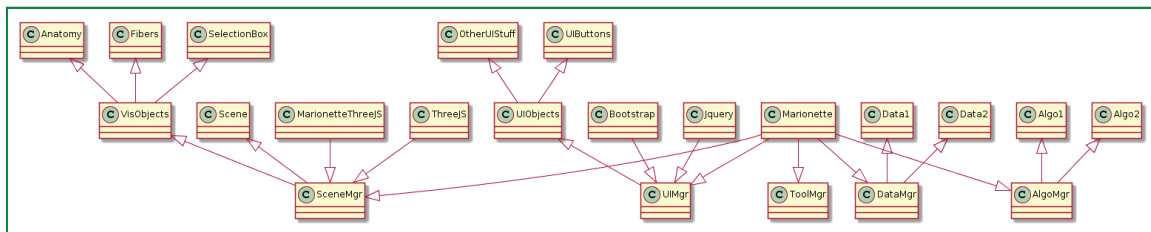


figure 2.2 – Diagramme représentant le design général du Fiberweb.

2.3. LOGICIELS PRÉ-EXISTANTS

2.3 Logiciels pré-existants

Le Fiberweb n'est pas le seul logiciel permettant de faire de la visualisation à l'intérieur du web. Ce sont en partie ces applications déjà existantes qui ont servi d'inspiration pour le développement de notre projet, et qui ont servi de point de départ sur lequel poursuivre la construction de notre application.

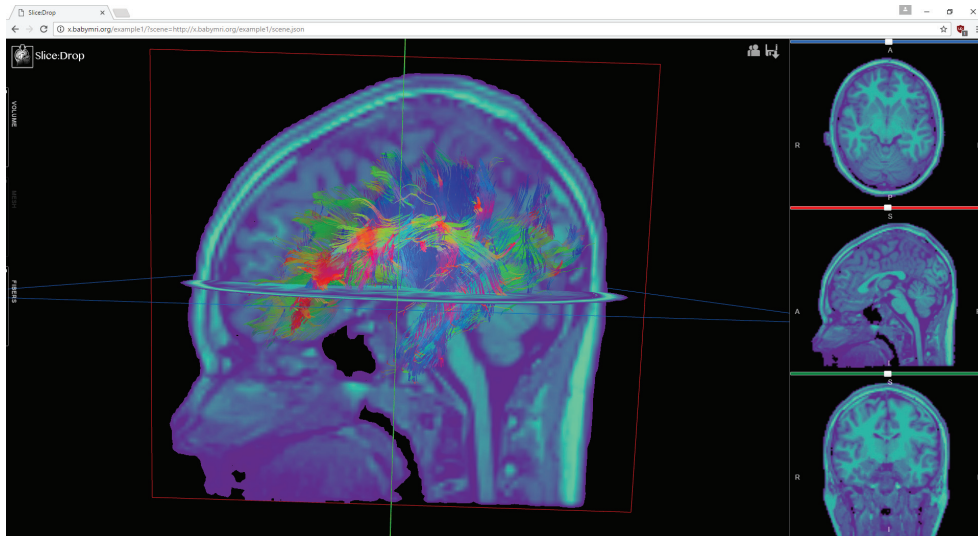


figure 2.3 – Slice:Drop.

2.3.1 Slice:Drop

Slice:Drop [26] est un logiciel qui permet de visualiser différents types de données, comme des fichiers trackvis ou NiFTI. Il permet également de visualiser des volumes, comme des maillages représentant un cerveau, ou encore un vaisseau sanguin. Slice:Drop est un logiciel qui a été fait à partir de XTK. C'est un choix judicieux pour faire de la visualisation, puisque la majorité du code pour gérer les données d'imagerie médicale est fourni. Cependant, Slice:Drop ne permet aucune interaction avec ses données. On peut voir à quoi ressemble Slice:Drop à la figure 2.3.

2.3. LOGICIELS PRÉ-EXISTANTS

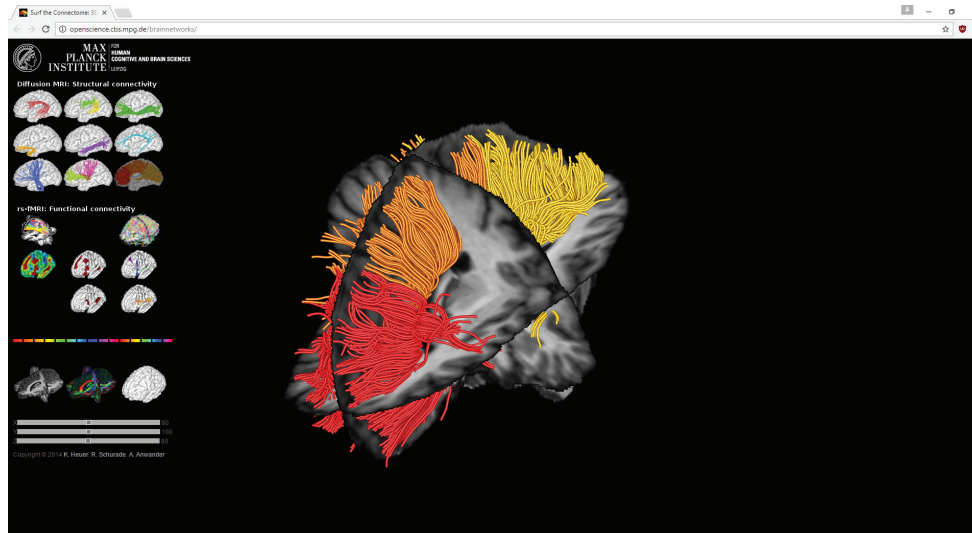


figure 2.4 – Brain Networks.

2.3.2 Brain Networks

Brain Networks est un logiciel développé par un groupe de chercheurs de l'institut Max Planck [29]. Il permet de visualiser des données structurales ainsi que fonctionnelles de connectivité. Cependant, le logiciel ne permet pas d'utiliser ses propres fichiers, seulement des exemples contenus à l'intérieur même de l'application. On peut voir un exemple de visualisation à l'intérieur de Brain Networks à la figure 2.4.

2.3.3 Article interactif

Avant de développer Brain Networks, l'institut Max Planck s'était déjà aventuré dans le domaine de la visualisation web [22]. En 2011, ils ont créé une page représentant un article de Friederici [21]. On peut explorer sur ce site les différents faisceaux les plus importants de cet article. La quantité de données que l'on peut visualiser est assez restreinte. Cependant, c'est l'une des applications web les plus anciennes permettant de faire de la visualisation web d'imagerie médicale en trois dimensions. On peut voir à quoi cette page ressemble sur la figure 2.5.

2.3. LOGICIELS PRÉ-EXISTANTS

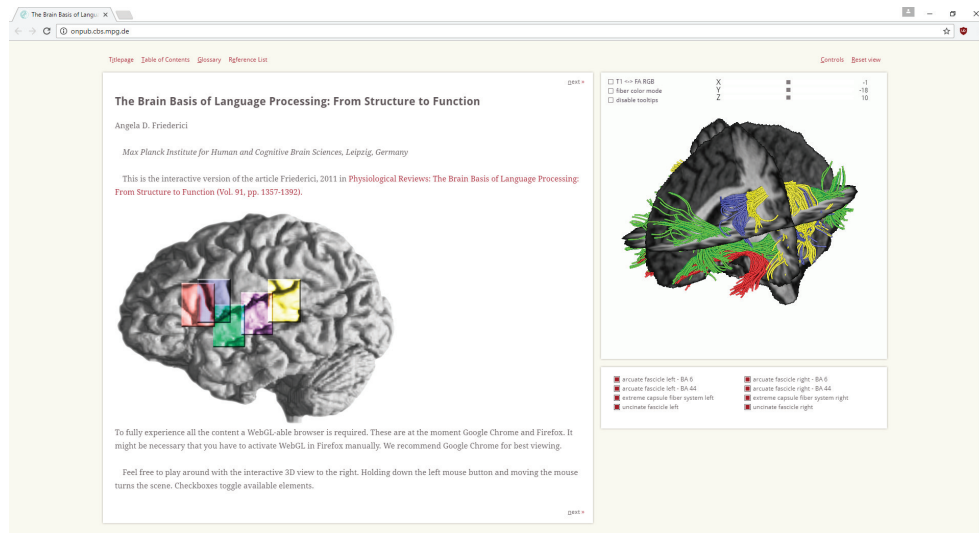


figure 2.5 – Version interactive de l'article de Friederici.

2.3.4 AFQ Browser

Plus récemment, une autre application web d'imagerie médicale est apparue : AFQ Browser [49]. C'est une application permettant de visualiser les résultats d'un algorithme de quantification automatique de fibres (AFQ). Le logiciel permet d'observer en détails les résultats en lien avec les tractes choisies par de l'utilisateur ainsi que les sujets voulus. Cela permet d'interpréter plus facilement les chiffres obtenus par l'AFQ, et d'avoir accès à toutes ses propres données sur une seule page, permettant aussi de tirer des conclusions plus rapidement. On peut voir un exemple à la figure 2.6.

2.3. LOGICIELS PRÉ-EXISTANTS

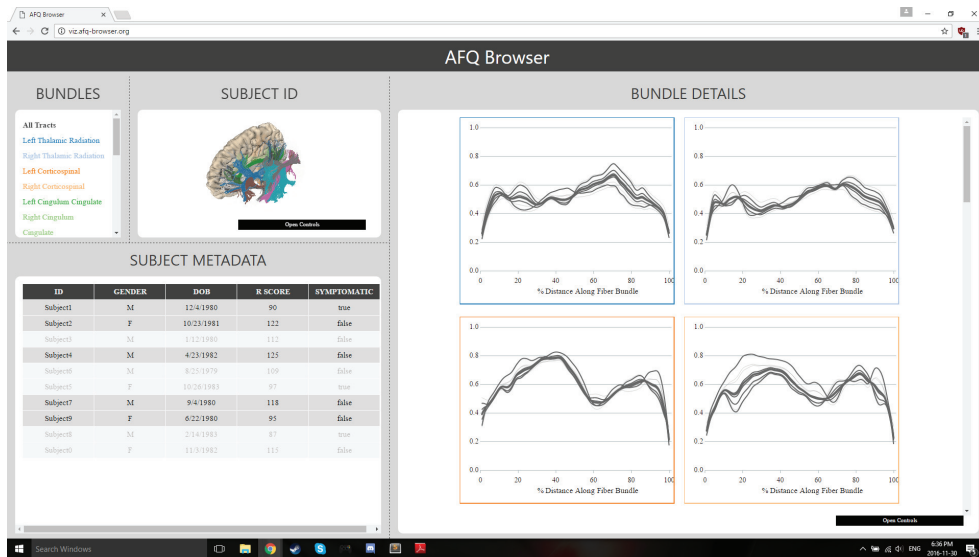


figure 2.6 – AFQ Browser.

*To create a new standard, you have to be up for that challenge
and really enjoy it.*

SHIGERU MIYAMOTO

Chapitre 3

Fiberweb : diffusion visualization and interactivity in the browser

Résumé

L'article présente la nouvelle application Fiberweb. C'est un logiciel permettant non seulement de visualiser des données d'imagerie médicale, mais aussi d'interagir avec elles, le tout directement à l'intérieur d'un navigateur web. La plateforme web est désormais omniprésente dans le monde moderne, et pourtant, elle est encore assez inexplorée dans le domaine de l'IRMd. Les quelques applications déjà existantes ne permettent pas de faire des interactions telle que la sélection de "streamlines", ou encore du RTT déterministe et probabiliste. De plus, cet article introduit cet algorithme de RTT probabiliste. Les résultats qui sont présentés démontrent le potentiel de visualisation de données d'imagerie médicale du web ainsi qu'une comparaison à la fois qualitative et quantitative de notre nouvel algorithme avec une méthode hors-ligne. Cela est effectué en reconstruisant trois faisceaux du cerveau.

Contributions

Les contributions principales de cet article sont :

- L'introduction du Fiberweb, une nouvelle application web permettant à la fois de visualiser des données d'imagerie médicale ainsi que d'interagir avec celles-ci.
- La présentation d'un nouvel algorithme de tractographie probabiliste en temps réel.
- Les résultats de la visualisation de divers types de données d'IRMd.
- Une comparaison qualitative et quantitative entre notre nouvel algorithme de RTT probabiliste et une méthode hors-ligne.

Commentaire

Cet article a été soumis en vue d'être publié dans la revue "Frontiers : Neuroinformatics". Fiberweb a été initialement présenté lors de la conférence *Breaking the Barriers of Diffusion MRI* (ISMRM workshop septembre 2016), voir l'abstract à l'annexe [A](#).

Fiberweb : diffusion visualization and processing in the browser

Louis-Philippe Ledoux^{1,*}, Felix C. Morency², Martin Cousineau^{1,6}, Jean-Christophe Houde², Kevin Whittingstall^{3,4,5} and Maxime Descoteaux^{1,2,3,*}

¹Sherbrooke Connectivity Imaging Lab, Computer Science Department, Faculty of Science, University of Sherbrooke, Sherbrooke, QC, Canada

²Imeka, Sherbrooke, QC, Canada

³Centre de Recherche CHUS, University of Sherbrooke, Sherbrooke, QC, Canada

⁴Department of Nuclear Medicine and Radiobiology, University of Sherbrooke, Sherbrooke, QC, Canada

⁵Department of Diagnostic Radiology, University of Sherbrooke, Sherbrooke, QC, Canada

⁶Videos & Images Theory and Analytics Laboratory, University of Sherbrooke, Sherbrooke, QC, Canada

Abstract

Data visualization is one of the most important tool to explore the brain as we know it. In this work, we introduce a novel browser-based solution for medical imaging data visualization and interaction with diffusion-weighted magnetic resonance imaging (dMRI) and tractography data : Fiberweb. It uses a recent technology, WebGL, that has yet to be fully explored for medical imaging purposes. There are currently very few software tools that allow medical imaging data visualization in the browser, and none of these tools support efficient data interaction and processing, such as streamlines selection and real-time deterministic and probabilistic tractography (RTT). We show results of the visualization of medical imaging data, and demonstrate that our new RTT probabilistic algorithm can compare to a state of the art offline algorithm. Overall, Fiberweb allows the user to interact and process his data in ways that were not previously possible in the browser, which opens great perspectives for quality control and neurosurgical navigation on browser-based mobile and static devices.

Keywords : diffusion MRI, tractography, medical visualization, interaction, web, WebGL

3.1 Introduction

In recent years, the world has seen a huge increase in the use of mobile devices and web technologies. It is becoming more and more important that all resources are always available at anytime, anywhere (e.g. Dropbox, GoogleDrive, iCloud, OneDrive, and many other such providers). Currently, in the field of medical imaging, there are a lot of software tools to view and interact with data. However, only a few exist for the browser, even with all the recent possibilities that the Web Graphics Library (WebGL) enables. Having the possibility to simply load a web page to start interacting with brain imaging data would open possibilities for the whole field.

There are a few existing software tools that are currently available and offer visualization of medical imaging data in the web browser. First, there is Slice :Drop ([26]). It allows its users to visualize anatomies, streamlines and volumes that are on their computer, and the file types it supports cover most of the standard medical imaging formats (e.g. NIFTI, TrackVis .trk, etc.). Then, there are two web applications that were developed by the Max Planck Institute. In 2011, they made an interactive web page covering an article by Friederici ([21], [22]). It allows the user to view the various fiber bundles presented in said article. In 2014, they developed Brain Networks ([29]), which allows exploring structural and functional connectivity data provided by the application. Finally, the AFQ Browser ([49]) was created to visualize the results obtained by Automated Fiber Quantification (AFQ [66]). Specific results can be selected from desired white matter bundles and subjects. All of those tools are strictly for visualization. An interesting evolution would be to allow interaction with the data. For instance, one could want to virtually dissect a tractogram, or rapidly explore tractography results of a tumor patient in real-time.

In this work, we propose *Fiberweb*, a web-based visualization tool with additional processing and interaction capabilities. The contributions of this work are :

- Visualization of 3D volumes ("anatomies") and principal diffusion directions ("peaks").
- Full-brain tractogram (up to 75,000 "streamlines") loading and smooth interaction using a selection object. This enables virtual dissection manipulations directly in Fiberweb.

3.2. METHODS

— Real-time deterministic and probabilistic web-based fiber tractography.

These features allow the user to interact and process his data in ways that were not previously possible in the browser. This means no installation is required for the user, and no server is required for backend computation. This opens great perspectives for quality control and neurosurgical navigation on browser-based mobile and static devices.

3.2 Methods

3.2.1 Visualization in the web

WebGL ([59], [40]) is the driving technology that enabled the creation of Fiberweb. WebGL's creation was made possible by the arrival of HTML5, and more specifically one of its new features, the canvas element. This element allows the creation of a drawable region in the browser, which can then be modified using JavaScript. WebGL is a low-level graphics library for 3D rendering in the browser and is functional with all the major players, such as Chrome, Firefox, Edge and Safari.

Three.js

To simplify the development of Fiberweb, a high-level library called Three.js ([7]) was used. This library wraps WebGL calls and makes it more convenient to use shaders, manage scenes and render primitives (e.g. lines). According to stats.js.org (as of january 2017), Three.js is one of the most used WebGL library (it is used by AFQ Browser), and has a very active community, leading to constant updates. This will make it easier to upgrade Fiberweb over time. It is also lower level than the X toolkit ([27], used by Slice :drop), making it a better choice for this implementation, since basic objects need to be directly manipulable for performance reasons.

Visualization techniques

Fiberweb can currently display three types of medical imaging data : anatomies (peaks) and streamlines. These datasets can be visualized both from a 3D perspective

3.2. METHODS

as well as a 2D one, as seen in Figure 3.1. This is implemented using four different renderers, one for the three dimensional view, and three for two dimensional views, representing the axial, coronal and sagittal views (Figure 3.1). The 3D view can be manipulated through translations, rotations and zooms. For the 2D views, only translations and zooms are available (see video at www.imeka.ca/FiberWeb).

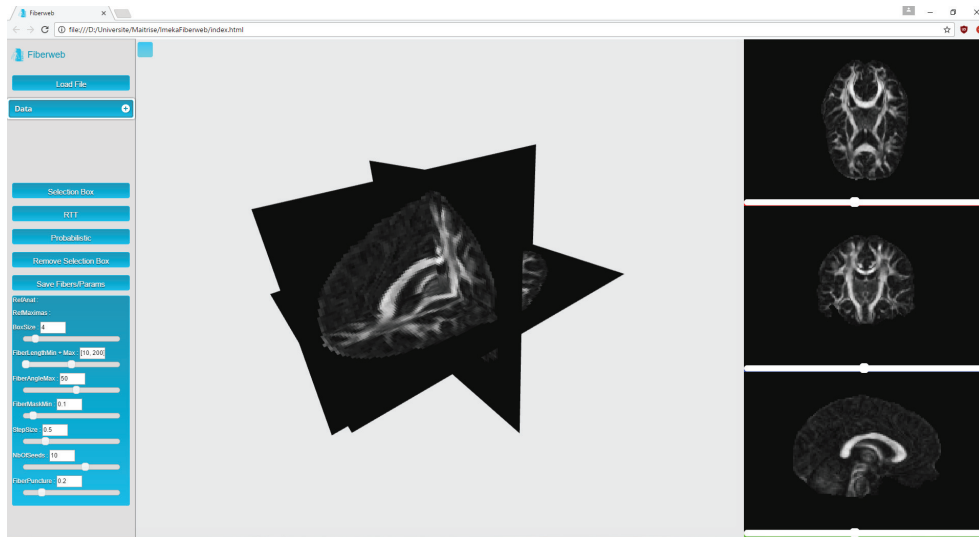


figure 3.1 – Anatomy visualization in 3D and 2D views inside Fiberweb.

"Anatomies" are any 3D volume datasets, such as T1, T2, fractional anisotropy (FA), mean diffusivity (MD), and colored FA (4D image). They are represented by plane buffer geometries. These are simple textured planes, both in 3D and 2D. These textures are computed during the initialization. They can be navigated through with the help of different sliders for each view. When a slice change is made, planes are simply translated, and a simple texture swap is applied since they are already loaded in memory.

Next are the "peaks", which are 4D volumes. The first three represent the coordinates of a peak (X,Y,Z). The fourth one contains the directions extracted from the fiber ODFs (X,Y,Z,N where $N = (x_1, y_1, z_1, \dots, x_5, y_5, z_5)$). Up to 5 directions can be kept per fiber ODF. The peaks can have an additional α parameter to represent the

3.2. METHODS

maxima's uncertainty (described later for probabilistic tracking, see Figure 3.6) if the dataset is probabilistic (X, Y, Z, α) . Peaks are represented by another object provided by Three.js called line segments (Figure 3.2). However, the geometry as well as the colors need to be recomputed for each new rendered slice, since this is computed only for the currently displayed slice (see video at www.imeka.ca/FiberWeb).

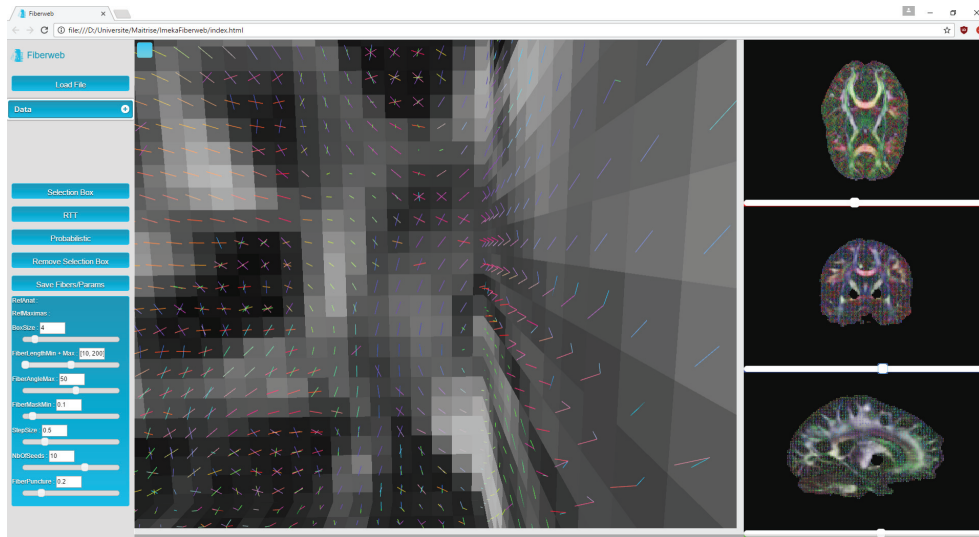


figure 3.2 – Anatomy and peaks visualization in 3D and 2D views inside Fiberweb.

Finally, there are the "streamlines". A streamline is a set of 3D points $(p_i = (x_{p_i}, y_{p_i}, z_{p_i}), \text{where } i \in (1, N) \text{ and } N \text{ is the number of points in the streamline})$. The distance between two consecutive points is called the step size, and is usually regular, though not required. Each pair of points of a streamline is represented by a line segment. During the initialization process, the color for each segment is computed. In the 3D view, line segments are simply drawn. However, in the multiple 2D views available, only the part of the streamlines matching the currently displayed anatomy slice are drawn. This is implemented through a shader which calculates the volume in which lines will be displayed, as seen on Figure 3.3. This also means that whenever the slice is changed, only the parameters sent to the shader are changed, making it extremely efficient and fast. As it will be discussed later, Fiberweb currently works

3.2. METHODS

best with streamlines datasets of up to 75,000 streamlines, which on average have 100 points per streamline. To sum up, there are currently two ways to visualize streamlines : viewing all the streamlines in 3D, or viewing only the intersected streamlines with the planes of the anatomies in the 2D views, as seen in Figure 3.3 (see video at www.imeka.ca/FiberWeb).

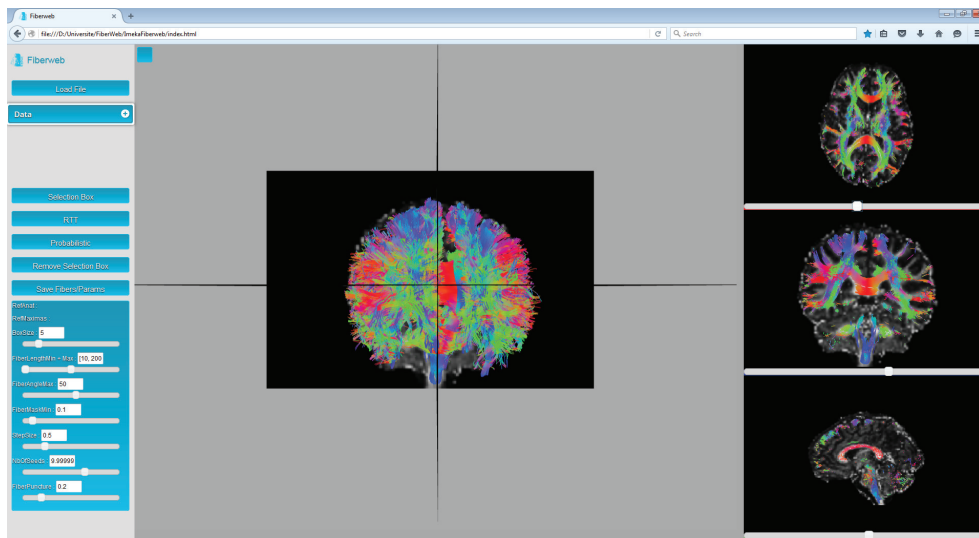


figure 3.3 – Anatomy and streamlines visualization inside Fiberweb, with the intersected streamlines in the 2D views.

3.2.2 Interactions and processing

Fiberweb supports various interaction mechanisms : streamlines selection and real-time tractography.

Selection

Fiberweb provides a streamlines selection mechanism based on a points inclusion test. All streamlines having at least one point inside the selection box will be displayed.

To reduce complexity and increase performance (trying to ensure that feedback is provided in real time), an octree ([41]) is used to classify streamlines points. An

3.2. METHODS

octree is a data structure that recursively subdivides space in eight smaller regions until a certain level of simplification is obtained. Once the space is divided, each point is classified into the octree. This results in a much smaller search space and fewer points to search through.

Even though an octree was already available in Three.js, a new implementation was done in Fiberweb. The main reason was that the octree of Three.js dynamically transforms the octree every time a new element is added to it. Since a full dataset of streamlines contains a high number of points, it was unnecessarily slow to initialize, especially since the position of the points is static and known a priori.

Menu and parameters

From the data drop-down menu (see Figure 3.5), the user can select which anatomy and which map of peaks are selected as a reference for the real-time tractography. The box size can be modified by the user both for the selection and the tractography, as seen on Figure 3.4. The rest of the parameters are specifically for the latter ([11]). Fiber length changes the minimum and maximum span a fiber must have to be valid. The fiber angle is the maximum angle a new direction can have with the previous one. The fiber mask parameter is the minimum value the voxel needs to have to continue tracking. The step size is the distance the tracking travels at every step of the algorithm. The number of seeds determines the quantity of streamlines the results will have. Finally, the fiber puncture is a variable that weighs the incoming direction of the tracking as well as the next one ([11]). The equation that the deterministic RTT algorithm follows is :

$$\mathbf{V}_{n+1} = (f\mathbf{V}_n) + (1 - f)((1 - g)\mathbf{V}_{n-1} + g\mathbf{V}_n), \quad (3.1)$$

where f is the FA (or GFA, AFD, or any other underlying scalar map or mask) at the current voxel, g is the puncture value that was mentioned previously and $\mathbf{V} \in \mathbb{R}^3$ is the direction (x,y,z) which the algorithm follows ($n - 1$ is the previous direction, n is the current direction and $n + 1$ is the next direction to follow).

Additionally, the user can save the computed/extracted streamlines, from both the selection and the real-time tractography. The parameters are also saved, making

3.2. METHODS

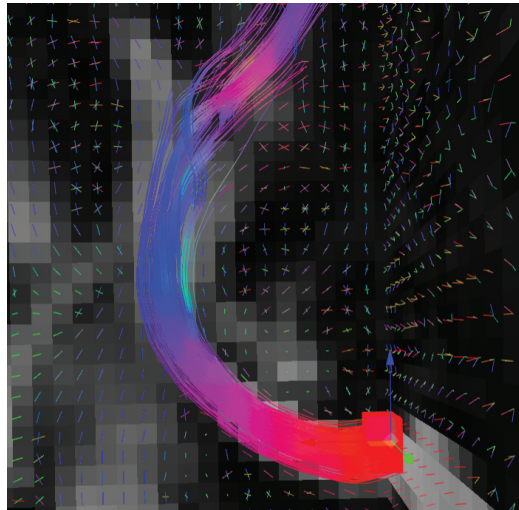


figure 3.4 – Fiber bundle (part of the corpus callosum), result of the real-time tractography, with the initialization box (red) in the mid-sagittal slice of the CC and the following parameters : angle threshold of 50 deg, fiber length minimum of 10mm and maximum of 200mm, FA threshold of 0.1, a step size of 0.5mm, 10 seeds per axis and a puncture value (g) of 0.2.

it easier for a user to recreate its selection or tracking.

Deterministic real-time tractography (RTT)

Chamberland et al.'s [11] deterministic RTT algorithm, which was originally implemented in the Fibernavigator ([58]), was implemented in Fiberweb. In order to use it, the first step is to load a precomputed map of peaks. These peaks are extracted from a map of fiber ODFs ([54], [17]) or any other diffusion model (multi-compartment models, DSI, etc...) ([50], [63], [16], etc.) and correspond to the maximas found in each voxel. Once the peaks are computed offline, the tracking starts by seeding randomly inside a small selection box. This seeding strategy is known as region-of-interest (ROI) seeding, which was chosen over complete seeding, which seeds everywhere in the tracking mask ([4], [8], [14]). For each seed, a random peak is selected from the voxel at its current position. The tracking consists in taking the current position, comparing

3.2. METHODS

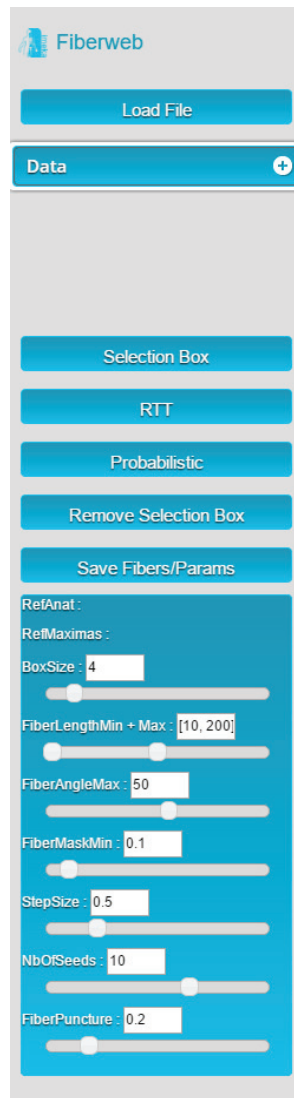


figure 3.5 – Fiberweb menu and parameters.

the current tracking direction with all the peaks of the current voxel, and selecting the one with the smallest angular difference. Then, the current direction as well as the next direction are weighted with a user-defined variable. The algorithm then moves a step in this resulting direction. These steps are repeated until various possible end criteria are met, such as fiber length. As can be seen at www.imeka.ca/FiberWeb,

3.2. METHODS

one can change the tracking parameters in real time (FA threshold, step size, curvature θ angle, puncture). This allows one to explore his peaks in real time for quality assurance or exploration purposes. [11] have shown the power of such an approach for surgical planning and exploring the white matter architecture around the tumor.

Probabilistic RTT

FiberWeb also introduces a new probabilistic real-time tracking technique, building on the already existing deterministic RTT technique. Instead of loading the standard peaks file containing only the directions of the maximas, the user needs to load a peaks file also containing the uncertainty α_i (X,Y,Z,N where $N = (x_1, y_1, z_1, \alpha_1, \dots, x_5, y_5, z_5, \alpha_5)$). The α_i could come from any source, but in this work, a Dipy script (Supplemental data) that computes the full width half max (FWHM) of each fODF peak is provided. As seen in Figure 3.6, one can play with the percentage at which the FWHM is computed to have a more or less conservative α value around each fODF peak. To do this, the fODF is mapped onto a sphere of the user's choice, a 724 vertices symmetric one in our case ([23]). Then, starting from the different maximas of the sphere, the neighbouring vertices are traversed until reaching the desired fraction of the peak. The angle between that vertex and the one from the maxima is then computed and saved along with the peak.

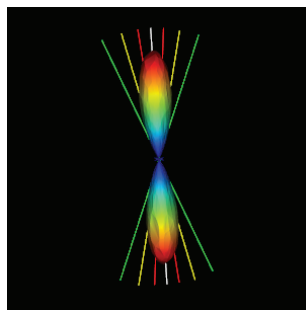


figure 3.6 – Illustration of the uncertainty alpha parameter at 95%, 75% and 35% of the maxima. The script to compute these alpha values for each fODF peak is given in the Appendix.

3.2. METHODS

The first part of the tracking algorithm is the same as for the deterministic version. Seeds are chosen randomly in the selection box. Then, a random peak associated to the voxel at the current position is chosen. Next, instead of simply following that tracking direction, the uncertainty angle associated with that peak is fetched. Then, a random direction in the "cone of uncertainty" formed by the uncertainty angle and the peak is chosen. Then, the algorithm continues executing normally, with the extra step of modifying the direction by an angle of uncertainty every time. Equation 3.1 then becomes :

$$\mathbf{V}_{n+1} = (f\theta(\mathbf{V}_n, \alpha)) + (1 - f)((1 - g)\mathbf{V}_{n-1} + g\theta(\mathbf{V}_n, \alpha)), \quad (3.2)$$

where $\theta()$ represents the function modifying the direction \mathbf{V} by picking a random direction in a cone formed by its uncertainty α .

Dataset

The dataset used for the evaluation of the probabilistic tracking algorithm is found on the SCIL website (http://scil.dinf.usherbrooke.ca/?page_id=822&lang=en) and came originally from the Max Planck Institute ([1], [17]). The data was collected on a 3T Magnetom Trio scanner (Siemens, Erlangen) using a spin-echo echo-planar-imaging sequence (TE/TR = 100 ms / 12 s) with a b-value of 1000 s/mm² and 60 diffusion encoding gradients ([33]). The dimensions are 93 x 116 x 93 isotropic voxel resolution of 1.72 mm for both the T1-anatomy and the fODF map. The data comes from 8 healthy right-hand volunteers (4 males, 4 females), from whom written consent was obtained. fODFs were estimated with Dipy ([23]) using a maximal spherical harmonics order of 8. Peaks were extracted using a tessellated sphere with 724 points.

Evaluation and validation

To determine the validity of our new real-time tracking algorithm, the evaluation was done in two ways, as similarly done in [11]. First, streamlines generated by our algorithm were qualitatively compared to the results of an offline probabilistic tracking algorithm, as implemented in [25] and MRtrix ([53]). The tracking was done on

3.3. RESULTS

the same dataset. Then, a quantitative evaluation is done using the Dice coefficient ([19]), as done in [13]. Using the arcuate fasciculus, corpus callosum and corticospinal tract, the reconstructed bundles from our RTT algorithm vs the offline one are compared using a weighted version of the Dice coefficient to account for the density of streamlines traversing each voxel (see Supplemental data for details). The evaluation of the framerate of the application was done on a laptop with the following system : Windows 8 64-bit, Video card (GPU) : NVIDIA GeForce GTX 675MX (4GB), CPU : Intel(R) Core(TM) i7-3740QM @ 2.70GHz (4 cores), 16 GB RAM.

3.3 Results

3.3.1 Visualization

In this work, it is considered that the baseline performance for the visualization part of the tool is to display an anatomical dataset. Displaying a peaks dataset does not affect the framerate in any noticeable way. In Figure 3.3, the Fiberweb interface is shown simultaneously displaying an anatomical dataset and a set of streamlines. In Figure 3.2, Fiberweb shows an anatomical dataset and its corresponding peaks dataset too. Currently, the 3D view directly allows for picking, interaction and selection. In the 2D renderers (axial/coronal/sagittal), streamlines intersecting with the anatomical dataset are shown (Figure 3.3).

The main bottleneck of Fiberweb is the display of streamlines in general. The framerate and the maximal streamlines capacity are directly affected by the number of streamlines and the number of points per streamline. To avoid any potential bias, all the streamlines datasets that were used are raw and have roughly the same average number points per streamline (100 points). In Figure 3.7, it can be seen that doubling the number of streamlines approximately halves the framerate per second (fps). With 25,000 streamlines, the framerate stops being affected in a noticeable way, but memory issues become problematic. With 80,000 streamlines, major memory issues arise due to browser limitations. This makes it impossible to currently load very large datasets. One could circumvent that problem by using a browser with a much higher memory limit (like the Firefox Nightly version : <https://nightly.mozilla.org/>). Other options

3.3. RESULTS

are to use streamline compression ([47], [48]) and grouping of similar streamlines ([24]) would help the framerate and fluidity of the application in the future since it equates to showing less points per streamline, thus pushing the limit of the current maximum number of streamlines.

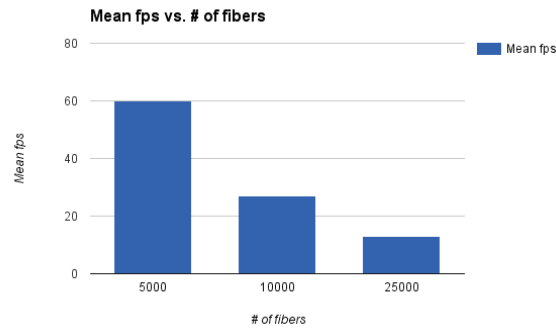


figure 3.7 – Mean framerate (in frames per second (fps)) versus number of streamlines.

3.3.2 Streamline interactions

Every interaction in Fiberweb works in real time. The framerate is not impacted whether or not the intersected streamlines are shown. This is because of the use of a specific shader to show only the portion intersecting the anatomy. This means that rendering the intersected streamlines is just like rendering standard streamlines.

The selection also has little impact on the framerate due to the use of an octree, and this data structure does not impact the loading time of the streamlines. However, when loading bigger streamline datasets, the framerate can drop when in areas of high streamline density during the selection. The framerate is only affected when moving the selection box, going back to normal once it is stopped.

Real-time tractography can have a bigger impact on the framerate of Fiberweb. A noticeable reduction in framerate is obtained when getting close to 1,000 seeds (10 seeds per axis). This drop is more present when tracking longer bundles. In general, the number of seeds is the limiting factor. The rest of the parameters do not negatively

3.3. RESULTS

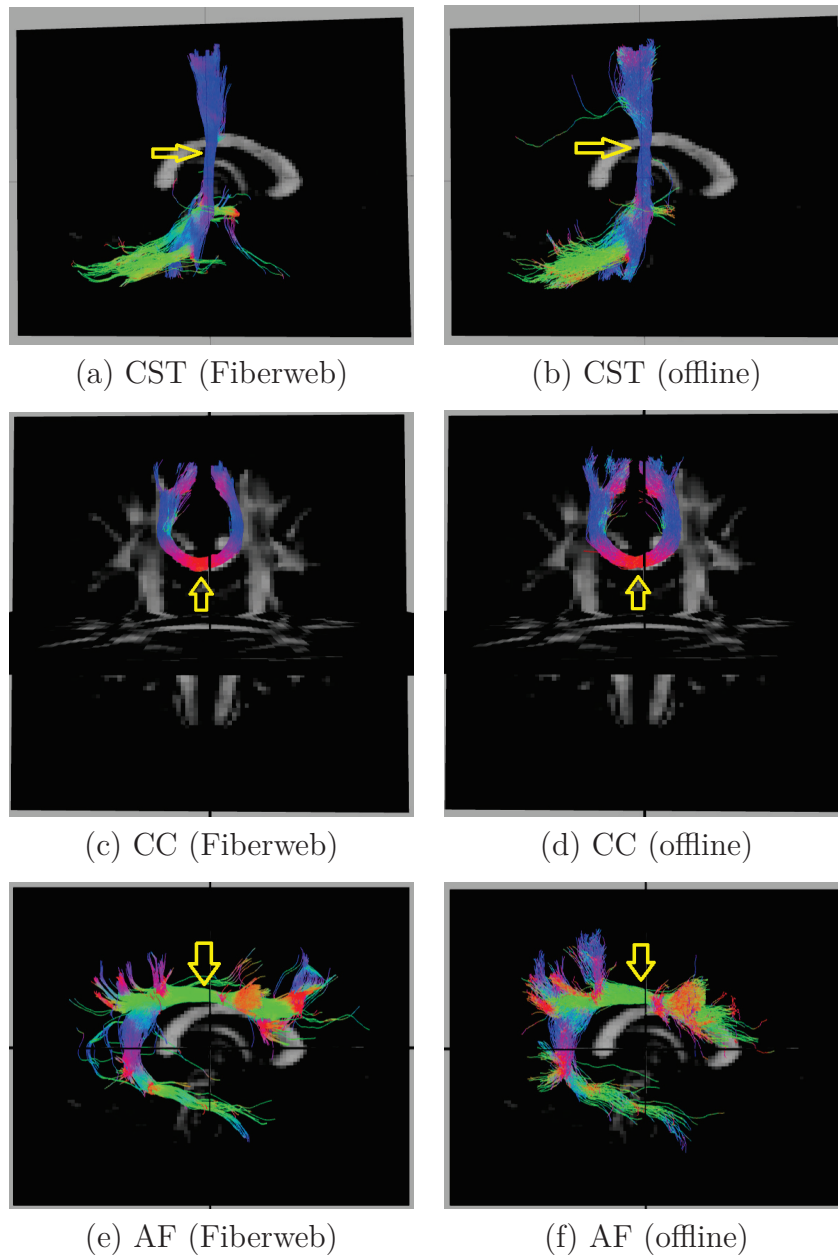


figure 3.8 – Various bundles resulting from probabilistic tractography (real-time in the Fiberweb vs offline). Dice coefficients are : CST (0.865), CC (0.928) and AF (0.811). The yellow arrows show the location of seeding regions.

3.3. RESULTS

impact the framerate. However, using stricter parameters can speed up the algorithm since the streamlines will be shorter in general. Furthermore, just like the selection, the problems with the framerate only exist when moving the box around, since this is when the tractography is computed.

3.3.3 Probabilistic tractography : online vs offline

The proposed probabilistic real-time tractography algorithm is based on a field of vectors (cartesian coordinates) of directions and their associated uncertainty angle. This data is extracted from fiber ODFs and is compared to probabilistic tracking based on fODF ([25]). In the case of the data that was used in this work, the uncertainty angles were extracted at 35% of their respective maxima value (see Figure 3.6).

To determine whether or not this proposed tracking was valid, both probabilistic algorithms were used to generate 1,000 streamlines from the same regions (Figure 3.8), represented by a box of 2x2x2 voxels. This was done with the help of a selection box in Fiberweb and with the use of a seeding mask for the offline algorithm. The parameters for both algorithms were the default ones. This was performed for three different bundles. The first bundle compared is the corticospinal tract (CST). It is important to note that both CSTs have projections to the cerebellum, which are most likely part of the corticopontine tract. These are not filtered out. All streamlines produced from the seed region are shown. The second compared bundle is a part of the corpus callosum (CC) under the primary motor cortex. The last bundle is the arcuate fasciculus (AF). In each case, the reconstructed bundles are qualitatively very similar. Overall, the online version always seems to produce slightly smaller and conservative bundles, with less fanning.

tableau 3.1 – Comparison between offline and online algorithms.

Bundles	Weighted Dice coefficient
CST	0.865
CC	0.928
AF	0.811

Bundles similarities were also quantitatively scored using a weighted Dice ratio. Table 3.1 shows the overlap ratio between the offline and online versions of each bundle

3.4. DISCUSSION

that was evaluated. All three scores range between 81% and 93%. These are very good overlap scores, considering the fact that both tracking algorithms use a different tracking equation and that the real-time Fiberweb version only uses peaks and their uncertainty as opposed to the full fODF. This is also the reason why it is not expected to find a perfect agreement between the two algorithms. Also, this indicates that overall, the two techniques share an excellent overlap varying slightly depending on the bundle being reconstructed. This means that our real-time probabilistic tractography algorithm rapidly produces an excellent approximation to a state of the art offline probabilistic tractography.

3.4 Discussion

Medical imaging software tools are often hard to install and hard to use for non-technical users. Fiberweb bypasses that problem by being built for the web. This makes the application cross-platform, and ready to use without the need for the installation of anything outside of having a web browser that is up-to-date. Furthermore, out of the few visualization web applications that already exist, none allow for real-time interaction and processing such as selection or real-time tractography.

3.4.1 New probabilistic algorithm

In this article, a modified version of an existing real-time tractography algorithm was introduced, making it possible to use it in a probabilistic way. This allows for a new way to visualize data in real time. It was validated qualitatively and quantitatively against a state of the art offline tractography algorithm. When compared visually, one can see that the offline version of the bundles recreated tend to look more probabilistic (more fanning and more spurious streamlines). This is due to multiple factors. It is also important to remember that both algorithms use a different tracking equation and that the real-time FiberWeb version only uses peaks and an uncertainty as opposed to the full fODF.

The uncertainty angle has the biggest impact on how probabilistic the algorithm will be. Using a 100% threshold, the tractography would give results identical to the

3.4. DISCUSSION

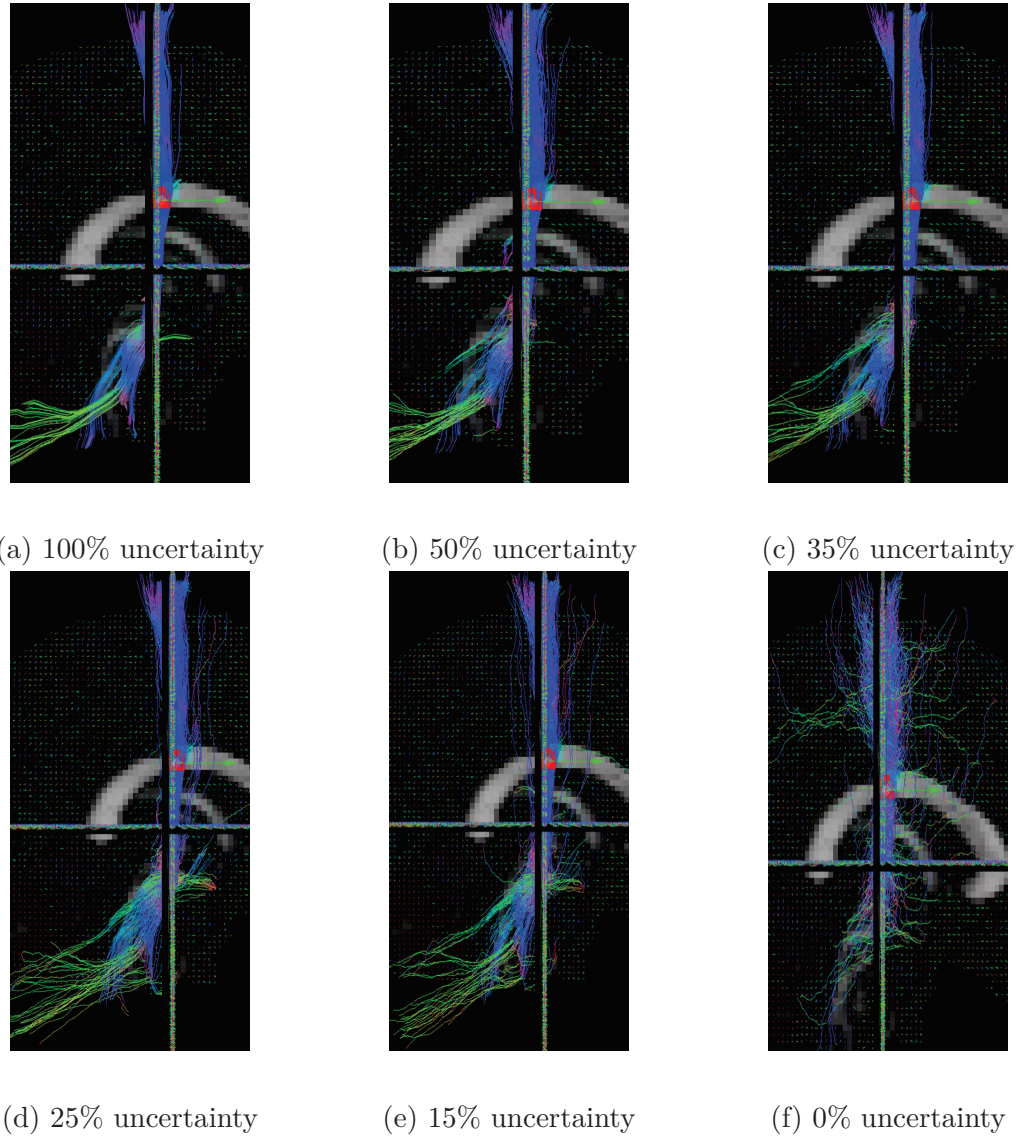


figure 3.9 – Various bundles resulting from probabilistic tractography, each with a different uncertainty percentage.

3.4. DISCUSSION

deterministic version of the algorithm. Inversely, using a 0% threshold, the tracking would take any random direction part of the fiber ODF. In our case, 35% was used because it yielded the best Dice quantitative scores, while still looking probabilistic. In Figure 3.9, it can be noted that lowering the uncertainty threshold induces more jaggy streamlines, while also adding more spurious streamlines.

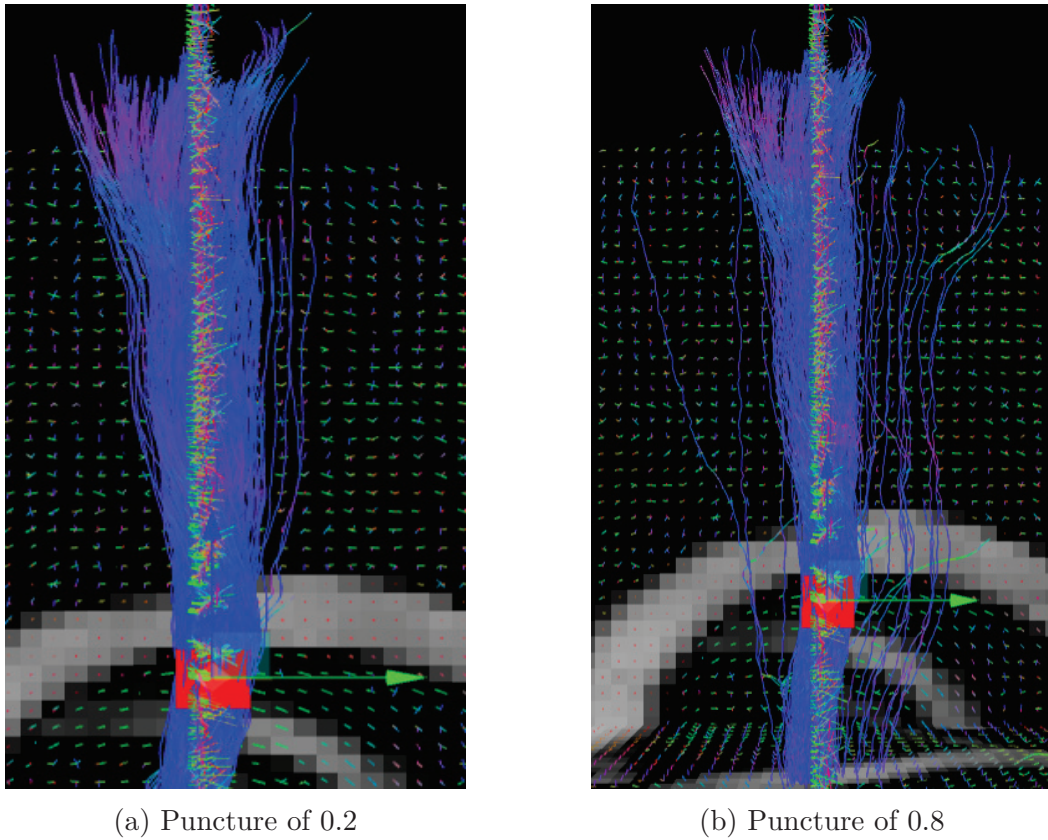


figure 3.10 – Probabilistic RTT of the CST for different puncture values.

The puncture parameter of the real-time tractography algorithm can modify the effect of the uncertainty angle since it differently weighs the incoming and next directions. A higher puncture value will make it look more probabilistic, but also generate more noisy spurious streamlines. Results in this paper were generated using a value of 0.2 since it is the suggested value for the deterministic real-time tractography algorithm ([11]). In Figure 3.10, it can be seen that the changes of direction are much smoother when the puncture value is of 0.2.

3.4. DISCUSSION

The resolution of the sphere on which the fiber ODFs are mapped affects the accuracy of the uncertainty angle, even though it is the factor that has the least impact on the final result. Results presented in this paper were generated using a sphere with 724 symmetrical vertices. A sphere with a lower resolution would give us uncertainty angles that are less representative of the user-inputted percentage value of a maxima, and not even an accurate maxima. The difference between two different sphere resolutions to display the same fODF can be seen in Figure 3.11.

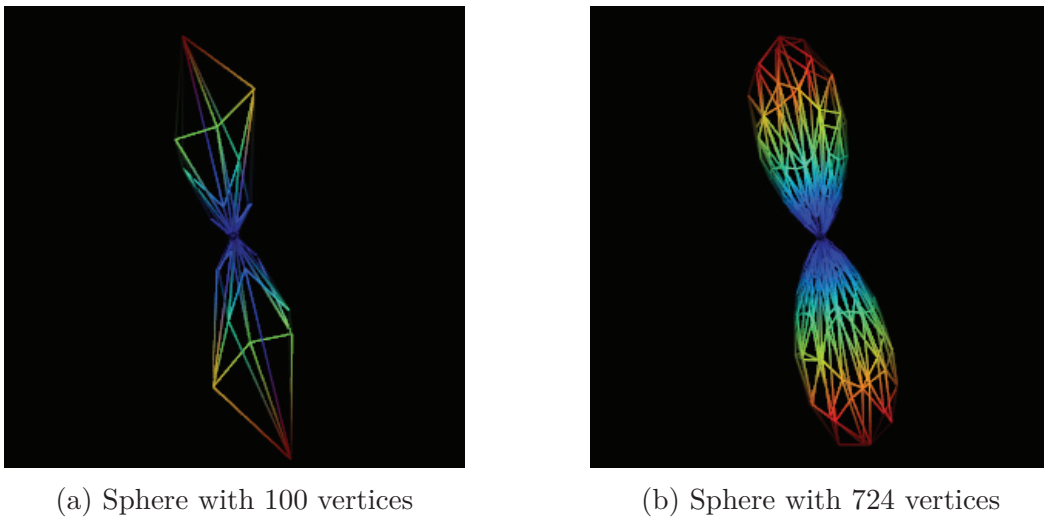


figure 3.11 – fODF represented by spheres of different resolutions.

3.4.2 Applications

One of the applications of FiberWeb is quality assurance. With the possibility to access a medical visualization tool anywhere or on any platform, without the need for anything else, it will make it easier to validate and check smaller datasets. For instance, suppose one has 1000 subjects in a tract-based white matter study. One can imagine using the FiberWeb to rapidly inspect tractograms through a web quality control interface such as CBRAIN ([51]). Alternatively, before doing a full-brain tractography of a million streamlines, it would be possible to explore smaller sets of 25,000 streamlines beforehand really quickly, especially since parameters used for tractography can change depending on the subject or region ([45], [11]). Furthermore,

3.4. DISCUSSION

with the help of the selection feature, it would be possible to validate the presence of certain bundles.

Fiberweb could also be used as a teaching tool. For example, one could imagine a scenario where a teacher in medical school would want to illustrate a few fiber bundles to his class without having prepared for it. This could be achieved simply by opening a web page, loading his medical brain data and using the visualisation, selection or real-time tracking features of Fiberweb to display the bundles. Then, by just sharing the data he used for the demonstration, every student could be able to review and explore it on their own device without the need to install an external software.

The new probabilistic algorithm allows for a new way to explore data without the need of having huge streamlines datasets, especially in the case where a specific region of interest is the sole object of the exploration. For example, in neurosurgical planning, it could be of great interest to use Fiberweb. Around regions of oedema, necrosis and mass effect due to the presence of a tumor, standard tracking parameters no longer hold. It is important to be able to reduce the FA threshold, change the step size and explore the uncertainty of the fODF peaks to present a more realistic representation of the data to the surgeon. Having both deterministic and probabilistic RTT could help more easily assess the effect of the tumor mass effect and oedema on the peaks, diffusion measures and corresponding reconstructed streamlines. It could be the first step towards showing the surgeon the most probable and improbable streamlines around and inside the tumor region of interest.

3.4.3 Future work

The first goal of Fiberweb is to support larger datasets. Unfortunately, it is mostly tied with the evolution of the varying browsers. Another main goal is making Fiberweb work just as well on tablets as in the browser. With the recent release of WebGL 2.0, one of the first task that will need to be done is to upgrade Fiberweb to use this version, mainly through the Three.js library. WebGL 2.0 will bring performance upgrades compared to the 1.0 version. For example, it will enable the use of 3D textures instead of only 2D, which is great considering anatomies are just like 3D textures. Currently, WebGL 2.0 was only announced, and it will probably be a while before

3.5. CONCLUSION

it is widely available. Another major focus of Fiberweb will be to add new features. Currently, only selection is available to interact with streamlines datasets. It would be interesting to add the possibility to use more than one selection box to make the streamlines selection possibly more precise. It would also be interesting to add exclusion boxes for the same reason. Additionally, it would be great to add segment based filtering ([30]) to Fiberweb to make sure the selection would work well with compressed streamlines. Since performance is key, making Fiberweb multi-threaded would be a good way to gain a lot of optimization, like splitting the rendering and the algorithms into separate threads. Also, it would be interesting to add streamlines compression and clustering during load time ([48]), allowing us to load bigger datasets. Finally, the last goal of Fiberweb will be to allow more file types to be used, since currently, only NIFTI and TrackVis files are supported.

3.5 Conclusion

In this article, Fiberweb, a new browser-based medical dMRI tractography visualization tool, was presented. This web application, unlike others of its kind, allows the user to interact with his data through virtual dissection with the help of a selection box, and through streamlines reconstruction in real time. Furthermore, it introduces a new real-time probabilistic tractography algorithm. It was shown to perform very closely to a state of the art probabilistic algorithm, both qualitatively and quantitatively. This makes data exploration efficient, simple and fast. This is especially appropriate when you want to look at a smaller region in the brain, or know the effect of anomalies, like a tumor, lesion or oedema. This new tool will be particularly important to do quick quality validation imaging datasets, without the need to install any external software, which are sometimes restricted to a specific platform. You can see Fiberweb in action at : www.imeka.ca/FiberWeb.

Disclosure/Conflict-of-Interest Statement

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of

3.5. CONCLUSION

interest.

Acknowledgments

We would like to thank Matthieu Dumont and Nil Goyette for their help with the logic behind the implementation of Fiberweb, and Jean-René Bélanger for his constant support. We would also like to thank the funding sources that made this work possible : Imeka, CRSNG, FRQNT and NSERC.

If you don't learn from your mistakes, then they become regrets.

JOHN CENA

Conclusion

Contributions

Tout au long de ce mémoire, nous avons présenté un nouveau logiciel de visualisation d'imagerie médicale du cerveau. Ce nouvel outil utilise la portabilité du web afin d'éliminer certains problèmes traditionnels qu'ont les logiciels, comme être spécifique à un système d'opération. Tous les pré-requis du Fiberweb sont directement à l'intérieur de votre fureteur internet, qu'on soit sous Windows, OSX ou encore Linux. Comme mentionné auparavant, il existe déjà plusieurs applications web permettant de visualiser des données d'imagerie du cerveau, que ce soit des anatomies ou bien des fibres. Cependant, le Fiberweb permet également de traiter ces données. Il est possible d'effectuer de la dissection virtuelle à l'aide d'une boîte de sélection. Il est également possible de faire de la tractographie en temps réel. En plus de la version de l'algorithme qui est déterministe, le Fiberweb en introduit une autre, celle-ci, probabiliste. Il a été montré que ce nouvel algorithme est comparable à une version qui ne fonctionne pas en temps réel. Cela permet d'explorer ses données d'une nouvelle façon. Le Fiberweb pourra être utile au domaine de la chirurgie notamment dans la préparation des opérations. De plus, cette application a le potentiel d'être utilisée sur des tablettes ou tout autre appareil portable, puisqu'il est facile d'accéder au web sur ceux-ci. On peut retrouver des vidéos du Fiberweb en action au www.imeka.ca/FiberWeb.

Problèmes ouverts et non-résolus

Tout au long de son développement, le Fiberweb a également amené son lot de nouveaux problèmes. Le plus important qui a été rencontré est la limite de la taille des

3.5. PROBLÈMES OUVERTS ET NON-RÉSOLUS

jeux de données que le Fiberweb peut utiliser. Cependant, cette limite n'est pas due à l'implémentation de l'application. En effet, la majorité des fournisseurs internet limitent la quantité de mémoire qu'ils peuvent utiliser. C'est un problème qui touche particulièrement les fibres, puisque ce type de données prend généralement plus d'espace en mémoire (environ 20mb pour 10,000 "streamlines" ; environ 200mb pour 100,000 "streamlines" ; environ 2gb pour 1,000,000 "streamlines"). Ce problème ne pourra être réglé que d'une seule façon : avec l'évolution des fournisseurs internet en général, car leur limite de mémoire devrait continuer d'augmenter.

De plus, que ce soit pour les interactions que l'on peut présentement pratiquer à l'intérieur du Fiberweb, ou bien pour de futures interactions, il sera important d'améliorer la performance de l'application. Cela pourra être fait de multiples façons. Tout d'abord, WebGL 2.0 a récemment été dévoilé. Basé sur OpenGL ES 3.0, cette nouvelle version amènera son lot de nouveautés et d'améliorations. Par exemple, les textures 3D seront supportées. Cela permettrait d'avoir une seule texture pour afficher une anatomie, plutôt qu'en utiliser plusieurs en mode 2D. De plus, une autre façon simple d'améliorer la performance du Fiberweb serait de le rendre "multi-threader". En séparant les tâches de rendu et les algorithmes d'interaction dans des "threads" différentes, on gagnerait beaucoup de vitesse d'affichage. En effet, puisque le Fiberweb est présentement "single threaded", lorsqu'un algorithme devient plus lourd en terme de performance (on peut penser à la tractographie en temps réel avec beaucoup de "seeds"), l'affichage est également affecté. Cela permettrait donc à la fréquence d'images de ne pas souffrir et de rester constante peu importe l'interaction en cours.

Ensuite, il faudra mettre le Fiberweb entre les mains des neurochirurgiens et cliniciens. Puisque ce sont eux qui seront les utilisateurs de ce logiciel, il est important d'obtenir leurs impressions, principalement par rapport à la facilité d'utilisation de l'application, ainsi que l'interface usager. Il est primordial que le Fiberweb soit simple d'utilisation, que les options disponibles soient facilement visibles et reconnaissables grâce à l'interface, et que peu importe l'action désirée, que la façon de l'exécuter soit intuitive.

Finalement, il y a plusieurs autres fonctionnalités qui devraient faire partie du Fiberweb afin d'augmenter la "quality of life" d'utilisation de l'application. Par exemple, les types de fichiers qui sont présentement supportés sont les NiFTI ainsi que les fi-

3.5. CONCLUSION GÉNÉRALE

chiers Trackvis. Il serait intéressant d'en supporter d'autres, comme les fichiers tck, un autre format de fibres. De plus, afin de pouvoir faire des dissections plus précises, il faudra faire des ajouts au mode de sélection. On devra ajouter la possibilité d'utiliser plusieurs boîtes de sélection, ainsi que des boîtes d'exclusion, permettant une nouvelle façon de disséquer.

Conclusion générale

En conclusion, le Fiberweb est un nouvel outil qui ouvre de nouveaux horizons pour la visualisation de données d'imagerie médicale. C'est un outil qui convient particulièrement pour faire l'assurance et le contrôle qualité. En effet, comme le Fiberweb est présentement limité en terme de grosseurs de jeux de données pouvant être chargés, il est efficace et rapide pour pouvoir explorer de plus petits fichiers. De plus, l'utilisation du Fiberweb pourra aider lors du transfert clinique d'un patient. Puisque qu'aucune installation n'est requise et que le logiciel est fonctionnel sur n'importe quel navigateur web standard, il est simple pour un médecin d'inspecter le dossier d'un nouveau patient. Pour terminer, le futur laisse présager des choses excitantes pour le Fiberweb. Pour le moment, il a été principalement développé pour être utilisé sur un ordinateur. Un des objectifs du Fiberweb sera aussi d'être fonctionnel sur d'autres médias, tel que la tablette, afin de rendre l'utilisation encore plus simple n'importe où. En pensant aux possibilités à plus long terme, il serait particulièrement intéressant d'améliorer les possibilités de collaboration que pourrait permettre le Fiberweb. L'évolution logique dans ce cas serait de tenter de transformer l'application pour permettre à différentes personnes de collaborer en temps réel à travers leur navigateur respectif. Bref, le Fiberweb démontre toutes les possibilités que le web peut offrir au domaine de l'imagerie médicale, des possibilités qui s'avèrent être fort intéressantes.

*Software is a great combination between artistry and
engineering.*

BILL GATES

Annexe A

**Fiberweb : a new web solution for
medical visualization and
interactions**

Fiberweb : a new web solution for medical visualization and interactions

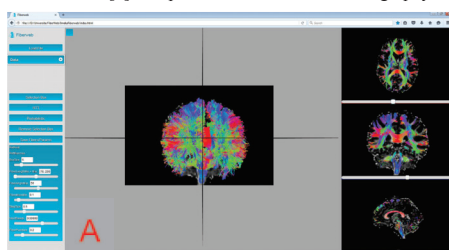
Louis-Philippe Ledoux¹, Jean-Christophe Houde¹, Félix C. Morency², Kevin Whittingstall¹, Jean-René Bélanger², Maxime Descoteaux¹

¹Université de Sherbrooke, Sherbrooke, Quebec, Canada; ²Imeka, Sherbrooke, Quebec, Canada

Target audience – Researchers, physicians and clinicians who need to visualize anatomical datasets and tractography streamlines rapidly and simply in a web browser. It is also of interest to people who want to do quick quality assurance on their data or perform real-time tractography.

Introduction – A novel browser-based solution for medical data visualization and interaction is proposed: *Fiberweb*. WebGL [1] is a recent technology that has not yet been fully explored for medical imaging visualization. Very few software tools allow for medical data visualization in the web browser [2][3][4][5] and none of the existing tool supports efficient data interaction such as streamline segmentation and real-time deterministic [6] and probabilistic tractography. Fiberweb currently supports anatomical dataset, whole brain tractograms, individual bundle and fiber ODF (fODF) peaks orientations [7].

Methods – Fiberweb was made possible due to the release of WebGL and the HTML5 standard. WebGL is a low-level graphics library for 3D rendering in the browser. To simplify development, we use the Three.js [8] high-level library, which wraps WebGL calls and makes it more convenient to use shaders, manage scenes and render primitives. An octree space partitioning data structure is filled at load time to reduce the search area when filtering streamlines. This makes real-time selection of streamlines possible without performance degradation. Fiberweb also supports deterministic [6] and probabilistic real-time tractography. Both algorithms require a pre-computed dataset of peaks (e.g. extracted from the fODFs), as well as a fractional anisotropy mask. The probabilistic algorithm also requires an uncertainty angle for each peak.



While the deterministic algorithm directly uses the peaks, the probabilistic algorithms selects its next tracking direction in a cone of uncertainty formed from the uncertainty angle associated with the peak. In addition to selection and real-time tractography, it is also possible to visualize the streamlines intersecting a slice of the anatomy.

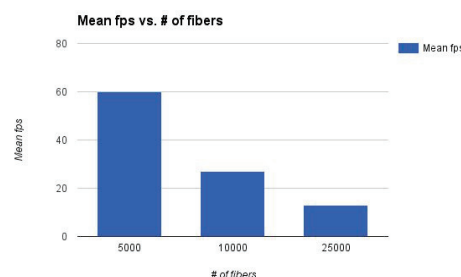


Figure 1 : Mean framerate in regards to # of fibers

Results – Visualization: Considering displaying an anatomical dataset as the baseline, simultaneously displaying fODFs peaks datasets does not affect the framerate. Image 1a) shows the interface when simultaneously displaying streamlines and an anatomical dataset. The 3D view directly allows for picking, interaction and selection, and the 2D renderers (axial / coronal / sagittal) show how streamlines intersect the anatomical dataset. Displaying streamlines is the main bottleneck of the application. Both the rendering framerate and the maximal streamlines loading capacity are affected by the number of streamlines and the amount of points per streamline. One should note that to avoid bias, the average number of points per streamline is roughly the same across all our datasets. Figure 1 shows that, for very small datasets, doubling the number of streamlines approximately halves the framerate. Going over 25,000 streamlines does not affect the framerate. However, reaching around 80,000 streamlines starts creating memory issues due to browser limitations. This makes it currently impossible to load very large datasets, unless you use a browser with a higher memory limit (like the Firefox Nightly version [9]). Current results are shown on raw streamlines. Using streamline compression [10] and grouping of similar streamlines [11] could help the framerate and fluidity of the application in the future.

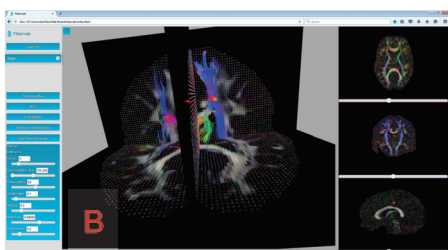


Image 1 : a) Full brain and its corresponding anatomy
b) Probabilistic real-time tractography

Interaction: Streamline interaction works in real-time. Whether we show the intersected fibers or not does not affect the framerate. The real-time tractography framerate starts dropping when we go over 1,000 seeds inside the seeding box. The other parameters do not have a noticeable effect on the framerate, neither does the choice of the algorithm (deterministic versus probabilistic).

Conclusion – We present a new tool that enables both visualization and interaction with diffusion MRI data directly inside the browser. Currently, it is more suitable for quality assurance and control due to the limitations in the sizes of the data the software can handle. For the future, Fiberweb is headed toward two main goals: supporting larger datasets and working on mobile devices like tablets.

References – [1] <https://www.khronos.org/webgl/> [2] <https://code.google.com/archive/p/brainngl/> [3] <http://slicedrop.com/> [4] <http://openscience.cbs.mpg.de/brainnetworks/> [5] <http://onpub.cbs.mpg.de/> [6] Chamberland et al., Frontiers 2014 [7] Descoteaux, Wiley EEEE 2015 [8] <https://github.com/mrdoob/three.js/> [9] <https://nightly.mozilla.org/> [10] Presseau et al, ISMRM 2014 [11] Garyfallidis et al, Frontiers 2012

*No matter what people tell you, words and ideas can change the
world.*

ROBIN WILLIAMS

Annexe B

Supplemental data

Fiberweb : diffusion visualization and processing in the browser

Louis-Philippe Ledoux^{1,*}, Felix C. Morency², Martin Cousineau^{1,6}, Jean-Christophe Houde², Kevin Whittingstall^{3,4,5} and Maxime Descoteaux^{1,2,3,*}

¹Sherbrooke Connectivity Imaging Lab, Computer Science Department, Faculty of Science, University of Sherbrooke, Sherbrooke, QC, Canada

²Imeka, Sherbrooke, QC, Canada

³Centre de Recherche CHUS, University of Sherbrooke, Sherbrooke, QC, Canada

⁴Department of Nuclear Medicine and Radiobiology, University of Sherbrooke, Sherbrooke, QC, Canada

⁵Department of Diagnostic Radiology, University of Sherbrooke, Sherbrooke, QC, Canada

⁶Videos & Images Theory and Analytics Laboratory, University of Sherbrooke, Sherbrooke, QC, Canada

Supplementary code

Here is the code allowing us to compute peaks of the fODFs, as well as their uncertainty alpha parameter.

B.0. SUPPLEMENTARY CODE

```
#!/usr/bin/env python
from __future__ import print_function
import os
import numpy as np
import nibabel as nib
import math

import argparse
import os
import logging
import warnings

import time
from dipy.reconst.shm import sph_harm_lookup
from dipy.core.sphere import Sphere
from dipy.data import get_sphere
from dipy.reconst.peaks import peak_directions
from dipy.core.ndindex import ndindex

def load(input) :
    img = nib.load(input)
    return img.get_data(), img.affine

def find_order(data) :
    return int((-3 + np.sqrt(1 + 8 * data.shape[-1])) / 2)

def get_B_matrix(order, sphere, basis_type) :
    sph_harm_basis = sph_harm_lookup.get(basis_type)
    if sph_harm_basis is None:
        raise ValueError("Invalid basis name.")
```

B.0. SUPPLEMENTARY CODE

```
B, m, n = sph_harm_basis(order, sphere.theta, sphere.phi
    )
return B

def get_maximas(data, sphere, B, threshold) :
    sf = np.dot(data, B.T)
    if False:#sf.sum() != 0:
        print(sf)
    directions, values, indices = peak_directions(sf, sphere
        , threshold)
    if False:#sf.sum() != 0:
        print(values)
        itemIndex = np.where(sf == values)
        print(itemIndex)
        print(indices)
        input()
    return directions, values, indices

def get_uncertainty(data, sphere, B, maximas,
    uncertaintyThreshold) :
    odf = np.dot(data, B.T)
    angles = np.array(())
    for aMaxima in maximas :
        #Init arrays
        verticesProcessed = np.array(())
        verticesToProcess = np.array(())
        verticesToProcess = np.insert(verticesToProcess,0,
            aMaxima)
        #Init scalars
        maximaOdfValue = odf[aMaxima]
        threshold_not_met = True
```

B.0. SUPPLEMENTARY CODE

```
bestOdfValue = maximaOdfValue
bestVertex = aMaxima
bestDifferenceValue = maximaOdfValue
targetValue = (uncertaintyThreshold * maximaOdfValue
)

while threshold_not_met and verticesToProcess.shape
[0] > 0 : ##Full Width Half Max
    #Init array
    neighboursToAdd = np.array(())

    #Check for best result
    for aVertex in verticesToProcess :
        if abs(odf[aVertex] - targetValue) <
            bestDifferenceValue :
            bestOdfValue = odf[aVertex]
            bestVertex = aVertex
            bestDifferenceValue = abs(odf[aVertex] -
                targetValue)

        #Find neighbours
        if odf[aVertex] > (targetValue - 0.000001) :
            neighboursToAdd = get_neighbour_vertices
                (aVertex, sphere, verticesProcessed,
                neighboursToAdd, verticesToProcess)

    #Handle vertices
    verticesProcessed = np.concatenate((
        verticesProcessed, verticesToProcess), axis=0)
    verticesToProcess = neighboursToAdd

if aMaxima == bestVertex :
```


B.0. SUPPLEMENTARY CODE

```
        angles = np.insert(angles,angles.shape[0],0)
    elif uncertaintyThreshold < 0.000001 :
        angles = np.insert(angles,angles.shape[0],math.
            pi/2)
    else :
        angles = np.insert(angles,angles.shape[0],
            get_angle( np.array([0,0,0]), sphere.vertices
                [aMaxima], sphere.vertices[bestVertex] ))

    return angles

def get_angle( sphereCenter, peakPos, halfMax ) :
    v1 = np.subtract(peakPos, sphereCenter)
    v2 = np.subtract(halfMax, sphereCenter)
    return math.acos(np.dot(v1,v2) / (np.linalg.norm(v1) *
        np.linalg.norm(v2)))

def get_neighbour_vertices(aVertex, aSphere,
    aVerticesProcessed, aNeighboursIndex, aVerticesToProcess)
    :
    neighboursIndex = aNeighboursIndex
    listOfFaces = np.where(aSphere.faces == aVertex)
    for i in range(listOfFaces[0].shape[0]) :
        faceId = listOfFaces[0][i]
        aVertexId = listOfFaces[1][i]
        for j in range(3):
            currentIndex = aSphere.faces[faceId][j]
            if j != aVertexId and np.where(neighboursIndex
                == currentIndex)[0].shape[0] == 0 and np.
                where(aVerticesProcessed == currentIndex)[0].
                shape[0] == 0 and np.where(aVerticesToProcess
                    == currentIndex)[0].shape[0] == 0 :
```

B.0. SUPPLEMENTARY CODE

```
        neighboursIndex = np.insert(neighboursIndex,
                                     neighboursIndex.shape[0],currentIndex)

    return neighboursIndex

def get_maps(data, mask, args) :
    maxNbOfMaximas = 20
    nufo_map = np.zeros(data.shape[0:3])
    afd_map = np.zeros(data.shape[0:3])
    uncertainty_map = np.zeros((data.shape[0],data.shape[1],
                                data.shape[2],maxNbOfMaximas))
    order = find_order(data)
    sphere = get_sphere(args.number)
    B = get_B_matrix(order, sphere, args.basis)

    count = 0

    for index in ndindex(data.shape[0:3]) :
        count += 1
        if mask[index] > 0 :
            if np.isnan(data[index]).any() :
                print('NaN')
            else :
                maximas, afd, indices = get_maximas(data[
                    index], sphere, B, args.threshold)
                angles = get_uncertainty(data[index], sphere
                    , B, indices, args.uncertainty)
                angles = np.expand_dims(angles,axis=1)
                awesomeIndices = np.argsort(angles,axis=0)
                awesomeIndices = awesomeIndices[::-1]
                angles = angles[awesomeIndices[:,0]]
                maximas = maximas[awesomeIndices[:,0]]
```

B.0. SUPPLEMENTARY CODE

```
directionsPlusAngles = np.concatenate((
    maximas,angles),axis=1).flatten()
if directionsPlusAngles.shape[0] <
    maxNbOfMaximas:
    directionsPlusAngles = np.concatenate((
        directionsPlusAngles,np.zeros(
            maxNbOfMaximas-directionsPlusAngles.
            shape[0])),axis=0)
if directionsPlusAngles.shape[0] >
    maxNbOfMaximas:
    directionsPlusAngles =
        directionsPlusAngles[0:maxNbOfMaximas
    ]

    uncertainty_map[index] =
        directionsPlusAngles
print('Progress [' + str(int(count / float(data.size
    /data.shape[3]) * 100)) + '%]...', end='\r')

return uncertainty_map

DESCRIPTION = """
    Script to compute the uncertainty peaks map from fiber
    ODFs.
    """

def buildArgsParser():

    p = argparse.ArgumentParser(description=DESCRIPTION)
```

B.0. SUPPLEMENTARY CODE

```
p.add_argument('input', action='store', metavar='fODFs',
              type=str,
              help='Path of the fODF volume in
                    spherical harmonics (SH).')

p.add_argument('--basis', action='store', dest='basis',
              metavar='string', required=False, default
              ='fibernav',
              type=str, help='Basis used for the SH
                    coefficients. Must \
                    be either mrtrix or fibernav (default).')

p.add_argument('--mask', action='store', dest='mask',
              metavar='', required=False, default=None,
              type=str,
              help='Path to a binary mask. Only the
                    data inside the mask \
                    will be used for computations and
                    reconstruction.')

p.add_argument('-t', dest='threshold', action='store',
              type=float, default='0.25',
              help='Threshold on fodf amplitude in
                    percentage (default : 0.25)')

p.add_argument('-n', dest='number', action='store', type
              =str, default='symmetric724',
              help='Number of directions uniformly
                    distributed on sphere (default : 100)
                    ')

```

B.0. SUPPLEMENTARY CODE

```
p.add_argument('-u', dest='uncertainty', action='store',
               type=float, default=1.0,
               help='Uncertainty percentage /1.0')

p.add_argument('-f', action='store_true', dest='
               overwrite', required=False,
               help='If True, the saved files volume
                     will be overwritten \
                     if they already exist.')

return p

def main():
    parser = buildArgsParser()
    args = parser.parse_args()

    tps1 = time.clock()

    # Load input image
    data, affine = load(args.input)

    if args.mask is None :
        mask = np.ones(data.shape[0:3])
    else :
        mask, affine2 = load(args.mask)

    # Compute uncertainty map
    uncertainty_map = get_maps(data, mask, args)

    # Save result
```

B.0. WEIGHTED DICE COEFFICIENT

```
img = nib.Nifti1Image(np.array(uncertainty_map, 'float32
    '), affine)
nib.save(img, 'maximasUncertainties'+ str(int(args.
    uncertainty*100)) + '.nii')

tps2 = time.clock()
print('Time elapsed : ' + str((tps2 - tps1)/60.0) + '
    min.')
```

```
if __name__ == "__main__":
    main()
```

Weighted Dice coefficient

In order to measure the overlap of two bundles reconstructed differently, we used a slightly modified version of the Dice coefficient [19], as proposed by Cousineau et al. [13]. Let W_i be a bundle reconstructed using a certain tracking algorithm and W_j another bundle extracted from another algorithm. Note that in this study, W_i and W_j are the same anatomical bundle. The standard Dice coefficient between W_i and W_j is

$$D(W_i, W_j) = \frac{2 \sum_v (W_i \cap W_j)_v}{\sum_v W_{i,v} + \sum_v W_{j,v}}. \quad (\text{B.1})$$

where W_i and W_j contain binary values (1 inside the bundle and 0 otherwise) and v is a voxel index. As is, the Dice coefficient greatly penalizes for spurious streamlines that would be far from the core of the bundle. Given that white matter bundles have more tracts in the middle than in their periphery, Cousineau et al. [13] proposed a weighted Dice coefficient which accounts for the number of streamlines per voxel. In that perspective, each voxel in W_i and W_j contains a value between 0 and 1 expressing

B.0. WEIGHTED DICE COEFFICIENT

the fraction of tracts passing through that position. The weighted Dice metric sums the voxels that overlaps in W_i and W_j and divide by the total sum of voxels :

$$D(W_i, W_j) = \frac{\sum_{v'} W_{i,v'} + \sum_{v'} W_{j,v'}}{\sum_v W_{i,v} + \sum_v W_{j,v}}, \quad (\text{B.2})$$

where v' stands for the voxels that are within the intersection of the W_i and W_j bundles. The weighted Dice gives a larger importance to areas with dense fibers.

Bibliographie

- [1] Alfred ANWANDER, Marc TITTEMEYER, D Yves von CRAMON, Angela D FRIEDERICI et Thomas R KNÖSCHE.
« Connectivity-based parcellation of Broca's area ».
Cerebral Cortex, 17(4):816–825, 2007.
- [2] Peter J BASSER et D LEBIHAN.
« Fiber orientation mapping in an anisotropic medium with NMR diffusion spectroscopy ».
Dans *11th Annual Meeting of the SMRM, Berlin*, page 1221, 1992.
- [3] Peter J BASSER, James MATTIELLO et Denis LEBIHAN.
« MR diffusion tensor spectroscopy and imaging. ».
Biophysical journal, 66(1):259, 1994.
- [4] Peter J BASSER, Sinisa PAJEVIC, Carlo PIERPAOLI, Jeffrey DUDA et Akram ALDROUBI.
« In vivo fiber tractography using DT-MRI data ».
Magnetic resonance in medicine, 44(4):625–632, 2000.
- [5] PJ BASSER, J MATTIELLO et D LEBIHAN.
« Diagonal and off-diagonal components of the self-diffusion tensor : their relation to and estimation from the NMR spin-echo signal ».
Proc. 11th Annu. Meet. SMRM, Berlin, 1:1222, 1992.
- [6] TEJ BEHRENS, MW WOOLRICH, M JENKINSON, H JOHANSEN-BERG, RG NUNES, S CLARE, PM MATTHEWS, JM BRADY et SM SMITH.
« Characterization and propagation of uncertainty in diffusion-weighted MR imaging ».
Magnetic resonance in medicine, 50(5):1077–1088, 2003.

BIBLIOGRAPHIE

- [7] Ricardo CABELLO.
« Three. js ».
URL : <https://github.com/mrdoob/three.js>, 2010.
- [8] Marco CATANI, Robert J HOWARD, Sinisa PAJEVIC et Derek K JONES.
« Virtual in vivo interactive dissection of white matter fasciculi in the human brain ».
Neuroimage, 17(1):77–94, 2002.
- [9] Marco CATANI et Marsel MESULAM.
« The arcuate fasciculus and the disconnection theme in language and aphasia : history and current state ».
Cortex, 44(8):953–961, 2008.
- [10] CEPHALICVEIN.
« Human Brain », 2016.
<http://cephalicvein.com/2016/09/human-brain/>.
- [11] Maxime CHAMBERLAND, Kevin WHITTINGSTALL, David FORTIN, David MATHIEU et Maxime DESCOTEAUX.
« Real-time multi-peak tractography for instantaneous connectivity display ».
Frontiers in neuroinformatics, 8:59, 2014.
- [12] Thomas E CONTURO, Nicolas F LORI, Thomas S CULL, Erbil AKBUDAK, Abraham Z SNYDER, Joshua S SHIMONY, Robert C MCKINSTRY, Harold BURTON et Marcus E RAICHLE.
« Tracking neuronal fiber pathways in the living human brain ».
Proceedings of the National Academy of Sciences, 96(18):10422–10427, 1999.
- [13] Martin COUSINEAU, Eleftherios GARYFALLIDIS, Marc-Alexandre CÔTÉ, Pierre-Marc JODOIN et Maxime DESCOTEAUX.
« Tract-profiling and bundle statistics : a test-retest validation study ».
2016.
- [14] Maxime DESCOTEAUX.
« *High angular resolution diffusion MRI : from local estimation to segmentation and tractography* ».
Thèse de doctorat, Citeseer, 2008.

BIBLIOGRAPHIE

- [15] Maxime DESCOTEAUX.
« Reconstruction et analyse d'images médicales - IRM de diffusion », 2014.
Présentation dans le cadre du cours IMN 530.
- [16] Maxime DESCOTEAUX.
« High Angular Resolution Diffusion Imaging (HARDI) ».
Wiley Encyclopedia of Electrical and Electronics Engineering, pages 1–25, 2015.
- [17] Maxime DESCOTEAUX, Rachid DERICHE, Thomas R KNÖSCHE et Alfred ANWANDER.
« Deterministic and probabilistic tractography based on complex fibre orientation distributions ».
Medical Imaging, IEEE Transactions on, 28(2):269–286, 2009.
- [18] Maxime DESCOTEAUX et Cyril POUPON.
« Diffusion-weighted MRI ».
Comprehensive Biomedical Physics, 3(6):81–97, 2014.
- [19] Lee R DICE.
« Measures of the amount of ecologic association between species ».
Ecology, 26(3):297–302, 1945.
- [20] Albert EINSTEIN.
« Über die von der molekularkinetischen theorie der wärme geforderte bewegung von in ruhenden flüssigkeiten suspendierten teilchen. ».
Ann. der Physik, 322:549, 1905.
- [21] Angela D FRIEDERICI.
« The brain basis of language processing : from structure to function ».
Physiological reviews, 91(4):1357–1392, 2011.
- [22] Angela D FRIEDERICI, THEPRODUCT(IMPLEMENTATION), T R(implementation) KNÖSCHE et R (WebGL) SCHURADE.
« The brain basis of language processing : from structure to function (interactive version) ».
URL : <http://onpub.cbs.mpg.de/>, 2012.
- [23] Eleftherios GARYFALLIDIS, Matthew BRETT, Bagrat AMIRBEKIAN, Ariel ROKEM, Stefan VAN DER WALT, Maxime DESCOTEAUX et Ian NIMMO-SMITH.

BIBLIOGRAPHIE

- « Dipy, a library for the analysis of diffusion MRI data ». *Frontiers in neuroinformatics*, 8:8, 2014.
- [24] Eleftherios GARYFALLIDIS, Matthew BRETT, Marta Morgado CORREIA, Guy B WILLIAMS et Ian NIMMO-SMITH.
« Quickbundles, a method for tractography simplification ». *Frontiers in neuroscience*, 6:175, 2012.
- [25] Gabriel GIRARD, Kevin WHITTINGSTALL, Rachid DERICHE et Maxime DESCOTEAUX.
« Towards quantitative connectivity analysis : reducing tractography biases ». *Neuroimage*, 98:266–278, 2014.
- [26] Daniel HAEHN.
« Slice : drop : Collaborative medical imaging in the browser ». Dans *ACM SIGGRAPH 2013 Computer Animation Festival*, pages 1–1. ACM, 2013.
- [27] Daniel HAEHN, Nicolas RANNOU, Banu AHTAM, Ellen GRANT et Rudolph PIENNAAR.
« Neuroimaging in the browser using the X Toolkit ». *Frontiers in Neuroinformatics*, 101, 2014.
- [28] Dennis M HEALY, Harrie HENDRIKS et Peter T KIM.
« Spherical deconvolution ». *Journal of Multivariate Analysis*, 67(1):1–22, 1998.
- [29] K HEUER, R SCHURADE et A ANWANDER.
« Brain Networks ». *URL : [http ://openscience.cbs.mpg.de/brainnetworks/](http://openscience.cbs.mpg.de/brainnetworks/)*, 2014.
- [30] M-A. Côté HOUDE, J-C. et M. DESCOTEAUX.
« How to avoid biased streamlines-based metrics for streamlines with variable step sizes ». *Proceeding of : International Society of Magnetic Resonance in Medicine (ISMRM)*, 2015.
- [31] IMÉKA.
« MI-Brain », 2016.
<http://www.imeka.ca/mi-brain>.

BIBLIOGRAPHIE

- [32] Andrei IRIMIA, Micah C CHAMBERS, Carinna M TORGERSON et John D VAN HORN.
« Circular representation of human cortical networks for subject and population-level connectomic visualization ».
Neuroimage, 60(2):1340–1351, 2012.
- [33] DK JONES, MA HORSFIELD et A SIMMONS.
« Optimal strategies for measuring diffusion in anisotropic systems by magnetic resonance imaging ».
Magn. Reson. Med., 42:515–525, 1999.
- [34] L KAY.
« Scene. js. ».
URL : <http://www.scenejs.org>, 2010.
- [35] Michael Misha KAZHDAN.
« FFTs in Graphics and Vision », 2015.
Presentation as part of a course <http://www.cs.jhu.edu/~misha/Spring15/17.pdf>.
- [36] Jan KLEIN, Ola FRIMAN, Markus HADWIGER, Bernhard PREIM, Felix RITTER, Anna VILANOVA, Gabriel ZACHMANN et Dirk BARTZ.
« Visual computing for medical diagnosis and treatment ».
Computers & Graphics, 33(4):554–565, 2009.
- [37] Paul C LAUTERBUR.
« Image formation by induced local interactions : examples employing nuclear magnetic resonance ».
pages 190–191, 1973.
- [38] Denis LE BIHAN et E BRETON.
« Imagerie de diffusion in-vivo par résonance magnétique nucléaire ».
Comptes-Rendus de l'Académie des Sciences, 93(5):27–34, 1985.
- [39] Denis LE BIHAN, Eric BRETON, Denis LALLEMAND, Philippe GRENIER, Emmanuel CABANIS et Maurice LAVAL-JEANTET.
« MR imaging of intravoxel incoherent motions : application to diffusion and perfusion in neurologic disorders. ».
Radiology, 161(2):401–407, 1986.

BIBLIOGRAPHIE

- [40] Chris MARRIN.
« WebGL specification ».
Khronos WebGL Working Group, 2011.
- [41] Donald JR MEAGHER.
Octree encoding : A new technique for the representation, manipulation and display of arbitrary 3-d objects by computer.
Electrical and Systems Engineering Department Rensselaer Polytechnic Institute
Image Processing Laboratory, 1980.
- [42] Klaus-Dietmar MERBOLDT, Wolfgang HANICKE et Jens FRAHM.
« Self-diffusion NMR imaging using stimulated echoes ».
Journal of Magnetic Resonance (1969), 64(3):479–486, 1985.
- [43] Susumu MORI, Barbara J CRAIN, VP CHACKO et Peter VAN ZIJL.
« Three-dimensional tracking of axonal projections in the brain by magnetic resonance imaging ».
Annals of neurology, 45(2):265–269, 1999.
- [44] ME MOSELEY, J KUCHARCZYK, J MINTOROVITCH, Y COHEN, J KURHANEWICZ, N DERUGIN, H ASGARI et D NORMAN.
« Diffusion-weighted MR imaging of acute stroke : correlation with T2-weighted and magnetic susceptibility-enhanced MR imaging in cats. ».
American Journal of Neuroradiology, 11(3):423–429, 1990.
- [45] Carlo PIERPAOLI, Peter JEZZARD, Peter J BASSER, Alan BARNETT et Giovanni DI CHIRO.
« Diffusion tensor MR imaging of the human brain. ».
Radiology, 201(3):637–648, 1996.
- [46] Cyril POUPON.
« *Détection des faisceaux de fibres de la substance blanche pour l'étude de la connectivité anatomique cérébrale* ».
Thèse de doctorat, 1999.
- [47] Caroline PRESSEAU, Pierre-Marc JODOIN, Jean-Christophe HOUDE et Maxime DESCOTEAUX.
« A new compression format for fiber tracking datasets ».
NeuroImage, 109:73–83, 2015.

BIBLIOGRAPHIE

- [48] F. RHEAULT, J-C. HOUDE et M. DESCOTEAUX.
« Real time interaction with millions of streamlines ».
Proceeding of : International Society of Magnetic Resonance in Medicine (ISMRM), 2015.
- [49] Adam RICHIE-HALFORD, Ariel ROKEM, Josh SMITH et Jason D YEATMAN.
« AFQ Browser ».
URL : <https://github.com/yeatmanlab/AFQ-Browser>, 2016.
- [50] Kiran K SEUNARINE et Daniel C ALEXANDER.
« Multiple fibers : beyond the diffusion tensor ».
Diffusion MRI : From quantitative measurement to in vivo neuroanatomy, 1:55–72, 2009.
- [51] Tarek SHERIF, Pierre RIOUX, Marc-Etienne ROUSSEAU, Nicolas KASSIS, Natacha BECK, Reza ADALAT, Samir DAS, Tristan GLATARD et Alan C EVANS.
« CBRAIN : a web-based, distributed computing platform for collaborative neuroimaging research ».
Recent Advances and the Future Generation of Neuroinformatics Infrastructure, page 102, 2015.
- [52] DG TAYLOR et MC BUSHELL.
« The spatial mapping of translational diffusion coefficients by the NMR imaging technique ».
Physics in medicine and biology, 30(4):345, 1985.
- [53] J TOURNIER, Fernando CALAMANTE, Alan CONNELLY et OTHERS.
« MRtrix : diffusion tractography in crossing fiber regions ».
International Journal of Imaging Systems and Technology, 22(1):53–66, 2012.
- [54] J-Donald TOURNIER, Fernando CALAMANTE et Alan CONNELLY.
« Robust determination of the fibre orientation distribution in diffusion MRI : non-negativity constrained super-resolved spherical deconvolution ».
NeuroImage, 35(4):1459–1472, 2007.
- [55] J-Donald TOURNIER, Fernando CALAMANTE, David G GADIAN et Alan CONNELLY.
« Direct estimation of the fiber orientation density function from diffusion-

BIBLIOGRAPHIE

- weighted MRI data using spherical deconvolution ». *NeuroImage*, 23(3):1176–1185, 2004.
- [56] David S TUCH, Timothy G REESE, Mette R WIEGELL, Nikos MAKRIS, John W BELLIVEAU et Van J WEDEEN.
« High angular resolution diffusion imaging reveals intravoxel white matter fiber heterogeneity ». *Magnetic Resonance in Medicine*, 48(4):577–582, 2002.
- [57] Ugur TÜRE, M Gazi YASARGIL, Allan H FRIEDMAN et Ossama AL-MEFTY.
« Fiber dissection technique : lateral aspect of the brain ». *Neurosurgery*, 47(2):417–427, 2000.
- [58] Olivier VAILLANCOURT, A BORÉ, G GIRARD et Maxime DESCOTEAUX.
« A fiber navigator for neurosurgical planning (neuroplanningnavigator) ». Dans *IEEE Visualization*, volume 231, 2010.
- [59] Vladimir VUKICEVIC.
« Canvas 3d : GL power, webstyle ». *Vladimir Vukicevic-Words*, November, 26, 2007.
- [60] W3C.
« Position Paper for the W3C Workshop on Web Applications and Compound Documents », 2004.
<https://www.w3.org/2004/04/webapps-cdf-ws/papers/opera.html>.
- [61] W3C.
« Shall we Adopt HTML5 as our specification text for review? », 2007.
<https://www.w3.org/2002/09/wbs/40318/htmlbg/results>.
- [62] W3C.
« HTML5 IS A W3C RECOMMENDATION », 2014.
<https://www.w3.org/blog/news/archives/4167>.
- [63] Van J WEDEEN, RP WANG, Jeremy D SCHMAHMANN, T BENNER, WYI TSENG, Guangping DAI, DN PANDYA, Patric HAGMANN, Helen D'ARCEUIL et Alex J de CRESPIGNY.
« Diffusion spectrum magnetic resonance imaging (DSI) tractography of crossing

BIBLIOGRAPHIE

- fibers ».
Neuroimage, 41(4):1267–1277, 2008.
- [64] VJ WEDEEN, TG REESE, DS TUCH, MR WEIGEL, JG DOU, RM WEISKOFF et D CHESSLER.
« Mapping fiber orientation spectra in cerebral white matter with Fourier-transform diffusion MRI ».
Dans *Proceedings of the 8th Annual Meeting of ISMRM, Denver*, page 82, 2000.
- [65] A YAMAMOTO, Y MIKI, S URAYAMA, Y FUSHIMI, T OKADA, T HANAKAWA, H FUKUYAMA et K TOGASHI.
« Diffusion tensor fiber tractography of the optic radiation : analysis with 6-, 12-, 40-, and 81-directional motion-probing gradients, a preliminary study ».
American Journal of Neuroradiology, 28(1):92–96, 2007.
- [66] Jason D YEATMAN, Robert F DOUGHERTY, Nathaniel J MYALL, Brian A WANDERL et Heidi M FELDMAN.
« Tract profiles of white matter properties : automating fiber-tract quantification ».
PloS one, 7(11):e49790, 2012.