Hierarchical visualization using fiber clustering

Moberts, B.

Award date:
2005

Technische Universiteit Eindhoven

Department of Mathematics and Computer Science

Master's Thesis

HIERARCHICAL VISUALIZATION USING FIBER CLUSTERING

by

Ing. B. Moberts

Supervisors: Dr. A. Vilanova
Prof.dr.ir. J.J. van Wijk

Eindhoven, June 2005

Abstract

Diffusion Tensor Imaging (DTI) is a Magnetic Resonance Imaging (MRI) technique for measuring diffusion in biological tissue. DTI data is difficult to visualize because of the high amount of information available in each sample point. A prominent DTI visualization technique is fiber tracking. The fiber tracking algorithm creates streamlines (fibers) that correspond to the major white matter fiber bundles in the brain. Initialization of the fiber tracking algorithm is done through the placement of seeds. The placement of these seeds can be done in two ways; either the user indicates a region of interest or the seeding is done throughout the whole volume. A problem with seeding throughout the whole volume is that the amount of fibers that is created is enormous. As a result, the display becomes cluttered, individual structures are virtually indistinguishable and it is very difficult to extract any useful information.

To overcome this problem, we use a clustering algorithm to organize the fibers into groups that are meaningful and anatomically correct. Two clustering methods are employed: hierarchical clustering and shared nearest neighbor clustering. The most appropriate method is determined by validating the cluster results using a manual classification of the fibers. We examine two kinds of validation methods: Receiver Operator Characteristic (ROC) Curves and external indices. Because these methods use different criteria for validation, they also give different results. In the context of fiber clustering, the goal is to find a validation method that meets the criteria of physicians. For this purpose, we present a new method based on the Adjusted Rand index, and we show that it is more suited to the task of fiber cluster validation. Finally, we use the new validation method to assess the quality of the segmentations produced by the various clustering methods.

Contents

Chapter 1

Introduction

1.1 Motivation

Diffusion Tensor Imaging (DTI) is a magnetic resonance technique for measuring diffusion in biological tissue. Diffusion is the result of randomly moving water molecules. Organized tissues such as muscles and the white matter in the brain restrict this movement in certain directions. By measuring the diffusion in different directions the underlying structures can be explored on a microscopic scale. In contrast, current state of the art MRI only shows the macrostructures. Information provided by DTI is used to investigate brain diseases, muscle structure and the development of the brain. A tool for visualizing DTI data was created in collaboration with the Maxima Medisch Center (MMC) in Veldhoven [2].

Diffusion can be represented by a second order tensor (a 3×3 symmetric matrix). DTI data is difficult to visualize because of the high amount of information available in each sample point. A very interesting and often used technique for visualizing DTI datasets is fiber tracking. Fiber tracking simplifies the tensor field to a vector field of the main diffusion direction. This vector field is then used as a velocity field into which particles are released. The paths these particles follow can be visualized as streamlines. When applied to brain data, the streamlines correspond to a good approximation to the major white matter fiber bundles [3].

Initialization of the fiber tracking algorithm is done through the placement of seeds. The placement of these seeds can be done in two ways; either the user indicates a region of interest (ROI) or the seeding is done throughout the whole volume. A problem with ROI fiber tracking is that important fibers may be missed due to the placement of the ROI. Also, in healthy human subjects the position of the major fiber bundles is known, but in patients some structures might not be in the expected position and can therefore be missed. With seeding throughout the whole volume this problem is avoided, but the amount of fibers that is created is enormous and the display becomes cluttered: individual structures are virtually indistinguishable and it is very difficult to extract any useful information.

1.2 Methods and approach

In order to overcome the visual cluttering and other difficulties related to seeding throughout the whole volume, this study investigates hierarchical visualization methods for streamlines. A cluster algorithm is used to organize the fibers into groups that are meaningful and anatomically correct. The enormous amount of individual fibers is reduced to a limited number of logical fiber clusters that are more manageable and usable. Once a clustering is obtained, the DTI data can be viewed at different levels of detail; a global view which shows the fiber clusters and a local view which shows the individual fibers of a specific cluster.

To assess the quality of the cluster results, we perform a limited validation by manually classifying the fibers into a number of groups that correspond to actual anatomical structures. The manual classification can be seen as a gold standard against which we compare the clusters from the cluster methods. A number of validation methods are examined, and we propose several improvements to make them more suitable for the task of fiber cluster validation.

The clustering and validation methods are then applied to DTI data sets of (healthy) human brains. The results of two clustering methods, hierarchical clustering and shared nearest neighbor clustering, are presented, validated and compared with each other.

1.3 Outline of the thesis

Chapter 2 provides background information on DTI. It discusses fiber tracking and other DTI visualization techniques. Chapter 3 reviews fiber cluster methods available from literature. Also, a cluster method that has not yet been used for fiber clustering is presented here. Chapter 4 describes the validation framework. Chapter 5 presents a comparison of two cluster methods. Chapter 6 contains the conclusion and future work sections.

Chapter 2

Diffusion Tensor Imaging

Diffusion Tensor Imaging (DTI) gives insight into the structure of the brain and other living tissue. This chapter provides an introduction to DTI and discusses the difficulty of correctly visualizing DTI data. First, section 2.1 gives a biological and mathematical overview of DTI. After that, several visualization techniques are discussed in section 2.2. Finally, a prominent visualization technique called fiber tracking is explained in more detail in section 2.3.

2.1 Basics

Diffusion Tensor Imaging (DTI) is a magnetic resonance technique that quantifies the average diffusion of moving water molecules in biological tissue [3]. This random movement of water molecules is caused by internal thermal energy and is known as Brownian motion. Certain tissues limit the movement of water molecules, reducing the distance they travel. By measuring the preferred direction of diffusion it is possible to reconstruct the underlying structure of the tissue. Due to its ability to measure this physical diffusion process, DTI allows visualization of micro-structures below the resolution of the scanner.

Tissue that lets molecules travel more easily in certain directions is called anisotropic. An example of anisotropic tissue is white matter in the brain. White matter consists of fiber tracts that connect regions of grey matter. In white matter water diffuses more in the direction of fiber tracts than in the perpendicular direction. Figure 2.1 shows how fiber tracts hinder the movement of molecules (indicated by arrows). In contrast to white matter in which diffusion is anisotropic, grey matter is largely isotropic: diffusion is equal in all directions. Other kinds of tissue that show anisotropic diffusion include muscles and the heart. In this thesis only DTI scans of the human brain are considered.

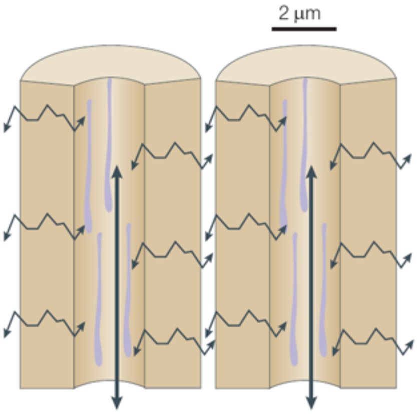


Figure 2.1: Anisotropic diffusion [3]

Diffusion can be represented by a 3×3 positive symmetric tensor:

$$\mathbf{D} = \begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{xy} & D_{yy} & D_{yz} \\ D_{xz} & D_{yz} & D_{zz} \end{pmatrix}$$

Diagonalization of this tensor gives three positive eigenvalues λ_1, λ_2, λ_3 (sorted in decreasing order) and corresponding eigenvectors $e\vec{v}_1$, $e\vec{v}_2$, $e\vec{v}_3$. The eigenvectors are orthogonal to each other and represent the three principal diffusivity directions in a voxel. Together with the eigenvalues they contain all information of the original tensor.

2.1.1 Scalar indices

A scalar index is a measure that classifies the diffusion tensor using the relations between the eigenvalues [25]. By applying a scalar index, a DTI dataset can be simplified to a scalar dataset. Although a scalar cannot represent all the information of the tensor, a scalar dataset is often more easy to interpret and visualize than a complex DTI dataset. Westin et al. [25] present indices that distinguish between three categories of diffusion: linear anisotropy, planar anisotropy and isotropy.

Linear anisotropy (*Cl*) is diffusion mainly in one direction; the eigenvalue of the main eigenvector is much larger than the other two eigenvalues ($\lambda_1 > \lambda_2 = \lambda_3$) and can be visualized with a cigar shape (see figure 2.2a). It is defined as:

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

Planar anisotropy (*Cp*) is diffusion restricted to a plane defined by the two eigenvectors corresponding to the two largest eigenvalues ($\lambda_1 = \lambda_2 > \lambda_3$) and can be thought of as a pancake shape (see figure 2.2b). It is defined as:

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

Isotropy (*Cs*) indicates diffusion in all directions ($\lambda_1 = \lambda_2 = \lambda_3$); this is best visualized with a spherical shape (see figure 2.2c). It is defined as:

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

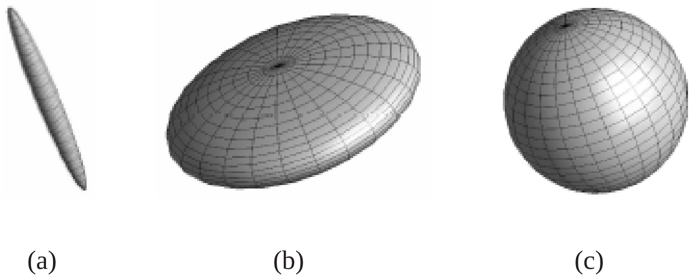

(a) (b) (c)

Figure 2.2: Classification of diffusion [2].

Another often used index is called Fractional Anisotropy (FA)[1]. FA distinguishes between isotropic and anisotropic diffusion, but not between linear and planar diffusion. It is defined as:

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

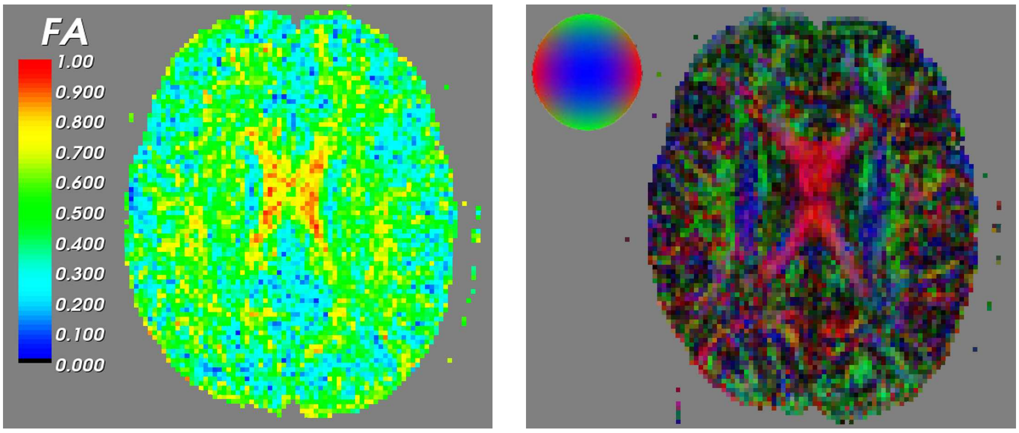
In isotropic tissue FA = 0 and in anisotropic tissue FA = 1.

2.2 Visualization

Visualization of DTI data is difficult because of the high dimensionality of the information. Diffusion is represented by a 3×3 symmetric tensor, which means that each voxel contains 6 scalar values. Creating a DTI visualization that is both orderly as well as detailed is a complex task and the topic of ongoing research. Some visualization methods show the complete tensor, but only in a small area, where they provide very detailed local information. Glyphing is an example of such a method. Other visualization methods, for instance fiber tracking, simplify the tensor field to a vector field, thereby making it easier to display the data throughout the whole volume and provide global information to some extent.

2.2.1 Color-coding

Color-coding is a 2D visualization technique in which voxels are assigned a color according to some local characteristic of the tensor. An example of such a characteristic is the type of diffusion in a particular voxel, which can be measured by a scalar index like FA. Figure 2.3a shows a slice of DTI data that is color-coded by mapping the FA index of each voxel to a color using a look-up-table.



(a)

(b)

Figure 2.3: Color-coded slices of DTI data.

Another characteristic that can be used for color-coding is the direction of the main eigenvector. In this case, different colors are assigned to the principal directions of the main eigenvector. The X, Y and Z direction of the main eigenvector correspond to the primary R, G, B color channels. Red voxels indicate diffusion mainly in the left-right direction, blue voxels in the bottom-top direction and green voxels in the front-back direction. Because the sign of the eigenvector is not defined opposing directions have the same color. In figure 2.3b a slice of DTI data is color-coded using the main eigenvector and then weighted with the FA index. Voxels with a high FA (anistropic tissue) get a high intensity, and voxels with a low FA (isotropic tissue) get a low intensity. Both visualizations were created with the DTI Tool [2].

2.2.2 Glyphing

A glyph is a geometric object which size and orientation are defined by the tensor. The orientation of the glyph is determined by the main eigenvector and its size by the eigenvalues. Glyphs can be basic shapes like boxes and ellipsoids or more complex shapes such as superquadric tensor glyphs [19]. Glyphs can be used in 3D as well as 2D visualizations, but because of occlusion and the amount of information glyphs convey they are mostly used in small 2D regions. Figure 2.4 shows two kinds of glyphs which are color-coded using the FA index mapped to a hue lookup-table. This visualization was created with the DTI Tool [2].

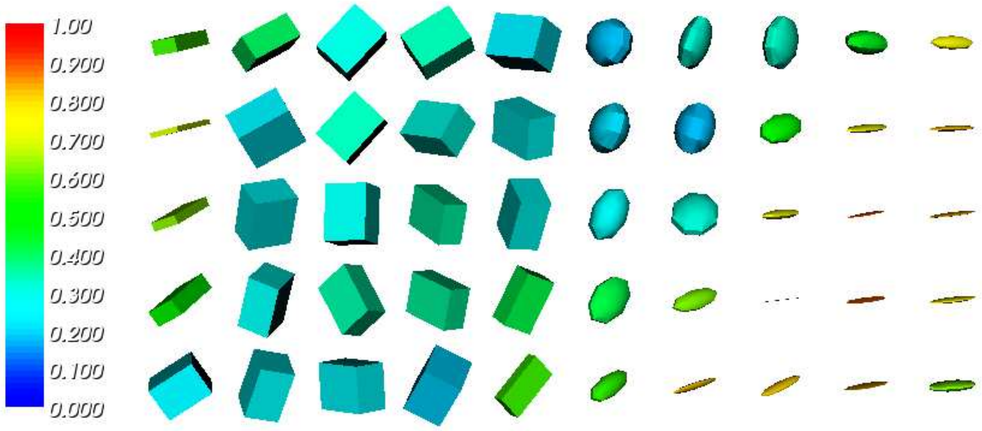


Figure 2.4: Glyphs: boxes (left) and ellipsoids (right).

2.2.3 Direct volume rendering and texture based methods

Direct volume rendering is a visualization technique in which no intermediate geometry is created. Instead, transfer functions are used that map certain properties of the tensor field to visual properties like color, opacity and shading. A challenge with volume rendering is to define these transfer functions; some features are of more interest than others, and these must not be concealed by less important structures. A problem with volume rendering is that it is computational expensive, and user interaction is also limited (see figure 2.5).

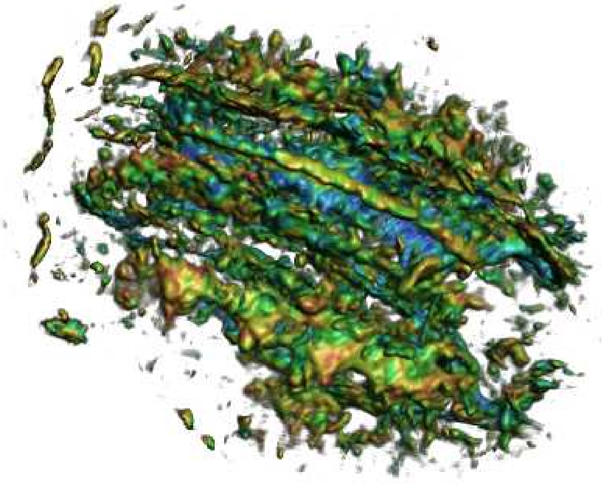


Figure 2.5: Volume rendering [20].

2.2.4 Other tensor visualization techniques

Other visualization techniques that can be applied to tensor fields are volume deformation, geodesics and topology visualization. Volume deformation [30] considers the tensor field to be a force field that deforms an object placed in it. Geodesic surfaces [11] show the effect of the tensor field as a deformation of flat space. And finally, in a topology based method [31] a skeleton is created by extracting certain specific features from the tensor field.

2.3 Fiber tracking

At the moment, one of the most promising DTI visualization techniques is fiber tracking. The goal of fiber tracking is to reconstruct continuous 3D trajectories from the discrete DTI data. There are two types of fiber tracking algorithms: line propagation and energy minimization [22]. Line propagation works by assuming that the main diffusion direction in a voxel is aligned with the orientation of the white matter tracts. From a starting point a line is propagated through the volume in the direction of the main diffusion. Energy minimization techniques, on the other hand, search for the energetically most favorable path between points. In this thesis only fibers created with the line propagation method are considered.

2.3.1 Algorithm

The fiber tracking algorithm is initialized by the placement of seed points. The placement of these seed points can be done in one of two ways:

- Seeding is done in a region of interest (ROI) which is defined by the user. There are two kinds of ROI's: seed-ROI's and through-ROI's. Seed-ROI's are regions in which seeds are placed at a regular distance from each other. Through-ROI's do not contain seeds, but are regions that fibers have to pass through to be included in the final visualization. What fibers are created depends heavily on the placement of the ROI's: important fiber tracts may be missed if a region is incorrectly placed. Also, the anatomy of a patient may differ from the anatomy of a healthy subject and this makes the placement of the ROI difficult. Multiple ROI's can be used to find complex structures.

- Seeding is done throughout the whole volume. This reconstructs fibers in the complete volume and is therefore computationally very expensive. The number of fibers can be very large depending on the size of the data set, the distance between seeds and the stopping criterion. The advantage is that structures are not missed due to the wrong placement of the ROI. However, it is very difficult to find specific structures because of occlusion and limited user interaction.

Starting from a seed point, a fiber is not only traced in the direction of the main eigenvector, but also in the opposite direction. This is because the sign of the eigenvector is undefined; it can be positive or negative.

Fiber tracking is usually done in a continuous vector field. DTI data however, is measured on a regular discrete 3D grid. In order to get a continuous vector field, the eigenvectors are interpolated or calculated from an interpolated tensor field.

Fiber tracking is stopped in areas with low anisotropy. Low anisotropy means that the main diffusion direction is poorly defined, very sensitive to noise and therefore not reliable anymore. Another reason to stop tracking is when the angle between two steps becomes too big, because it is assumed that fibers from anatomical structures are smooth most of the time. And finally, fiber tracking is discontinued if fibers go beyond the boundaries of the volume.

2.3.2 Problems

A major challenge with DTI data is noise. Noise causes fibers to be broken or leads to erroneous pathways. At the moment it is not exactly known what anatomical structures can be found in DTI data and it is therefore very difficult to identify noise. Only when it causes a major artifact or distorts a large well-known structure, like the corpus callosum, it can be clearly identified. However, this requires prior medical knowledge about how specific structures look like, which is not always available; identification and validation of fiber tracts is an active research area.

Another problem with fiber tracking is a phenomenon called partial volume effect. Due to the limited resolution of DTI datasets certain voxels contain information about more than one fiber bundle. This causes trouble because the fiber tracking algorithm assumes that each voxel contains only one main fiber direction. In areas where planar anisotropy is high this assumption does not hold anymore. Places at which fibers cross, kiss, converge or diverge have planar anisotropy (see figure 2.6). In these voxels, diffusion is high in more than one direction and it is unclear which direction should be followed. The fiber tracking algorithm simply stops in these ambiguous areas, which results in broken fibers. Instead of stopping, another option is to generate a surface in areas with high planar anisotropy [2].

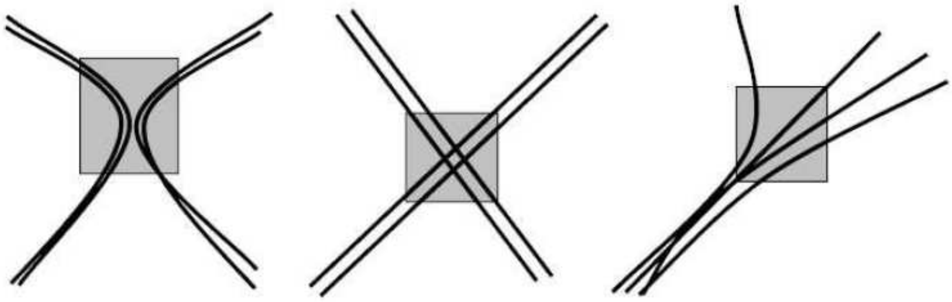


Figure 2.6: Ambiguous areas: kissing fibers (left), crossing fibers (middle) and converging/diverging fibers (right) [2]

A related problem occurs when two areas corresponding to different fiber bundles are poorly separated. Fiber tracking initiated in one area often continues in the other area, resulting in fiber tracts that are ”glued” together. That is, these fibers consist of two parts which belong to different anatomical structures. This problem can be partially solved by changing the stopping criterion: setting a higher minimum anisotropy reduces the number of fibers that are glued together, but increases the number of broken fibers. Another way to solve this is by using AND-regions:

erroneous fiber tracts are filtered out by specifying regions through which the fibers must pass. However, this is not possible with seeding throughout the whole volume.

Chapter 3

Fiber Clustering

The fiber tracking algorithm described in the last chapter produces a set of fibers (see figure 3.1). This chapter reviews the methods available in literature for clustering the fibers into meaningful groups. After the introduction the two essential components of the clustering process are described: the proximity measure and the clustering algorithm. Finally, the postprocessing of clusters is discussed.

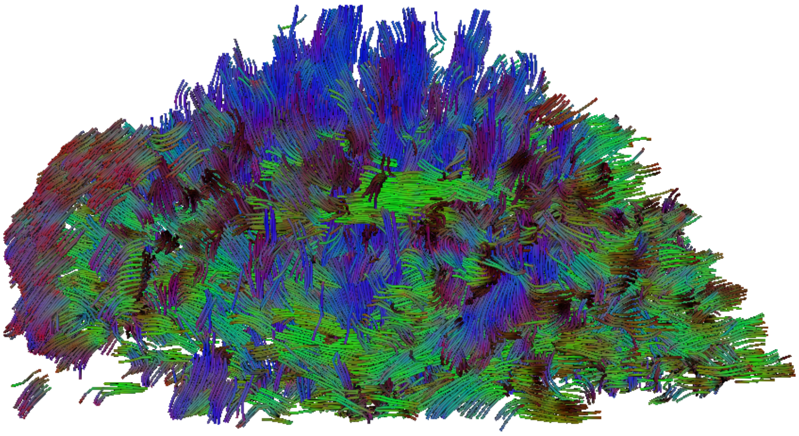


Figure 3.1: Fibers of a human brain created by a fiber tracking algorithm with seeding throughout the whole volume. This visualization was created with the DTI Tool [23].

3.1 Overview

Clustering is the classification of a set of objects into groups that have meaning in the context of a specific problem [17]. The problem in this case is to partition the fibers in such a way that the clusters correspond to the anatomical fiber bundles present in the human brain. Section 3.2 describes the main anatomical characteristics of fiber bundles.

Figure 3.2 depicts the steps that are involved in the visualization of fiber clusters. It shows how the data flows through the system; from the original tensor field to the final fiber clusters. As can be seen, the fiber clustering process gets its input from the fiber tracking algorithm. The performance of the fiber clustering is therefore directly dependent on the quality of the fibers produced by the fiber tracking algorithm [5].

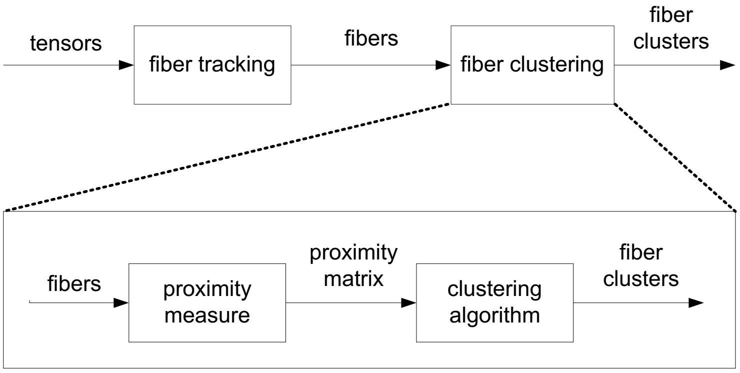


Figure 3.2: Fiber clustering in the visualization process

The steps involved in the fiber clustering process are (see figure 3.2 bottom):

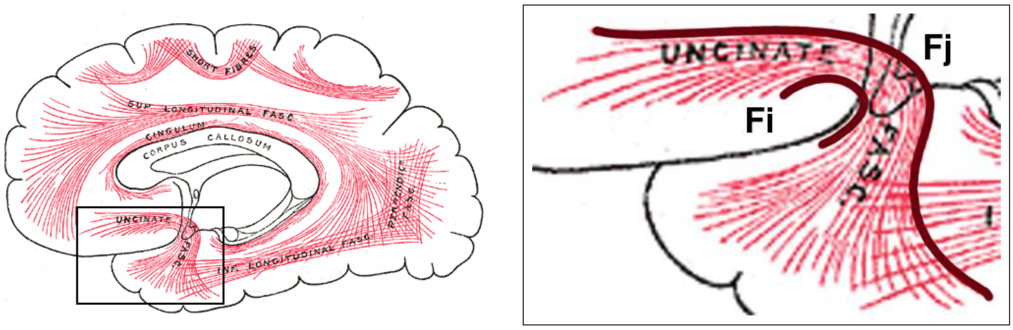
Proximity measure. This is a function that computes the (dis)similarity between pairs of fibers. Section 3.3 gives an overview of the proximity measures that can be found in literature.

Clustering algorithm. The proximity function is used by the clustering algorithm to produce a partition of the set of fibers. Clustering algorithms come in various forms; some produce a single partition, others create a hierarchy of partitions. Section 3.4 reviews the clustering methods that have been used to cluster fibers. Also, a clustering method is presented that has not yet been used for fiber clustering.

Once the clusters are acquired, they can be quantified and visualized. This includes fiber coloring, shape analysis and surface rendering. Section 3.5 describes the techniques that are used to postprocess fiber clusters.

3.2 Characteristics of fiber bundles

The function of white matter tracts (fiber bundles) is to interconnect regions of grey matter in the brain. Figure 3.3 shows a schematic picture of the brain in which a number of fiber bundles are depicted.



(a) Global view

(b) Detailed view

Figure 3.3: Schematic picture of the brain. Adapted from Brun et al. [6] and Gray [14].

As can be seen in figure 3.3a, fiber bundles come in various shapes and sizes. Some bundles consist of a relatively small number of long fibers which form a kind of tube structure. Other bundles consist of a large number of smaller fibers which form a thin surface.

Figure 3.3b shows a closeup of a fiber bundle. A number of observations can be made about the relationship between fibers:

- A pair of fibers from the same bundle that are direct neighbors of each other, are separated by a small distance and have a similar shape.
- A pair of fibers from the same bundle that are *not* direct neighbors, can have a considerable distance between them, and can have quite different shapes. However, between any two dissimilar and distant fibers from the same bundle, there are other fibers in between that cover the distance and

change of shape. That is, there is a smooth transition between any two fibers from the same bundle.

Here is an example to illustrate these observations. Consider the two emphasized fibers F_i and F_j in figure 3.3b. As far as shape is concerned, they represent the two fibers from this particular bundle that are the least similar. But although F_i and F_j are very different they are surrounded by fibers that are quite similar in shape. In other words, there is a gradual change of shape between any two fibers from the same bundle.

Also important to keep in mind is that the fiber bundles depicted in figure 3.3 are idealized versions of the bundles that are typically found in DTI scans of actual human brains. A limited resolution, noise and other problems might cause the absence of certain parts of bundles, or the presence of erroneous pathways (see section 2.3.2).

3.3 Proximity measures

A clustering method groups items together that are similar in some way, and thus needs a way to measure similarity between objects. A proximity measure computes either similarity or dissimilarity between a pair of objects. The more equal two objects are, the larger a similarity measure and the smaller a dissimilarity measure. For instance, the Euclidean distance between two points in space is a dissimilarity measure, while a correlation coefficient is an example of a similarity measure [17].

In this thesis all proximity measures are symmetric: the proximity between fibers F_i and F_j is the same as the proximity between fibers F_j and F_i. Also, a fiber has the same degree of proximity with itself.

There is no standard way to compute the proximity between a pair of fibers. Computing the proximity between a pair of *points* is relatively easy: the Euclidean distance gives a good indication of dissimilarity. However, a fiber is represented by an ordered list of points and for such a high dimensional object the definition of proximity is less obvious. This section reviews some of the proximity measures that can be found in literature.

In the following equations, $\| \, . \, \|$ is the Euclidean norm.

3.3.1 Closest point distance

A proximity measure that provides only very coarse information about the dissimilarity of a pair of fibers is the closest point distance [8]. The closest point distance d_c is defined as the minimum distance between points p_k and p_l, where p_k is a point on fiber F_i and p_l is a point on fiber F_j:

$$d_c(F_i, F_j) = \min_{p_k \in F_i, p_l \in F_j} \parallel p_k - p_l \parallel .$$

The closest point distance is not able to differentiate between fibers from different bundles if they cross, kiss, converge or diverge. In all these cases this measure underestimates the distance.

3.3.2 Mean of closest point distances

A distance measure that provides more global information about the dissimilarity of a fiber pair is the mean of closest point distances [8]. Each point on one fiber is mapped to the closest point on the other fiber, thus forming point pairs. The fiber distance is defined as the mean of these closest point pair distances:

$$d_M(F_i, F_j) = \text{mean}(d_m(F_i, F_j), d_m(F_j, F_i))$$

with

$$d_m(F_i, F_j) = \operatorname*{mean}_{p_l \in F_i} \min_{p_k \in F_j} \parallel p_k - p_l \parallel .$$

This measure has the potential to give an accurate indication of distance between fibers. A problem might be if two fibers from the same bundle have widely different lengths, for example due to limitations of the fiber tracking algorithm. This could cause an overestimation of the distance.

3.3.3 Hausdorff distance

The Hausdorff distance is very conservative: two fibers are considered similar only if all distances between closest point pairs are small [8]. The Hausdorff distance is defined as the maximum distance between two closest point pairs:

$$d_H(F_i, F_j) = \max(d_h(F_i, F_j), d_h(F_j, F_i))$$

with

$$d_h(F_i, F_j) = \max_{p_k \in F_i} \min_{p_l \in F_j} \parallel p_k - p_l \parallel .$$

This measure has the tendency to overestimate the distance between fibers. For various reasons fibers from the same bundle may be of different length, or might not run close for the entire length, and in these cases the maximum distance between two closest point pairs might be fairly large.

3.3.4 End points distance

Brun et al. [6] consider fibers that have close endpoints as similar. The reasoning behind this is that fibers from the same anatomical structure connect the same areas of the brain. Except for the positions of the endpoints all other information regarding the fibers is discarded.

Similarity between fibers i and j is defined as:

$$\begin{aligned} f_i &= (f_{i,1}, f_{i,end}), \\ \tilde{f}_i &= (f_{i,end}, f_{i,1}), \\ S_E(i, j) &= \exp\left(-\frac{\parallel f_i - f_j \parallel^2}{2\sigma^2}\right) + \exp\left(-\frac{\parallel f_i - \tilde{f}_j \parallel^2}{2\sigma^2}\right). \end{aligned}$$

In this equation, $f_{i,1}$ and $f_{i,end}$ are the first and last coordinates of fiber i.

For this similarity measure we propose an alternative definition which measures distance and does not have any additional parameters:

$$\begin{aligned} d_1 &= \parallel f_{i,1} - f_{j,1} \parallel + \parallel f_{i,end} - f_{j,end} \parallel \\ d_2 &= \parallel f_{i,1} - f_{j,end} \parallel + \parallel f_{i,end} - f_{j,1} \parallel \\ D_E(i, j) &= \min(d_1, d_2). \end{aligned}$$

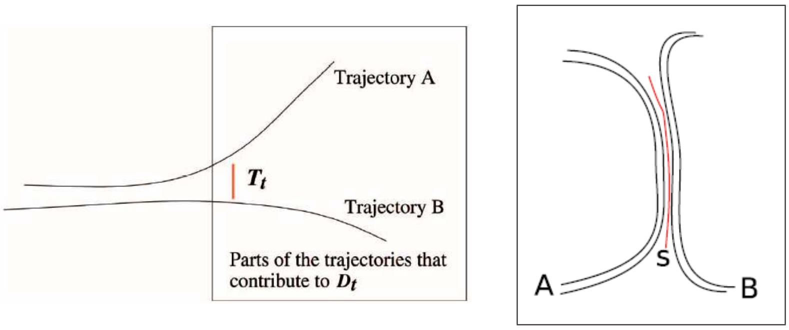
These measures could have problems when fibers are damaged or when fibers from different bundles start and end in approximately the same region. This might happen when fibers from different bundles are ”glued” together (see section 2.3.2).

3.3.5 Distance above threshold

Zhang and Laidlaw [27] define the dissimilarity between two fibers as follows:

$$D_t = \frac{\int_s \max(\mathrm{dist}(s) - T_t, 0)\mathrm{d}s}{\int_s \max\left(\frac{\mathrm{dist}(s)-T_t}{|\mathrm{dist}(s)-T_t|}, 0\right)\mathrm{d}s}.$$

In this equation, s parameterizes the arc length of the shorter fiber, and dist(s) is the shortest (Euclidean) distance from location s of the shorter fiber to the longer fiber. T_t is a threshold, and only distances above this threshold contribute to the distance (see figure 3.4a).



(a) Threshold T_t[26].

(b) Potential problem case [28].

Figure 3.4: Distance above threshold

Fibers do not need to be of comparable length to be considered similar by this measure. This works in favor of damaged fibers which are often much shorter than undamaged fibers from the same bundle.

This distance measure works less well if two fibers with different lengths actually belong to different bundles (see figure 3.4b). In this figure, the shorter fiber s is considered to be close to both fiber A as well as fiber B, although fibers A and B are not similar at all. Because fiber s acts as a bridge, all three fibers might end up in the same cluster.

3.3.6 Corresponding segment distance

Ding et al. [9] define similarity by establishing a corresponding segment between pairs of fibers. A corresponding segment can be thought of as the portion of a fiber that has a point-wise correspondence to a portion of another fiber (see figure 3.5). The more the fibers overlap, the more similar they are. A seed plane (also called region of interest, see section 2.3.1) is used to determine a corresponding point on both fibers. From this (seed) point the corresponding segment can be found by searching the shorter end along both directions.

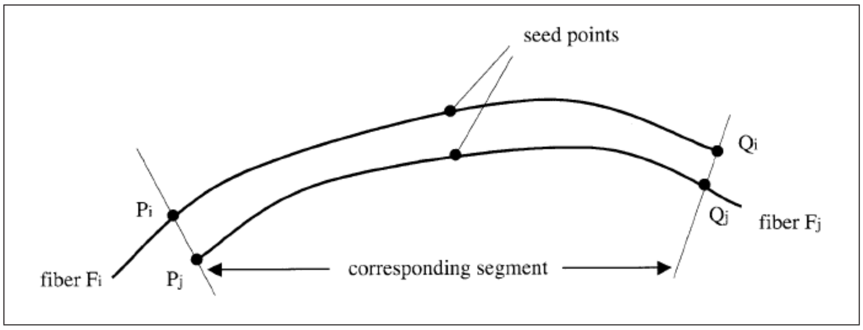


Figure 3.5: Definition of a corresponding segment. In this figure, portion $P_i Q_i$ of fiber F_i is the corresponding segment to portion $P_j Q_j$ of fiber F_j [10].

First, a corresponding ratio R_{cs} between a pair of fibers is defined:

$$R_{cs} = \frac{L_{cs}}{L_i + L_j - L_{cs}}.$$

In this equation, L_{cs} is the length of the corresponding segment, L_i and L_j are the length of F_i and F_j respectively. This ratio is 0 if fibers have no overlap at all, and is 1 if they overlap completely.

Then, the similarity S_{CS} between a pair of fibers F_i and F_j is defined as:

$$S_{CS}(F_i, F_j) = R_{cs} \cdot \exp(-D/C).$$

In this equation, R_{cs} is the corresponding segment ratio, D is the mean Euclidean distance between corresponding segments, and C is a coefficient for D. If F_i and F_j are identical then S_{CS} is 1, and it decreases either if the corresponding segment ratio decreases, or if the mean distance increases. Coefficient C is used to weigh the influence of the corresponding ratio and the mean distance; the larger C is, the less influence D has on the similarity measure. Ding et al. [9] use a value of 1.0 for C.

This measure uses a seed (ROI) plane to define similarity and it is not directly obvious how to establish a corresponding segment without the use of such a seed plane. This is a problem in situations in which fibers have been created with the all volume seeding approach, in which case a seed plane does not exist.

3.3.7 Mapping to an Euclidean feature space

Brun et al. [5] map the high dimensional fibers to a relatively low dimensional Euclidean feature space and use a Gaussian kernel to compare the fibers in this new space.

First, each fiber is mapped to a 9-dimensional Euclidean feature space. This mapping maintains some but not all of the information about fiber shape and position. From the points of a fiber the mean vector m and the covariance matrix C is calculated. Furthermore, the square root of the covariance matrix, $G = \sqrt{C}$ is taken to avoid non-linear scaling behavior. A fiber can now be described as:

$$\Phi(F) = (m_x, m_y, m_z, g_{xx}, g_{xy}, g_{xz}, g_{yy}, g_{yz}, g_{zz})^T .$$

Then, the similarity between a pair of fibers F_i and F_j can be calculated using a Gaussian kernel:

$$S_K(\Phi(F_i), \Phi(F_j)) = \exp\left(-\frac{\| \Phi(F_i) - \Phi(F_j) \|^2}{2\sigma^2}\right) .$$

The parameter σ adjusts the sensitivity of the similarity function. Similar fibers are mapped to unity, while dissimilar fibers are mapped to values close to 0.

3.4 Clustering methods

The proximity measures defined in the last section are used to establish a relationship between fibers. The proximities are compiled in the proximity matrix, in which rows and columns correspond to fibers. The proximity matrix is the input to a clustering algorithm [17].

A clustering algorithm imposes a type of classification on the input data. This classification can take various forms. A *partitional* clustering algorithm produces a single partition of the input, while a *hierarchical* clustering algorithm creates a nested hierarchy of partitions. A *hard* clustering algorithm produces an exclusive partition, in which each object belongs to exactly one cluster, while a *fuzzy* clustering algorithm creates a nonexclusive classification, in which each object has a certain degree of membership to each cluster [17].

The following section reviews the clustering methods that have been used by other research groups for clustering fibers. After that, an additional clustering method called shared nearest neighbor clustering is presented which has not yet been used for fiber clustering.

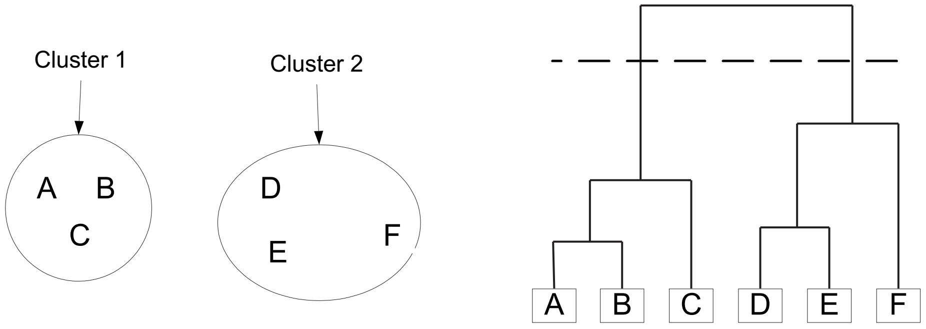
3.4.1 Hierarchical clustering

Zhang and Laidlaw [27] use a hierarchical clustering method to cluster fibers. A hierarchical clustering method transforms a proximity matrix into a sequence of nested partitions [17]. An *agglomerative* hierarchical clustering method works as follows:

1. Put each item into an individual cluster.
2. Merge the two most similar clusters.
3. Repeat step 2 until there is only a single cluster left.

A *divisive* method works the other way around: it starts with a single cluster containing all the items, and at each stage splits one cluster until every item is in a singleton cluster.

Based on the way similarity between clusters is defined, several variations of the agglomerative hierarchical clustering method can be devised. The two most basic variations are single-link and complete-link [18].



(a)

(b)

Figure 3.6: On the left is a clustering consisting of two clusters. On the right is the dendrogram resulting from the single-link method. To obtain the segmentation on the left the dendrogram is cut at the level indicated by the the dotted line.

In the single-link algorithm, the distance between two clusters is the distance between the closest pair of items (one item from the first cluster, the other item from the second cluster). The single-link method works well for elongated and well separated clusters and it can find clusters of different sizes and complex shapes.

It performs poorly on data containing noise, because noise may act as a bridge between two otherwise separated clusters. This is known as the chaining effect [18].

In the complete-link algorithm, the distance between clusters is defined as the maximum distance between a pair of items (one item from either cluster). This tends to produce compact, more tightly bound clusters. The complete-link algorithm is less versatile than the single-link algorithm because it is unable to find clusters of varying sizes or complex shapes [18].

In the weighted-average algorithm, the distance between clusters is defined as the average of the minimum and the maximum distance between a pair of items from the different clusters.

The result of a hierarchical clustering method is a special tree structure called dendrogram. A dendrogram shows the nested clustering of items and the item distances at which clusters change. By cutting the dendrogram at a certain level a partition of the data is obtained (see figure 3.6).

Both the single-link and the complete-link method are used by Zhang and Laidlaw [27], although in subsequent papers [28, 29] they abandon the use of the complete-link method. The weighted-average method has not yet been used in the context of fiber clustering.

3.4.2 Partitional clustering

In contrast to hierarchical clustering methods, partitional clustering methods only produce a single partition of the data.

Corouge et al. [8] use a partitional clustering method that propagates cluster labels from fiber to neighboring fiber. It assigns each unlabeled fiber to the cluster of its closest neighbor, provided that the distance to this closest neighbor is below a threshold. A partition of the data with a specific number of clusters can be acquired by setting a threshold; a low threshold gives many clusters, whereas a high threshold results in a reduced number of clusters.

Ding et al. [9] propose a clustering method based on the K-most-similar neighbors method. A fiber F is grouped with up to k of its closest neighbors, provided that the distance to a neighbor is below a threshold. The neighbors of a fiber F are those (eight) fibers whose seedpoints are the neighbors of the seedpoint of F. This process is repeated for each fiber. At the end, the connected components form the clusters. This method assumes the presence of a seedplane, which is only the case for ROI fiber tracking. The parameters that have to be set are the threshold and the number of neighbors to consider. A high threshold prevents the grouping of fibers from different anatomical structures, whereas the number of neighbors k determines the compactness of the clusters.

3.4.3 Graph theoretic clustering

In graph theoretic clustering the items to be clustered are the nodes of an undirected graph and the edges represent the relationship between the nodes. The relationship can be based on similarity or dissimilarity depending on the algorithm.

Brun et al. [6] use a spectral embedding technique called Laplacian eigenmaps for clustering fibers. First, a sparse graph is created in which each fiber is a node and edges exist between nodes of neighboring fibers. Each edge receives a weight based on the distance between fibers; the larger the distance between fibers the smaller the weight. The structure of this graph can be mapped to a low dimensional Euclidean space by solving an eigenvector problem. Data points that are close to each other in the original space are mapped to nearby points in the new Euclidean space. Once the fibers are reduced to points in the low dimensional Euclidean space, they can be mapped to a continuous RGB color space. This way similar fibers are assigned similar colors.

In another paper by Brun et al. [5], a clustering method based on normalized cuts is used to group fibers. To start with, an undirected graph is created in which nodes correspond to fibers, and each edge is assigned a weight that represents the similarity between fibers. Most edges are expected to have a weight close to 0 (dissimilar) so the graph can be considered sparse. To partition the nodes into two disjoint groups the graph is cut. A normalized cut tries to minimize the cut between the two partitions and penalizes partitions in which some nodes are only loosely connected to the complete graph. A clustering can be achieved by cutting the graph repeatedly until the desired number of clusters are found or if the weights crossing the cut are above a certain threshold. The connected components of the graph define the clusters.

3.4.4 Fuzzy clustering

Shimony et al. [16] employ a fuzzy c-means algorithm. Fuzzy clustering methods do not produce a hard clustering of the data [18]. Instead, each item is associated with a cluster by a membership function that takes values between 0 and 1. A larger value of the membership function indicates a higher confidence that the item belongs to the cluster. The result of a fuzzy clustering can be converted to a hard clustering by thresholding the value of the membership function.

3.4.5 Shared nearest neighbor clustering

Shared nearest neighbor clustering [12] is a clustering algorithm that has not yet been used for fiber clustering. We want to use the shared nearest neighbor algorithm because it has a number of beneficial characteristics in the context of fiber clustering. In particular, it can find clusters of different sizes and shapes in data that contains noise and outliers. These characteristics are beneficial because the anatomical fiber bundles are also of different sizes and shapes, and DTI data is often very noisy.

The shared nearest neighbor algorithm is based on the notion that two data points that share a lot of neighbors probably belong to the same cluster. In other words, ”the similarity between two points is confirmed by their common (shared) neighbors” [12]. The algorithm works as follows:

1. A k nearest neighbor graph is constructed from the proximity matrix. In this graph, each data point corresponds to a node which is connected to the nodes of the k nearest neighbors of that data point.

2. A shared nearest neighbor graph is constructed from the k nearest neighbor graph. In a shared nearest neighbor graph, edges exist only between data points that have each other in their nearest neighbor lists. That is, if point p is one of the k closest neighbors of point q, and q is also one of the k closest neighbors of point p, then an edge exists between p and q. The weight of this edge is computed as follows:

   $$strength(p,q) = \sum (k+1-m)(k+1-n), \text{ where } i_m = j_n.$$

   In this equation, m and n are the positions of a shared neighbor in p and q's nearest neighbor lists. Thus, a ”close” shared neighbor is found to be more important than a ”far” shared neighbor. In general, a higher value for k increases the number shared neighbors and this in turn leads to higher weights between data points.

3. Clusters are obtained by removing all edges from the shared nearest neighbor graph that have a weight below a certain threshold. In general, a low edge threshold results in few clusters, because most connections are preserved. A high threshold results in a lot of clusters, because most connections are broken. Which value for the edge threshold is considered ”low” or ”high” depends on the value of k.

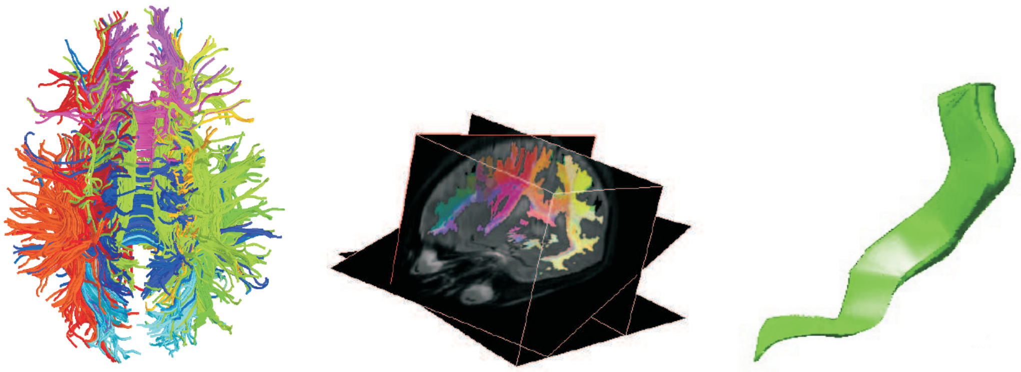
The parameters of the shared nearest neighbor algorithm are the size of the nearest neighbor list k and the edge threshold. Notice that the number of clusters is not a parameter of the algorithm. ”Depending on the nature of the data, the algorithm finds the natural number of clusters for the given set of parameters.” [12] Which parameter settings are appropriate in the context of fiber clustering remains unclear.

A question that is left unanswered for the moment is if the shared nearest neighbor clustering algorithm improves on any of the fiber clustering algorithms that have been discussed in this section. This question is answered in chapter 5, in which we apply the shared nearest neighbor clustering algorithm to actual DTI data and validate the results using the validation procedure described in chapter 4.

3.5 Postprocessing of fiber clusters

Once the fibers are clustered into coherent groups, they can be visualized and analyzed. This includes:

- Coloring the fibers according to which cluster they belong (see figure 3.7a).
- Segmentation of voxel space (see figure 3.7b).
- Quantification of bundle properties [9, 8]. Measuring the properties of a group of fibers might be of more interest than the properties of individual fibers. Information that can be derived from the bundles include geometric properties like curvature and torsion, and physical parameters like mean longitudinal and perpendicular diffusivity.
- Rendering of a bundle surface. Ding et al. [9] generate a surface by identifying a number of cross-sectional planes that are perpendicular to the average direction of the fibers in that bundle. In each plane a closed contour of the bundle is acquired by taking the convex hull of all the points at which fibers intersect the cross-sectional plane. The contour is interpolated and triangulated between neighboring cross-sectional planes. Flat shading is used to render the surface (see figure 3.7c).
- Matching of clusters across subjects [29]. Here, the goal is to identify clusters corresponding to anatomical bundles that can be reliably found across multiple datasets.



(a) Fiber coloring [5]. (b) Voxel coloring [5]. (c) Bundle surface[9].

Figure 3.7: Visualization techniques for fiber bundles.

Chapter 4

Validation

The distance measures and clustering methods described in the previous chapter can be used to partition a set of fibers. Figure 4.1 shows two different clusterings of the same set of fibers. This chapter describes various techniques for validating these cluster results. Our approach is based on the creation of a gold standard to which the cluster results are compared. Several comparison methods are examined and a suitable new method is developed with the help of physicians from the Maxima Medical Center.

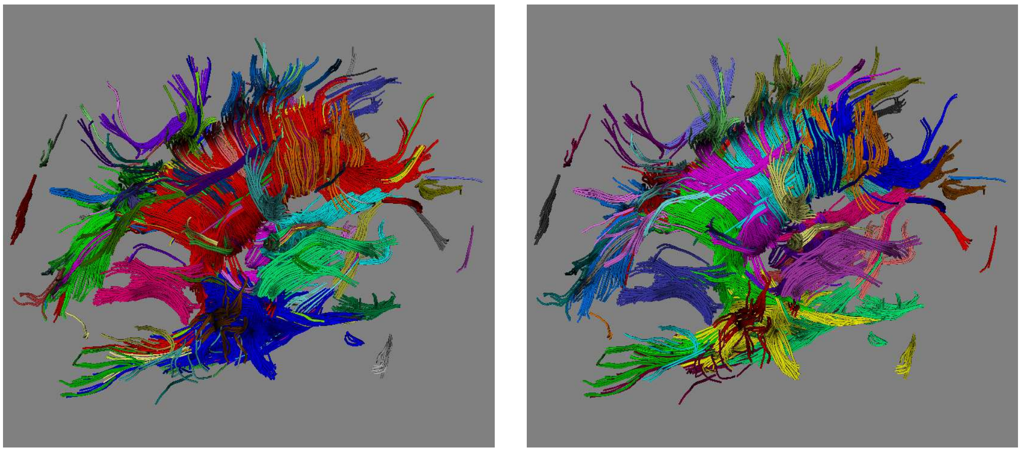


Figure 4.1: Two different clusterings of the same set of fibers.

4.1 Overview

Figure 4.2 shows the steps that are involved in the validation process. The fibers created with the fiber tracking algorithm are clustered using one of the proximity measures and one of the clustering algorithms described in chapter 3. The result is a partitioning of the fibers into a number of clusters. Each combination of proximity measure and clustering method produces a different clustering. The basic question here is what distance measure and what clustering method produce the clustering that is closest to the optimal clustering.

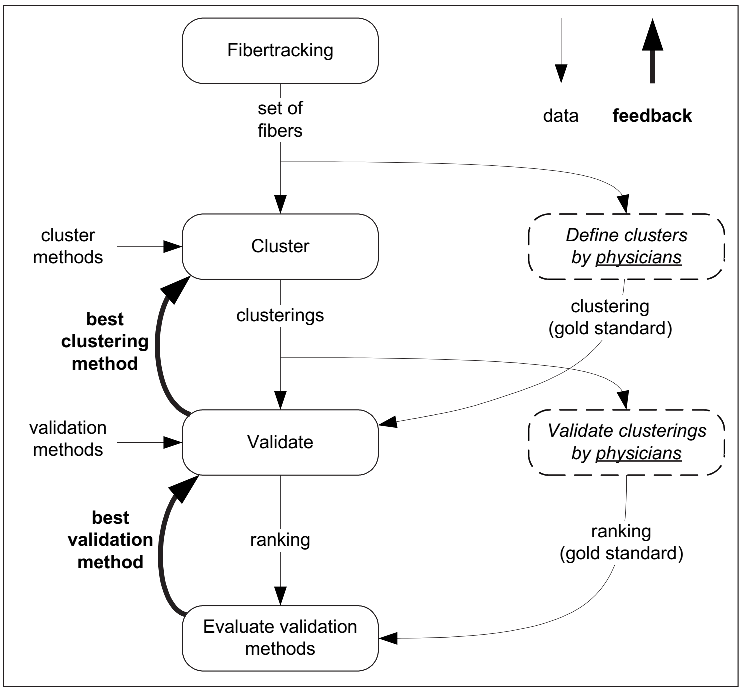


Figure 4.2: Overview of the validation process.

The first step in the validation process involves the creation of a gold standard, which is considered our optimal clustering. This is done by manually classifying the fibers into a number of bundles. The classification process is described in section 4.2.

Once a gold standard is established, a validation method is chosen to determine the agreement between the gold standard and the various cluster results. There are a number of validation methods available in literature. Section 4.3 examines Receiver Operator Characteristic (ROC) curves and external indices. Because these methods use different criteria for validation, they also give different results. In the context of fiber clustering, the goal is to find a validation method that meets the criteria of physicians. For this purpose, we propose several adjustments to the validation methods available in literature.

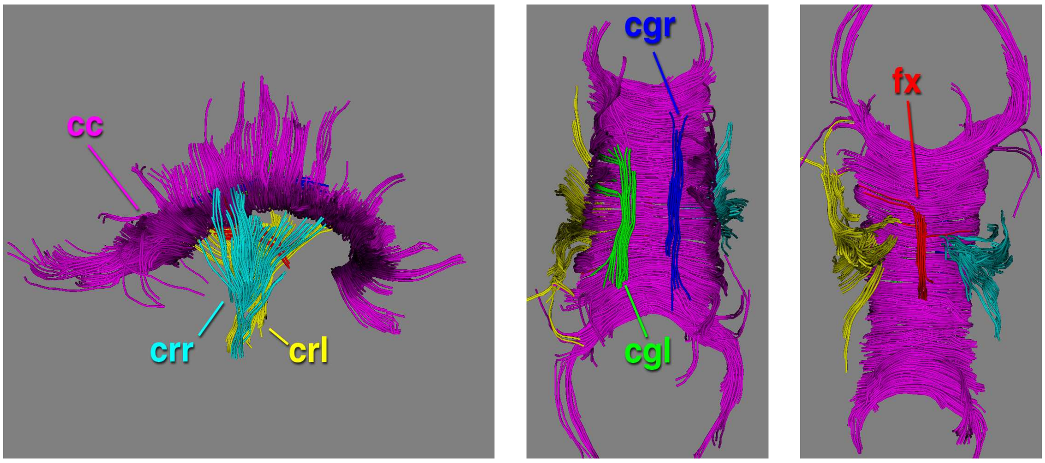
The various validation methods are evaluated in section 4.4. This is done by letting physicians create a ranking of a number of clusterings. This ranking is then used as a gold standard to which the rankings created by the various validation methods are compared. The validation method that produces the ranking that has the highest correlation with the ranking of the physicians is considered the best validation method. This method is used in the next chapter to pick the best clustering method.

4.2 Classification

The first step of the validation process is to establish a gold standard to which the cluster results can be compared. For our purposes, the gold standard is a manually defined classification of a set of fibers. The fibers are classified into a number of anatomical structures, called bundles, for which is known that they can be reliably identified using the fiber tracking technique. Ideally, the classification is done by physicians. However, for this study we did the classification ourselves, and it was verified by physicians from the MMC.

Our gold standard includes the following bundles: the corpus callosum, the fornix, the cingulum (both hemispheres) and the corona radiata (both hemispheres). These anatomical structures are identified in a number of studies [7, 13, 24] and can be reconstructed with the fiber tracking technique.

Figure 4.3 shows the result of a classification performed on an actual DTI data set of a healthy subject. Only fibers belonging to the gold standard are shown.



(a) Side view.

(b) Top view.

(c) Bottom view.

Figure 4.3: Three views of a classification of a DTI data set. Colors are used to distinguish between the different bundles. The meaning of the colors and abbreviations is given in table 4.1.

Bundle		Color	Number of fibers
cc	Corpus callosum	purple	716
crl	Corona radiata (left hemisphere)	yellow	110
crr	Corona radiata (right hemisphere)	light blue	69
cgl	Cingulum (left hemisphere)	green	23
cgr	Cingulum (right hemisphere)	blue	11
fx	Fornix	red	11
	Unclassified *(not shown)*		2655
		Total	3595

Table 4.1: Anatomical structures of the manual classification.

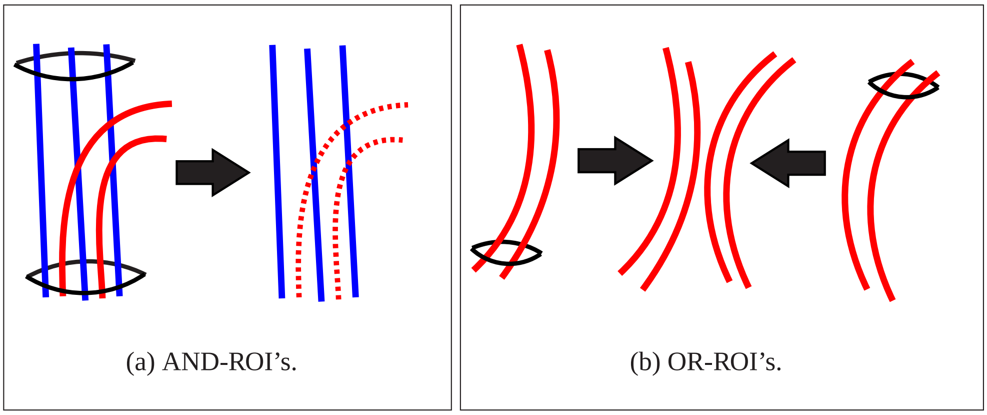
Of course, these six anatomical structures represent only a small portion of the complete set of structures known to be present in the human brain. There are several reasons for not using the other structures:

- Some anatomical structures require the parameters of the fiber tracking algorithm to be set to values that do not produce reasonable results when doing an all volume fiber tracking. For instance, some structures can only be found using a very low anisotropy threshold. This means that fiber tracking is done in areas where the main eigenvector is very unreliable. This leads to a lot of erroneous fibers. With ROI fiber tracking, most of these erroneous fibers are automatically removed, because they do not pass through the required ROI's. With all volume fiber tracking, these erroneous fibers are much harder to remove, and therefore the anisotropy threshold has to be set higher.
- Some anatomical structures can not yet be reliably identified with the current fiber tracking techniques. It is expected that in the future more anatomical structures can be recognized with the aid of higher-resolution scans [24], or more robust fiber tracking techniques. More generally, each technique that improves the quality of the fibers has an impact on the structures that can be used for classification.

Manually specifying for each individual fiber to which bundle it belongs is a tedious and time-consuming task. Therefore, classification is done using an semi-automatic approach similar to the ROI fiber tracking technique described in chapter 2. Each bundle is defined by a number of manually defined regions (ROI's). Fibers are classified as belonging to a particular bundle if they pass through a specific number of the ROI's.

The classification procedure consists of two steps:

1. Manual placement of ROI's. As with ROI fiber tracking, ROI's are 2D regions that are placed in areas for which is known that fibers from a particular structure pass through them. There are two types of ROI's: AND-ROI's and OR-ROI's. A fiber has to pass through all AND-ROI's and through at least one OR-ROI. Figure 4.4 illustrates the different kinds of ROI's.
2. Classification of individual fibers. Each fiber that intersects the required number of ROI's associated with a bundle is classified as belonging to that particular bundle.



(a) AND-ROI's.

(b) OR-ROI's.

Figure 4.4: Illustration of the different kinds of ROI's.

Fibers that cannot be assigned to a bundle are labelled ”Unclassified” and are not part of the gold standard. Therefore, they are not used for validation. There are several reasons why some fibers may be unclassifiable:

- They are part of an anatomical structure that is not part of the gold standard.
- Due to problems with the fiber tracking technique (see section 2.3.2) fibers can be incomplete or incorrect. Incomplete fibers often do not pass through the required number of ROI's, and are therefore automatically excluded. Incorrect fibers might be composed of parts that belong to more than a single anatomical structure. These ambiguous fibers could pass through all the required ROI's and have to be removed manually in some cases. Finally, some fibers do not correspond to actual anatomical structures at all, because they are entirely the result of an artifact in the DTI data set.

Note that the complete set of fibers is clustered, but only the classified fibers are used for validation.

4.3 Validation methods

This section examines various methods for comparing the gold standard with the results of automated clustering methods. We want to be able to say which distance measure and which cluster method can be used to partition the fibers into meaningful and anatomically correct clusters. More specifically, we want to be able to measure to which extent clustering methods and proximity measures produce clusters that match the bundles of the manual classification, according to the preferences of physicians.

The optimal parameter settings for a clustering algorithm can be found by searching for a clustering that has the highest agreement with the gold standard. Often, it is not immediately clear which parameter settings give the best results. For example, the output of the hierarchical clustering algorithm is a dendrogram. With a set of *n* fibers, the dendrogram can be cut at *n* places producing *n* possible clusterings. Manually searching for the optimal match would take a considerable amount of time. However, with the aid of a validation method the optimal level of the dendrogram can be found much more easily.

A validation method must take into account a number of aspects, which are discussed in section 4.3.1. Next, two kinds of validation methods are examined: Receiver Operating Characteristic (ROC) curves [4] and external indices [17].

4.3.1 Validation criteria

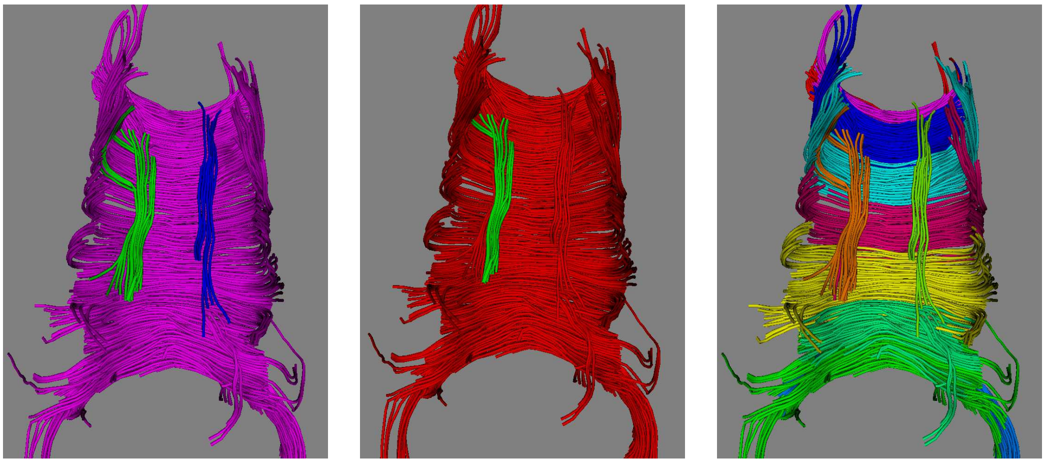
There are two important aspects, which we call correctness and completeness, that must be considered when comparing two partitions of items:

Correctness. Fibers belonging to different anatomical structures should not be clustered together. Correctness can be expressed as a percentage: 100% correctness means that no fiber is clustered together with any fibers from other bundles, and 0% correctness means that each fiber is clustered together with all fibers from other bundles.

Completeness. Fibers belonging to the same anatomical structure should be clustered together. Completeness can also be expressed as a percentage: 100% completeness means that each fiber is clustered together with all other fibers from the same bundle, and 0% completeness means that there is no fiber that is clustered together any fibers from the same bundle.

In practice there is a tradeoff between these two aspects. More correctness means less completeness, and vice versa. Achieving 100% correctness is not difficult: put every fiber into a singleton cluster, but this results in a completeness of 0%. On the other hand, achieving 100% completeness is also not difficult: put every fiber into the same cluster, but this results in a correctness of 0%. The comparison methods discussed in this section are all based on the notion that a good clustering must be both correct and complete with respect to the manual classification.

Here is an example to illustrate the concepts of correctness and completeness. Figure 4.5 shows three different partitions of the same set of fibers: the gold standard and two clusterings. The clustering in figure 4.5b is incorrect, because several bundles from the gold standard are together in the same cluster. The clustering in figure 4.5c is incomplete because a bundle from the gold standard is subdivided into several clusters.



(a) Gold standard.

(b) Incorrect clustering.

(c) Incomplete clustering.

Figure 4.5: Three different partitions of the same set of fibers.

A question is if a validation method should weight correctness and completeness equally. Physicians from the MMC indicated (see section 4.4) that they found an incorrect clustering worse than an incomplete clustering. For instance, consider the incorrect clustering in figure 4.5b. In this clustering, one of the small bundles has become almost invisible, because it is clustered together with the large bundle. On the other hand, in the incomplete clustering (figure 4.5c) all bundles are clearly visible. Also, if we wish to improve the clusterings manually, then this would be much easier for the incomplete clustering: we only have to specify which clusters should be joined. Manually improving the incorrect clustering is much more difficult, because we have to specify for each fiber to which cluster it belongs. As a result, we want to be able to specify different weights to the aspects of correctness and completeness.

Another aspect to consider is the contribution each bundle of the gold standard has. Should a bundle that consists of a lot of fibers weight more than a bundle which consists of a few fibers? For instance, in our gold standard the corpus callosum is a relatively large bundle. In most cases, it is an order of magnitude larger than some of the smaller bundles, like the cingula or the fornix. But for a global overview the corpus callosum is not of more interest than either the cingula or the fornix. In such an overview, large structures tend to dominate visually anyway, whereas small structures might be more difficult to see. Therefore, we can assume that each bundle is equally important, regardless of the number of fibers.

4.3.2 Receiver Operator Characteristic curves

Receiver Operating Characteristic (ROC) curves are often used to measure the performance of medical image analysis techniques [4]. A typical problem in this context might be the detection of abnormalities in MRI images. For such a problem, performance refers to the number of correct decisions made by the detection algorithm. More correct decisions indicates a better algorithm.

The following section defines ROC curves in the context of detection problems. After that, definitions for ROC curves in the context of fiber clustering are given.

General definitions

The decisions made by a detection algorithm can be categorized as follows (with respect to the gold standard or the actual clinical state):

True positive *(TP)*. The detection algorithm *correctly* decides that an abnormality exists.

True negative *(TN)*. The detection algorithm *correctly* decides that no abnormality exits.

False negative *(FN)*. The detection algorithm *incorrectly* decides that an abnormality exists.

False positive *(FP)*. The detection algorithm *incorrectly* decides that no abnormality exists.

This is summarized in the table 4.2.

		Detection algorithm	
		abnormality present	abnormality not present
Gold standard	abnormality present	true positive	false negative
	abnormality not present	false positive	true negative

Table 4.2: Categories for the decisions of a detection algorithm [4].

Sensitivity is the frequency of reporting an abnormality in the situation there actual is one. It is defined in terms of the number of true positives (*TPs*) and false negatives (*FNs*):

$$sensitivity = \frac{TPs}{(TPs{+}FNs)}.$$

Specificity is the frequency of reporting no abnormality when no abnormality exists. It is defined in terms of the number of true negatives (*TNs*) and false positives (*FPs*):

$$specificity = \frac{TNs}{(TNs{+}FPs)}.$$

A ROC curve shows the trade-off between sensitivity and specificity. Typically, a ROC curve is plotted with the ”true positive” fraction (sensitivity) on the vertical axis, and the ”false positive” fraction (1-specificity) on the horizontal axis [4]. Figure 4.6 shows an example of a ROC curve. The perfect algorithm has a ROC curve that reaches the upper left corner of the chart: at this point both sensitivity and specificity are 1.0. A guessing algorithm has a ROC curve that is the diagonal from the lower left corner to the upper right corner.

To create a ROC curve one has to identify the parameter in the detection algorithm that most directly controls the trade-off between sensitivity and specificity [4]. The ROC curve is defined by a number of (specificity, sensitivity) pairs that are obtained by varying this parameter. The situation is more difficult if there are several parameters that have an influence on the trade-off.

A common measure for the goodness of a ROC curve is the area under the curve (AUC) [4]. The AUC for a perfect algorithm is 1.0 and the AUC for a guessing algorithm is 0.5.

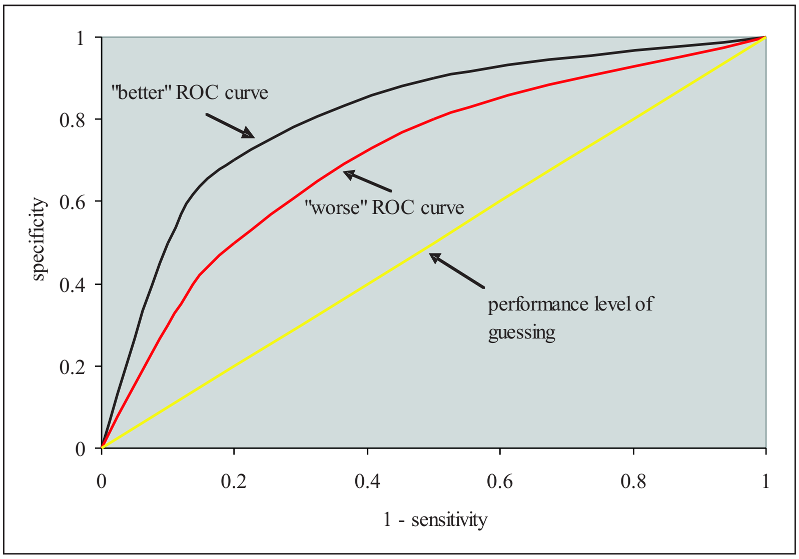


Figure 4.6: Example of a ROC curve, adapted from Browyer [4].

Fiber clustering definitions

ROC curves are usually applied to situations in which a detection algorithm has to make a binary choice: either the input is normal or abnormal. However, in the context of fiber clustering there are multiple bundles that must be ”detected” by multiple clusters.

The gold standard B and the cluster result C are both partitions of n items. The gold standard consists of R bundles and the cluster result consists of S clusters:

$$\begin{aligned} B &= \{b_1, b_2, \ldots, b_R\}, \\ C &= \{c_1, c_2, \ldots, c_S\}. \end{aligned}$$

Let u_i be the number of fibers in bundle b_i and let v_j be the number of fibers in cluster c_j.

Assume that we are only trying to detect fibers from bundle b_i. Furthermore, assume that cluster c_j is the set of fibers that the clustering algorithm presents as solution. Now, the complete set of fibers can be categorized as follows:

- TP_{ij} = the number of fibers that belong to both bundle b_i as well as cluster c_j.
- FN_{ij} = the number of fibers that belong to bundle b_i, but do not belong to cluster c_j.

- FP_{ij} = the number of fibers that belong to cluster c_j, but do not belong to bundle b_i.
- TN_{ij} = the number of fibers that do not belong to cluster c_j and do not belong to bundle b_i.

This is summarized in the table 4.3.

	in cluster c_j	not in cluster c_j
in bundle b_i	TP_{ij}	FN_{ij}
not in bundle b_i	FP_{ij}	TN_{ij}

Table 4.3: Possible categories for a pair of fibers.

Sensitivity can then be defined for bundle b_i and cluster c_j:

$$\mathit{sensitivity}(b_i, c_j) = \frac{TP_{ij}}{TP_{ij} + FN_{ij}}.$$

Sensitivity measures the completeness of a bundle and cluster pair: it is the fraction of fibers from bundle b_i that are in cluster c_j. A value of 1.0 means that all fibers from bundle b_i are in cluster c_j. If no fibers from bundle b_i are in cluster c_j then sensitivity is 0.0.

Similarly, specificity can be defined for bundle b_i and cluster c_j:

$$\mathit{specificity}(b_i, c_j) = \frac{TN_{ij}}{TN_{ij} + FP_{ij}}.$$

Specificity measures the correctness of a bundle and cluster pair. Specificity is 1.0 when cluster c_j only contains fibers from bundle b_i. It is 0.0 if cluster c_j only contains fibers from other bundles.

Now we can define the sensitivity of a bundle by taking the weighted average of the sensitivity scores of the individual clusters:

$$\mathit{bundle\text{-}sensitivity}(b_i) = \sum_{j=1}^{S} \frac{TP_{ij}}{u_i} \mathit{sensitivity}(b_i, c_j).$$

Bundle-sensitivity gives an indication of the completeness of a bundle b_i: if there is a cluster c_j that contains all the fibers from bundle b_i then bundle-sensitivity is

1.0. It approaches 0.0 if there is no cluster that contains more than one fiber from bundle b_i, for instance if every fiber is in a singleton cluster.

Similarly, we can the specificity of a bundle by taking the weighted average of the specificity scores of the individual clusters:

$$\textit{bundle-specificity}(b_i) = \sum_{j=1}^{S} \frac{TP_{ij}}{u_i} \textit{specificity}(b_i, c_j).$$

Bundle-specificity gives an indication of the correctness of a bundle b_i: it is 1.0 if all clusters that contain fibers from b_i do not contain fibers from other bundles. It approaches 0.0 if every fiber from bundle b_i is together with all fibers from other bundles, for instance if every fiber is in the same cluster.

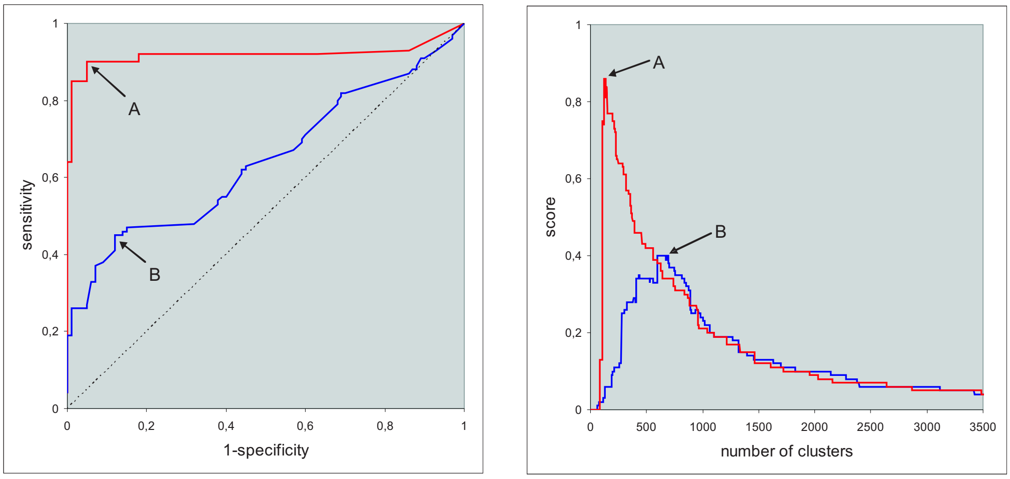
We measure the sensitivity and specificity of the complete cluster result by taking the average of the bundle-sensitivity and bundle-specificity scores:

$$\textit{overall-sensitivity} = \sum_{i=1}^{R} \frac{1}{R} \textit{bundle-sensitivity}(b_i),$$

$$\textit{overall-specificity} = \sum_{i=1}^{R} \frac{1}{R} \textit{bundle-specificity}(b_i).$$

An overall sensitivity of 1.0 means that all fibers that belong to the same bundle are also together in the same cluster. An overall specificity of 1.0 means that all fibers belonging to different bundles are also in different clusters.

As already mentioned earlier, to create a ROC curve one has to identify a parameter in the clustering algorithm that controls the trade-off between sensitivity and specificity. For instance, for a hierarchical clustering algorithm this parameter is the level at which the dendrogram is cut. By cutting the dendrogram at various levels we can obtain different clusterings, varying from a clustering with 1 cluster to a clustering with n clusters, where n is the number of fibers. Each clustering has a different value for sensitivity and specificity. In general, a clustering with too few clusters has a high sensitivity and a low specificity, whereas a clustering with too many clusters has a high specificity and a low sensitivity. The best clustering has both a high sensitivity as well as a high specificity. Such a clustering can be identified by the ROC curve that comes closest to the upper left corner of a ROC plot.



(a) ROC curves.

(b) Score chart.

Figure 4.7: Two charts created by comparing the cluster results of a hierarchical clustering algorithm to the gold standard.

Figure 4.7a shows two different ROC curves for the hierarchical clustering algorithm using different distance measures. The distance measure used for ROC curve *A* seems to produce better clusterings than the distance measure used for ROC curve *B*. The arrows indicate the position on the ROC curve corresponding to the clustering that has the most agreement with the gold standard. The AUC's for these two examples are 0.88 for ROC curve *A* and 0.60 for ROC curve *B*.

For visualization purposes we are not only interested in the global performance of a clustering algorithm, but also in the quality of each individual clustering. Therefore, we also assign a single score to each clustering:

$$\textit{ROC score} = \textit{overall-sensitivity} * \textit{overall-specificity}.$$

This score gives an overall indication of the quality of a clustering with respect to the gold standard. A score close to 1.0 means that the clustering is both complete (sensitive) as well as correct (specific). Consequently, a score close to 0.0 means that either the clustering is incomplete, incorrect or both.

Figure 4.7b shows an example of a plot with the ROC score on the vertical axis and the number of clusters on the horizontal axis. Again, the arrows indicate the optimal score.

In conclusion: ROC curves can be used for measuring the performance of detection algorithms. We proposed some additional definitions of ROC curves which make them also usable in the context of fiber cluster validation. However, during the verification of the validation methods (see section 4.4) it became clear that they are inappropriate for our purposes. Although there is still room for improvement, we decided to abandon the use of ROC curves altogether. Instead, we started using external indices which are normally used for the validation of cluster results with a gold standard.

4.3.3 External indices

An external index is a statistical measure that indicates the agreement between two partitions of a set of items [17]. In our case the items are fibers, and the segmentations to be compared are the manual classification, which is external to the clustering process, and a segmentation produced by a clustering algorithm. The level of agreement between these two partitions is expressed in a fraction between 0 and 1: if the two partitions agree perfectly then the index returns a value of 1, and if the two partitions disagree completely then the index is 0.

Definitions

The manual classification B and the cluster result C are both partitions of n items. The gold standard consists of R bundles and the cluster result consists of S clusters:

$$\begin{aligned} B &= \{b_1, b_2, \ldots, b_R\}, \\ C &= \{c_1, c_2, \ldots, c_S\}. \end{aligned}$$

Table 4.4 shows a contingency table, which is defined as follows: Let cell n_{ij} be the number of fibers that are both in bundle b_i as well as in cluster c_j. The row sum u_i is the number of fibers in bundle b_i and the column sum v_j is the number of fibers in cluster c_j.

Bundle/Cluster	c_1	c_2	$\dots$	c_S	Sums
b_1	n_{11}	n_{12}	$\dots$	n_{1S}	u_1
b_2	n_{21}	n_{22}	$\dots$	n_{2S}	u_2
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
b_R	n_{R1}	n_{R2}	$\dots$	n_{RS}	u_R
Sums	v_1	v_2	$\dots$	v_S	n

Table 4.4: Contingency table.

Let a be the number of pairs of fibers that are both in the same bundle and in the same cluster. a can be defined in terms of the contingency table:

$$a = \sum_{i=1}^{R} \sum_{j=1}^{S} \binom{n_{ij}}{2}.$$

Let b be the number of pairs of fibers that are both in the same bundle, but are not in the same cluster:

$$b = \sum_{i=1}^{R} \binom{u_i}{2} - \sum_{i=1}^{R} \sum_{j=1}^{S} \binom{n_{ij}}{2}.$$

Let c be the number of pairs of fibers that are not in the same bundle, but are in the same cluster:

$$c = \sum_{j=1}^{S} \binom{v_j}{2} - \sum_{i=1}^{R} \sum_{j=1}^{S} \binom{n_{ij}}{2}.$$

Let d be the number of pairs of fibers that are not in the same bundle and not in the same cluster:

$$d = \binom{n}{2} - a - b - c.$$

The number of pairs that are in the same bundle is

$$m1 = a + b,$$

and the number of pairs that are in the same cluster is

$$m2 = a + c.$$

The total number of pairs is denoted

$$M = \binom{n}{2} = a + b + c + d.$$

This is summarized in contingency table 4.5.

	same cluster	different cluster	
same bundle	a	b	$m1$
different bundle	c	d	$M - m1$
	$m2$	$M - m2$	M

Table 4.5: Categories of pairs of fibers.

The number of pairs on which the gold standard and the cluster result agree is $a + d$. Consequently, $b + c$ is the number of pairs on which the gold standard and the cluster result disagree.

Rand Index

The Rand index [17] is defined as the number of ”agreement” pairs divided by the total number of pairs:

$$Rand \quad = \quad (a + d)\Big/M.$$

If the two partitions agree completely then the Rand index returns a value of 1.00. Although the lower-limit of this index is 0.0, this value is rarely returned with real data [21]. This is because the Rand index is not corrected for agreement by chance.

Adjusted Rand Index

The Adjusted Rand index [15] is the Rand index corrected for chance agreement. The general form of a statistic S that is corrected for chance is:

$$S' = \frac{S - E(S)}{\mathrm{Max}(S) - E(S)}.$$

In this equation, Max(S) is the upper-limit of S, and $E(S)$ is the expected value of S. If the statistic S returns its expected value then the corrected statistic S' is 0.0, and if S returns a value of 1.0 then S' also returns 1.0.

The expected value of the Rand index is the value that is returned for a configuration of the contingency table in which the bundle and cluster sums are fixed, but the fibers are randomly assigned to clusters. Assuming a hypergeometric baseline distribution, the expected values for a and d are [15]:

$$\begin{aligned} E\,(a) &= \frac{m1m2}{M}, \\ E\,(d) &= \frac{(M - m1)(M - m2)}{M}. \end{aligned}$$

The expected value of the Rand is then:

$$\begin{aligned} E\,((a+d)/M) &= \frac{E(a) + E(d)}{M} \\ &= \frac{\frac{m1m2}{M} + \frac{(M-m1)(M-m2)}{M}}{M} \\ &= \frac{m1m2 + (M - m1)(M - m2)}{M^2}. \end{aligned}$$

As a result, the Adjusted Rand index is defined as:

$$\begin{aligned} AR &= \frac{((a+d)/M) - E\,((a+d)/M)}{1 - E\,((a+d)/M)} \\ &= \frac{a - (m1m2)/M}{(m1 + m2)/2 - (m1m2)/M}. \end{aligned}$$

For two partitions that agree perfectly the Adjusted Rand index returns a value of 1.0. For partitions where all agreement can be attributed to chance a value around

0.0 is returned (the lower bound of this index can be negative, depending on the partitioning).

Milligan and Cooper [21] compared the Rand, Adjusted Rand and a number of other external indices and concluded that the Adjusted Rand index is the measure of choice for cluster validation. However, the Adjusted Rand index has an undesired feature for our purposes: it does not account for bundles that are of widely varying sizes. That is, the Adjusted Rand index measures agreement on the level of fibers, not on the level of bundles. As a result, a bundle with a large number of fibers is weighted more than a bundle with a small number of fibers.

Normalized Adjusted Rand Index

To take into account the requirement that bundles should be weighted equally, we define the Normalized Adjusted Rand index. The idea is to modify the contingency table such that each bundle has the same number of fibers. A way to achieve this is by setting the row sum u_i of each bundle b_i in the contingency table to some nonnegative value k and to multiply each entry n_{ij} by a factor $\frac{k}{u_i}$ (see table 4.6).

Bundle/Cluster	c_1	c_2	$\dots$	c_S	Sums
b_1	$n_{11}\frac{k}{u_1}$	$n_{12}\frac{k}{u_1}$	$\dots$	$n_{1S}\frac{k}{u_1}$	k
b_2	$n_{21}\frac{k}{u_2}$	$n_{22}\frac{k}{u_2}$	$\dots$	$n_{2S}\frac{k}{u_2}$	k
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
b_R	$n_{R1}\frac{k}{u_R}$	$n_{R2}\frac{k}{u_R}$	$\dots$	$n_{RS}\frac{k}{u_R}$	k
Sums	v'_1	v'_2	$\dots$	v'_S	Rk

Table 4.6: Normalized contingency table.

The column sum v'_j is computed by taking the sum of the new cell values:

$$v'_j = \sum_{i=1}^{R} k\frac{n_{ij}}{u_i}.$$

With this contingency table we can calculate new values for a, b, c, d, $m1$, $m2$, M:

$$
\begin{aligned}
a' &= \sum_{i=1}^{R} \sum_{j=1}^{S} \binom{k\frac{n_{ij}}{u_i}}{2} \\
b' &= R\binom{k}{2} - a' \\
c' &= \sum_{j=1}^{S} \binom{v'_j}{2} - a' \\
d' &= \binom{Rk}{2} - a' - b' - c' \\
m1' &= a' + b' \\
m2' &= a' + c' \\
M' &= \binom{Rk}{2}.
\end{aligned}
$$

A remaining question is which value to use for k. Actually, what we would like to achieve is that the value of k does not make a difference for the outcome of the index. However, a simple example shows that this is not the case. Consider contingency table 4.7.

	c_1	c_2	
b_1	2	2	4
b_2	2	2	4
	4	4	8

Table 4.7: Example contingency table.

In this example we have 8 items in 2 bundles. Because the items are evenly distributed over 2 clusters, we expect a value of 0.0 from the Adjusted Rand index. Table 4.8 gives the values returned by the Adjusted Rand index for increasing values of k.

k	AR
1	-0.500000
10	-0.055556
100	-0.005051
1000	-0.000501
10000	-0.000050
100000	-0.000005

Table 4.8: Values of the Adjusted Rand index.

It seems that for increasing values of k, we get an Adjusted Rand index that converges to the expected value of 0.0. Indeed, this behavior is confirmed by Milligan and Cooper [21]. They report that increased cluster sizes result in an Adjusted Rand index that converges to their expected value.

Therefore, we propose to take k to infinity. The definition of the Normalized Adjusted Rand becomes:

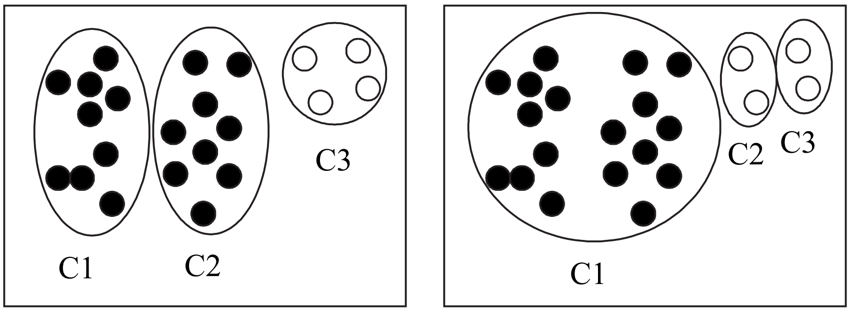
$$\begin{aligned} NAR &= \lim_{k\to\infty} \frac{a' - (m1'm2')/\binom{Rk}{2}}{(m1' + m2')/2 - (m1'm2')/\binom{Rk}{2}} \\ &= \frac{2f - 2Rg}{2f - Rf - R^2} \end{aligned}$$

with

$$\begin{aligned} f &= \sum_{j=1}^{S} \left(\sum_{i=1}^{R} \frac{n_{ij}}{u_i} \right)^2 \\ g &= \sum_{i=1}^{R} \sum_{j=1}^{S} \frac{n_{ij}^2}{u_i^2}. \end{aligned}$$

The complete calculation can be found in Appendix A.

Here is an example to illustrate the difference between the Adjusted Rand index and the Normalized Adjusted Rand index. Given is a set of 22 objects consisting of two bundles (see figure 4.8). The objects are clustered in two different ways: in clustering 1 the large bundle is split into two clusters, and in clustering 2 the small bundle is split into two clusters.



(a) Clustering 1: Large bundle split.

(b) Clustering 2: Small bundle split.

Figure 4.8: Example of the difference between the Adjusted Rand and Normalized Rand index.

The Adjusted Rand index returns a value of 0.38 for clustering 1, and 0.96 for clustering 2, which means that clustering 2 is considered much better than clustering 1. Indeed, if we solely look at the number of correct pairs then clustering 2 can be considered better. But if we instead examine the clustering at the level of bundles then these clusterings can be considered equal: in each clustering one of the bundles is complete, and one is subdivided. The Normalized Rand index returns a value of 0.75 for both clusterings, and thus better reflects the equality of the clusterings.

Weighted Normalized Adjusted Rand index

We propose a final modification to the Adjusted Rand index that enables us to weight correctness and completeness differently. The indices that are based on the Rand index assume that the correctness and completeness of a clustering are equally important, but this may be not necessarily the case in our situation. Actually, physicians assign different weights to the aspects of correctness and completeness.

Let us first define the Rand index in terms of the normalized contingency table:

$$\begin{aligned} NR &= \frac{a' + d'}{a' + b' + c' + d'} \\ &= 1 - \frac{b'}{a' + b' + c' + d'} - \frac{c'}{a' + b' + c' + d'} \\ &= 1 - \frac{b'}{M'} - \frac{c'}{M'}. \end{aligned}$$

In this equation the fraction $\frac{b'}{M'}$ indicates the incompleteness of the clustering. The fraction $\frac{c'}{M'}$ indicates the incorrectness of the clustering. We propose the following definition for a Weighted Normalized Rand index *WNR*:

$$WNR = 1 - 2(1-\alpha)\frac{b'}{M'} - 2\alpha\frac{c'}{M'}.$$

If $\alpha = 0.5$ then correctness and completeness are weighted equally. If α is between 0.0 and 0.5 then completeness is weighted more and if α is between 0.5 and 1.0 then correctness is weighted more.

The expected value of *WNR* becomes:

$$\begin{aligned} E(WNR) &= E\left(1 - 2(1-\alpha)\frac{b'}{M'} - 2\alpha\frac{c'}{M'}\right) \\ &= 1 - 2(1-\alpha)E\left(\frac{b'}{M'}\right) - 2\alpha E\left(\frac{c'}{M'}\right) \\ &= 1 - 2(1-\alpha)\frac{m1'(M' - m2')}{M'^2} - 2\alpha\frac{m2'(M' - m1')}{M'^2} \end{aligned}$$

because the expected value of b is $\frac{m1(M-m2)}{M}$ and the expected value of c is $\frac{m2(M-m1)}{M}$.

Now the Weighted Normalized Rand index (*WNAR*) is defined as:

$$\begin{aligned} WNAR &= \lim_{k\to\infty} \frac{NWR - E(NWR)}{1 - E(NWR)} \\ &= \frac{f - Rg}{f - \alpha Rf - R^2 - \alpha R^2} \end{aligned}$$

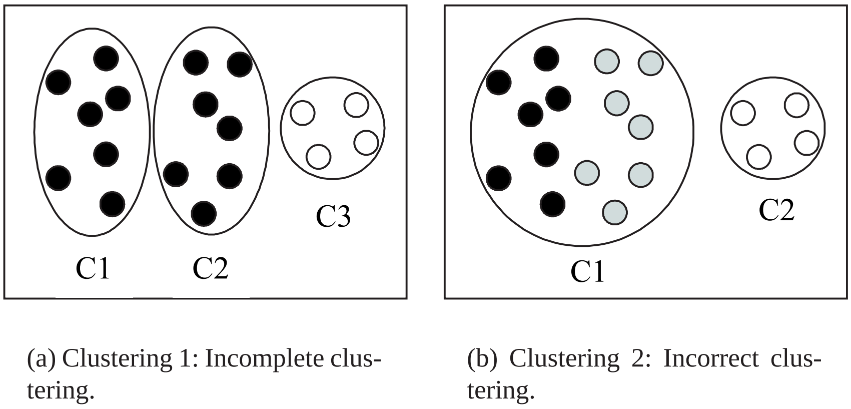
with

$$f = \sum_{j=1}^{S} \left(\sum_{i=1}^{R} \frac{n_{ij}}{u_i} \right)^2$$

$$g = \sum_{i=1}^{R} \sum_{j=1}^{S} \frac{n_{ij}^2}{u_i^2}.$$

The complete calculation can be found in Appendix B.

Here is an example to illustrate the WNAR index. Figure 4.9 shows two clusterings of a set of 18 objects. Clustering 1 consists of two bundles and can be considered incomplete. Clustering 2 consists of three bundles and can be considered incorrect. Table 4.9 shows the values that are obtained from the WNAR index for both clusterings for different values of α.



(a) Clustering 1: Incomplete clustering.

(b) Clustering 2: Incorrect clustering.

Figure 4.9: Example to illustrate the WNAR. Color is used to distinguish between bundles.

			WNAR	
α	Completeness	Correctness	Clustering 1	Clustering 2
0.00	100%	0%	0.60	1.00
0.25	75%	25%	0.67	0.73
0.50	50%	50%	0.75	0.57
0.75	25%	75%	0.86	0.47
1.00	0%	100%	1.00	0.40

Table 4.9: Values of the WNAR index.

4.4 Verification of validation methods

The goal is to identify the best validation method for measuring the agreement between the cluster results and the gold standard. Our approach is based on the notion that the optimal validation method assigns scores to clusterings that are similar to the scores assigned by a physician. For this purpose, two physicians from the Maxima Medical Center were asked to rank a number of clusterings. These clusterings were also ranked by the various validation methods discussed in the last section. The ranking of the physicians was then compared to the rankings from the validation methods.

Table 4.10 gives the ranking of the physicians and the scores assigned by the various validation methods. In this table, cc stands for corpus callosum, cr for corona radiata (both hemispheres), cg for cingula (both hemispheres) and fx for fornix. A ”++” means that the physicians found that particular aspect very good, a single ”+” means that they found that aspect good, a ”0” means they found it average (depending on the context), and a ”−” means that they found this aspect bad in every situation. Notice that no aspect has been labelled ”very bad”. This is because it is very difficult for physicians to distinguish between a ”bad” and a ”very bad” aspect; a ”bad” aspect is already something they cannot relate to.

The clusterings can be categorized based on the overall quality:

Good. Clustering A and B were considered good by the physicians. The validation methods agree with the physicians and return fairly high values, although the ROC scores are a little lower. The reason none of the validation methods return a 1.0 for these clusterings, is because there were some fibers

	Correctness				Completeness					ROC	AR	WNAR				
	cc	cr	cg	fx	cc	cr	cg	fx	Overall			0.00	0.25	0.5	0.75	1.00
A	++	++	++	++	++	++	++	++	good	0.84	0.96	0.83	0.86	0.89	0.92	0.96
B	++	++	++	++	+	+	+	++	good	0.73	0,85	0.70	0.75	0.81	0.88	0.96
C	++	++	++	++	0	0	+	++	average	0.56	0.09	0.51	0.59	0.68	0.82	1.00
D	++	++	++	++	0	0	+	++	average	0.65	0.36	0.60	0.68	0.75	0.86	1.00
E	+	++	+	++	0	0	+	++	average	0.61	0.31	0.57	0.64	0.71	0.84	0.99
F	++	++	++	++	+	+	−	++	average	0.65	0.77	0.63	0.67	0.71	0.76	0.82
G	−	++	−	−	++	++	++	++	bad	0.78	0.90	0.90	0.80	0.72	0.66	0.61
H	++	−	++	−	++	++	++	++	bad	0.86	0.93	0.88	0.81	0.75	0.70	0.66
I	−	−	−	+	−	−	−	+	very bad	0.40	0.01	0.34	0.33	0.32	0.30	0.29
									Rank correlation:	0.15	0.25	0.05	0.28	0.54	0.93	0.75

Table 4.10: Ranking of the physicians and the scores assigned by the validation methods.

from the smaller bundles that were in different clusters. The physicians did not mind that these outliers were clustered apart, because they were visually different.

Average. The physicians found the clusterings C, D, E and F average. All four clusterings suffered from the same defect: some bundles were subdivided. Although this might be desirable in some situations, the subdivision was not part of the gold standard. Therefore, the validation methods found these four clusterings to be incomplete. The physicians did not mind the subdivision in some cases, because large bundles like the corpus callosum and corona radiata can be further subdivided. The physicians found it less desirable that a small bundle like the cingula was subdivided. The Adjusted Rand index returns very low scores for clusterings in which the corpus callosum was subdivided into a number of smaller clusters (clustering C, D and E). The WNAR index returns higher, more balanced scores and seems to reflect the opinion of the physicians better, especially if correctness is weighted more then completeness.

Bad. The clusterings G and H were considered bad by the physicians, because several bundles from the gold standard were clustered together. The Adjusted Rand index returns very high scores for these clusterings because the largest bundle (the corpus callosum) is complete. The WNAR index with an α lower than 0.5 also assigns too high values to these clusterings. The WNAR index with $\alpha = 0.5$ returns values that are equal to the values for the average clusterings, and is therefore not able to distinguish between a clustering that is considered average and a clustering that is considered bad. However, if correctness is weighted more than completeness then the values returned by the WNAR index better reflect the opinion of the physicians.

Very bad. Clustering I was considered very bad because it was both incorrect as well as incomplete. Here the validation methods agree with the opinion of the physicians and return very low values.

The rank correlation is computed by comparing the ranking of the physicians to the ranking of the validation methods. Thus, only the ordering is taken into account assuming that the difference in quality between the clusterings is equal. Although this is not entirely true, we still use the rank correlation to get an indication of agreement between the ranking of the physicians and the rankings of the validation methods.

Table 4.11 gives the average values for the different categories. The WNAR index with $\alpha = 0.75$ is the only index that assigns values to the clusterings of different categories in a proper way. Both the Adjusted Rand index as well as the ROC index overestimate the quality of the bad clusterings. The bad clusterings are also overestimated by the WNAR index if completeness is weighted more than correctness. The WNAR index with $\alpha = 0.50$ does not distinguish between average and bad clusterings. The WNAR index with $\alpha = 1.00$ assigns too high scores to the average clusterings, because it completely ignores the completeness aspect. It is therefore not able to distinguish between a good clustering and an average clustering. Note that all methods return a low value for the very bad clustering.

			WNAR				
Overall	ROC	AR	0.00	0.25	0.50	**0.75**	1.00
good	0.79	0.91	0.77	0.80	0.85	**0.90**	0.96
average	0.62	0.38	0.58	0.64	0.71	**0.82**	0.95
bad	0.82	0.92	0.89	0.81	0.74	**0.68**	0.64
very bad	0.40	0.01	0.34	0.33	0.32	**0.30**	0.29

Table 4.11: Values of the validation methods per category.

Figure 4.10 shows the relation between the rank correlation and the weight α of the WNAR index. It confirms that 0.75 is indeed the optimal weight for validating the clusterings that were used in this experiment.

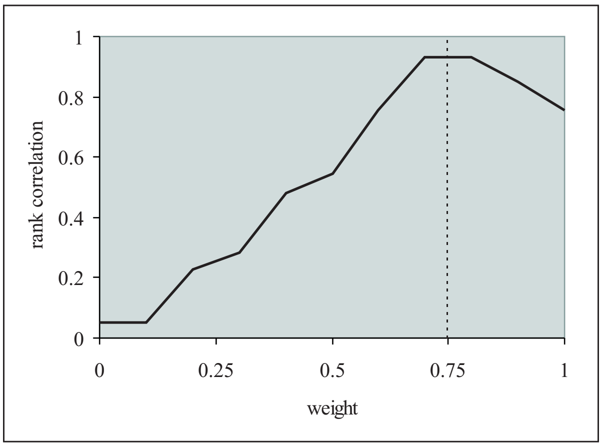


Figure 4.10: Graph showing the relation between the weight α and the rank correlation.

According to these results, the ranking created with WNAR index with $\alpha = 0.75$ has the most correspondence with the ranking of the physicians. Because this verification experiment is too small to be statistically significant, a larger experiment with a more complete gold standard is necessary to confirm these results. However, time constraints prevented us from performing such an experiment. Nevertheless, based on this experiment, the WNAR index with $\alpha = 0.75$ seems to be the most suitable validation method available and is therefore used in the next chapter to validate the cluster results.

Chapter 5

Results

This chapter presents the results of this study. It shows clusterings of fibers that can be obtained by using the cluster methods and proximity measures described in chapter 3. Furthermore, it demonstrates how the quality of these cluster results can be assessed by using the validation techniques described in the previous chapter.

5.1 Experimental setup

All visualizations in this chapter are created with the DTI Tool originally developed by Berenschot [2] in collaboration with the Maxima Medical Center (MMC). This tool visualizes DTI data in a variety of ways, one of which is fiber tracking. To allow for the classification and clustering of fibers, we extended the DTI Tool. See appendix C for a more detailed description of our modifications.

For the experiments, three different DTI data sets from healthy adults were used. Each data set has a resolution of $128 \times 128 \times 30$ with a voxel size of $1.8 \times 1.8 \times 3.0$mm. For each data set we defined a gold standard which consisted of the structures described in section 4.2. The gold standard of the first data set was verified by physicians. The data sets were selected at random: the only selection criterium was that the structures of the gold standard could be found using fiber tracking.

The fiber tracking algorithm has a considerable number of parameters. Table 5.1 identifies the parameter settings that we have used to create fibers. It lies outside the scope of this project to study how each of these parameters affects the fibers produced by the fiber tracking algorithm. Intuitively we can say that the minimum length and the minimum anisotropy have a significant influence on the outcome of the fiber tracking algorithm, and consequently, on the performance of the clustering methods. In general, lowering the minimum length produces shorter fibers for which identification is more difficult; lowering the minimum anisotropy reduces the separation between the various white matter bundles. Thus, a more challenging set of fibers can be created by choosing a lower minimum length and a lower minimum anisotropy.

Parameter	Value
Seed distance	1.0 mm
Anisotropy index	Cl
Minimum anisotropy	0.20
Minimum length	20 mm
Maximum length	500 mm
Maximum angle	100
Step length	0.10 voxel

Table 5.1: Parameters of the fiber tracking algorithm.

However, for our purposes the configuration given above is sufficient: fiber tracking with seeding throughout the whole volume gives us a set of 3500-5000 fibers, which can be clustered in approximately 15-20 minutes, depending on the chosen proximity measure and clustering method. Furthermore, each bundle of the manual classification contains at least 10 fibers with these settings.

5.1.1 Proximity measures

It is not clear from literature which of the available proximity measures described in section 3.3 produces the best results. As a starting point, we implemented the following four measures:

- Closest point distance,
- Mean of closest points distance,
- Hausdorff distance,
- End points distance.

We selected these measures primarily for practical reasons: they are straightforward to implement and require no extra parameters. However, if these four measures prove to be insufficient, more complex measures could be used in future experiments.

5.2 Hierarchical clustering results

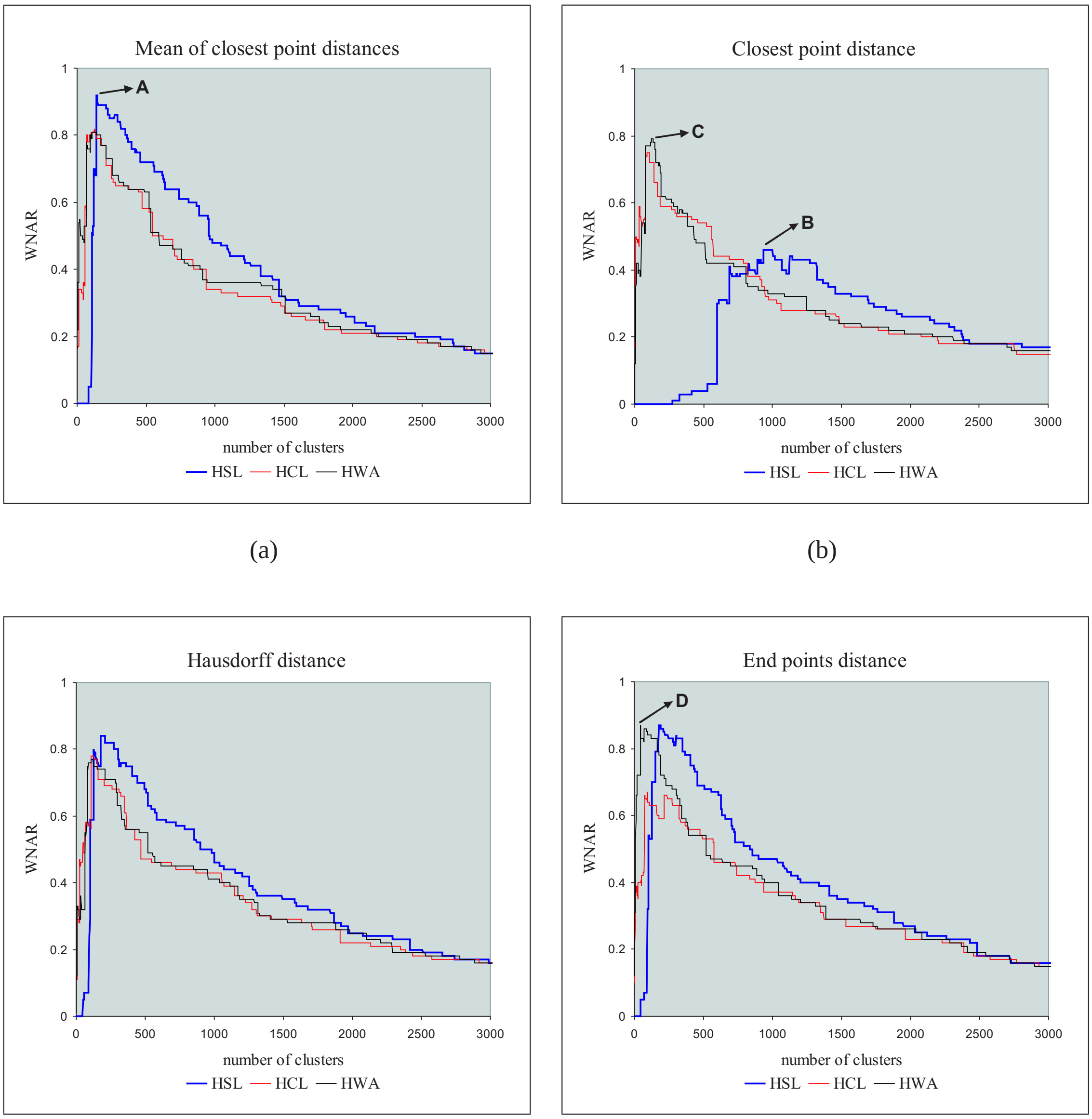
The first method that we have used for fiber clustering is the hierarchical clustering algorithm, which is a well established method that has been applied in a large number of contexts. It has been used for fiber clustering by Zhang and Laidlaw [27, 26, 29].

Hierarchical clustering is a very flexible clustering method: different results can be obtained by varying the way clusters are merged. Three hierarchical variations were implemented: single-link (HSL), complete-link (HCL) and weighted-average (HWA). Note that in contrast with the single-link and complete-link methods, the weighted-average method has not yet been used in the context of fiber clustering. See section 3.4.1 for a more detailed description of these methods.

The following section shows the results for the hierarchical clustering methods applied to a single data set. After that, the results for multiple data sets are presented.

5.2.1 Single data set

Hierarchical clustering methods have a single parameter that controls the output of the algorithm: the level at which the dendrogram is cut. A graph can be plotted by comparing the clustering at each level of the dendrogram to the manual classification. Figure 5.1 shows the graphs for the four proximity measures. Each graph is plotted with the number of clusters n on the horizontal axis and the value of the WNAR index with $\alpha = 0.75$ on the vertical axis. Each graph contains the output from the three different hierarchical variants: single-link (thick blue curve), complete-link (thin red curve) and weighted-average (thin black curve).



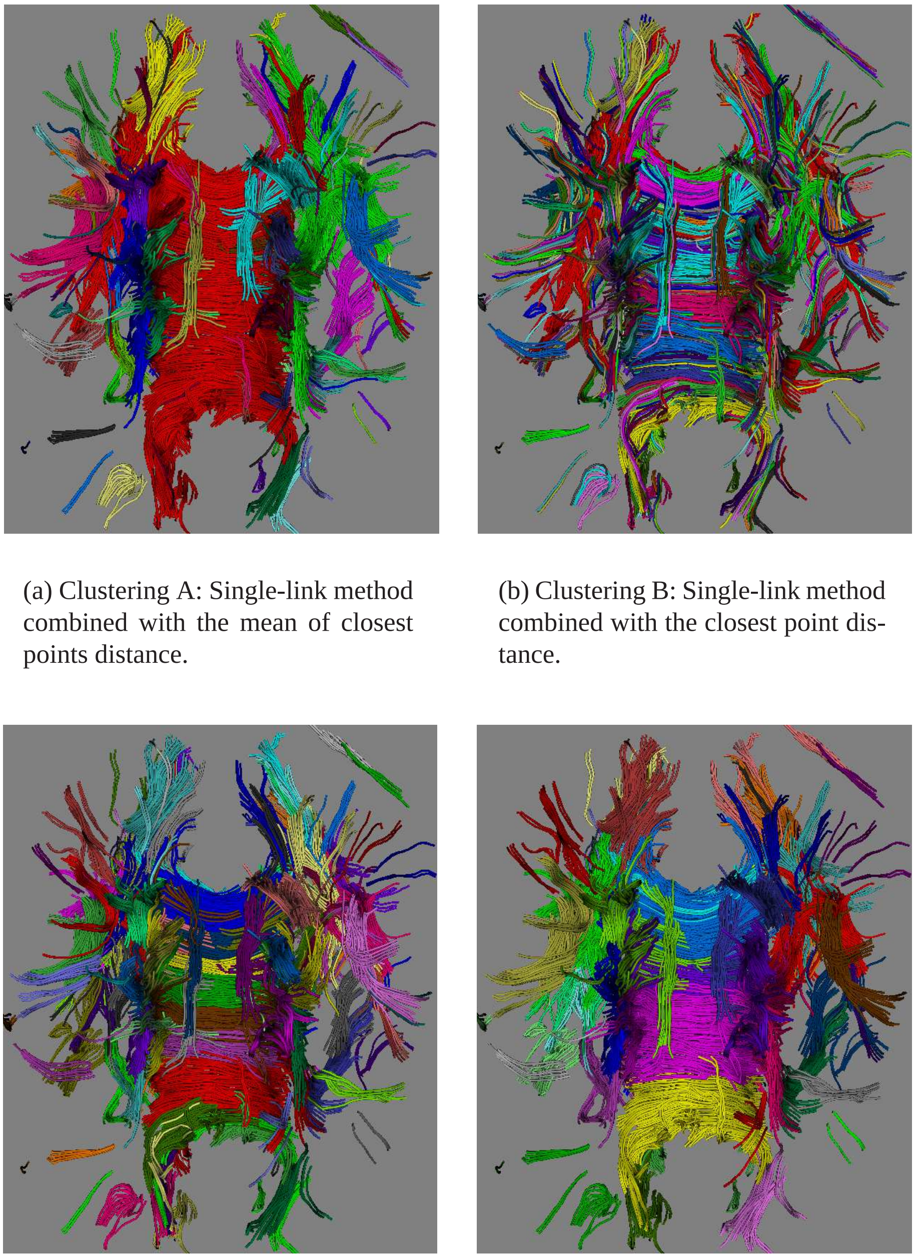
(a)

(b)

(c)

(d)

Figure 5.1: Graphs of the hierarchical clustering methods.



(a) Clustering A: Single-link method combined with the mean of closest points distance.

(b) Clustering B: Single-link method combined with the closest point distance.

(c) Clustering C: Complete-link method combined with the closest point distance.

(d) Clustering D: Weighted-average method combined with the end points distance.

Figure 5.2: Hierarchical clusterings of the first data set.

Table 5.2 gives the maximum values obtained from the WNAR index for each combination of proximity measure and hierarchical clustering method. The mean of closest points measure combined with the single-link method produces a clustering that has the most correspondence with the gold standard. This clustering is obtained by cutting the dendrogram at the level of 141 clusters (see figure 5.2a). The worst optimal clustering has 933 clusters and is also created with the single-link method, but now combined with the closest point measure (see figure 5.2b). The value of the WNAR index for this clustering is 0.46. Noticeable about this combination of clustering method and proximity measure is the high number of clusters needed to get to a clustering that is somewhat reasonable. Figure 5.2c shows the clustering obtained by using the closest point measure combined with the complete-link method. Figure 5.2d shows the clustering obtained by using the end point distance with weighted-average method.

Proximity measure	HSL		HWA		HCL	
	WNAR	*n*	WNAR	*n*	WNAR	*n*
Mean of closest points	**0.92**	141	0.81	110	0.82	125
Closest point	0.46	933	0.79	120	0.77	77
Hausdorff	0.84	178	0.77	107	0.78	107
End points	0.87	175	0.87	44	0.67	95

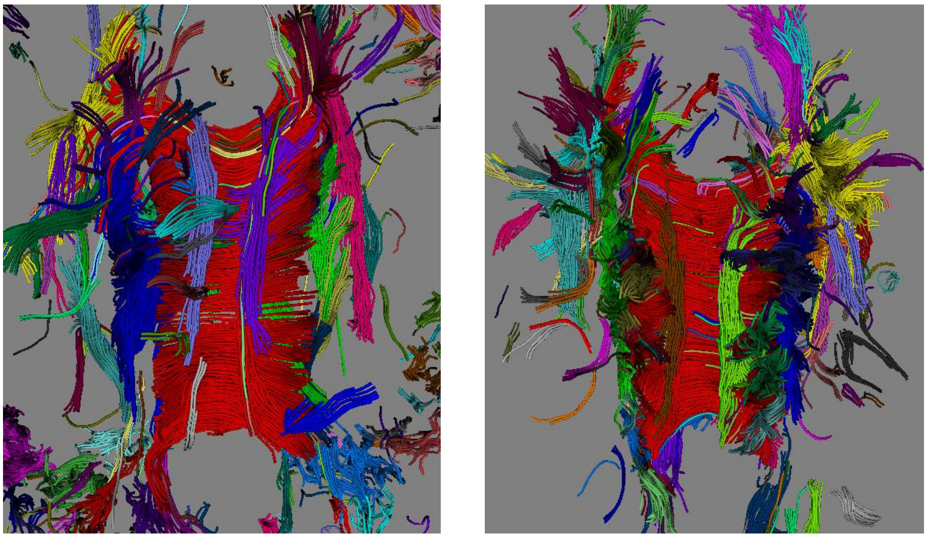
Table 5.2: Results of the hierarchical clustering algorithm for the first data set.

5.2.2 Multiple data sets

Table 5.3 gives the optimal values for the WNAR index for the other two data sets. Additionally, the average of the optimal values for all three data sets is given. Figure 5.3a shows the optimal clustering for the second data set, and figure 5.3b shows the optimal clustering for the third data set. Both clusterings are created with the single-link method combined with the mean of closest points measure. The value of the WNAR index for these clusterings is 0.99 and 0.95 for the second and third data set, respectively.

Proximity	Data set 2			Data set 3			Average		
measure	HSL	HWA	HCL	HSL	HWA	HCL	HSL	HWA	HCL
Mean of closest	**0.99**	0.90	0.87	**0.95**	0.86	0.77	**0.95**	0.86	0.82
Closest point	0.50	0.82	0.79	0.50	0.76	0.69	0.49	0.79	0.75
Hausdorff	0.85	0.82	0.85	0.91	0.77	0.66	0.89	0.80	0.72
End points	0.88	0.82	0.77	0.93	0.72	0.74	0.87	0.79	0.76

Table 5.3: Results of the hierarchical clustering algorithm for multiple data sets.



(a) Optimal clustering for the second data set: Created with the single-link method combined with the mean of closest points distance.

(b) Optimal clustering for the third data set: Created with single-link method combined with the mean of closest points distance.

Figure 5.3: Optimal hierarchical clusterings for the second and third data set.

For three of four measures, the single-link performs better than the weighted-average and complete-link methods. These higher values can be explained by the fact that the single-link method manages to keep the fibers from the larger bundles together. This is largely due to the chaining effect, which is a known characteristic of the single-link method [18]. Even the fibers from a large, elongated structure like the corpus callosum are almost entirely in a single cluster (the red cluster in figures 5.2a, 5.3a and 5.3b).

The chaining effect of the single-link method becomes a disadvantage when using the closest point measure, which can be seen as very ”optimistic”: two fibers only need to have two neighboring points to be considered close. Furthermore, the single-link method can also be seen as very ”optimistic”: two clusters only need to have two neighboring fibers to be considered close. In many cases, this results is an overestimation of the similarity between clusters.

The complete-link method has the opposite characteristic of the single-link method: it tries to make globular clusters, even when the data contains elongated structures [18]. This characteristic explains why a large structure like the corpus callosum is subdivided into a number of approximately equally sized clusters. In general, this reduces the completeness of a complete-link clustering, which explains the lower values of the WNAR index. Due to the requirement that bundles should be weighed equally, the values for the complete-link method are not that much lower than for the single-link method; the normalization of the bundles done by the WNAR index works to the advantage of methods that tend to break up large bundles.

The weighted-average method seems to fall in between the single-link and complete-link methods. The elongated structures are still subdivided, but the clusters tend to be less globular than for the complete-link method.

Concerning the proximity measures, the mean of closest points measure achieves the highest values for the WNAR index, although the difference with the end points distance and the Hausdorff distance is not very large. As mentioned above, the closest point distance performs poorly with the single-link method, but performs reasonably well with the complete-link and weighted-average methods. This is probably because the conservative nature of these methods counterbalances the overly optimistic nature of the closest point measure.

5.3 Shared nearest neighbor clustering results

The second method that we have used for fiber clustering is the shared nearest neighbor algorithm described in section 3.4.5. In contrast with hierarchical clustering, the shared nearest neighbor algorithm has not yet been used in the context of fiber clustering.

This section shows the results for the shared nearest neighbor algorithm. First, the results for a single data set are given, then the results for multiple data sets are presented.

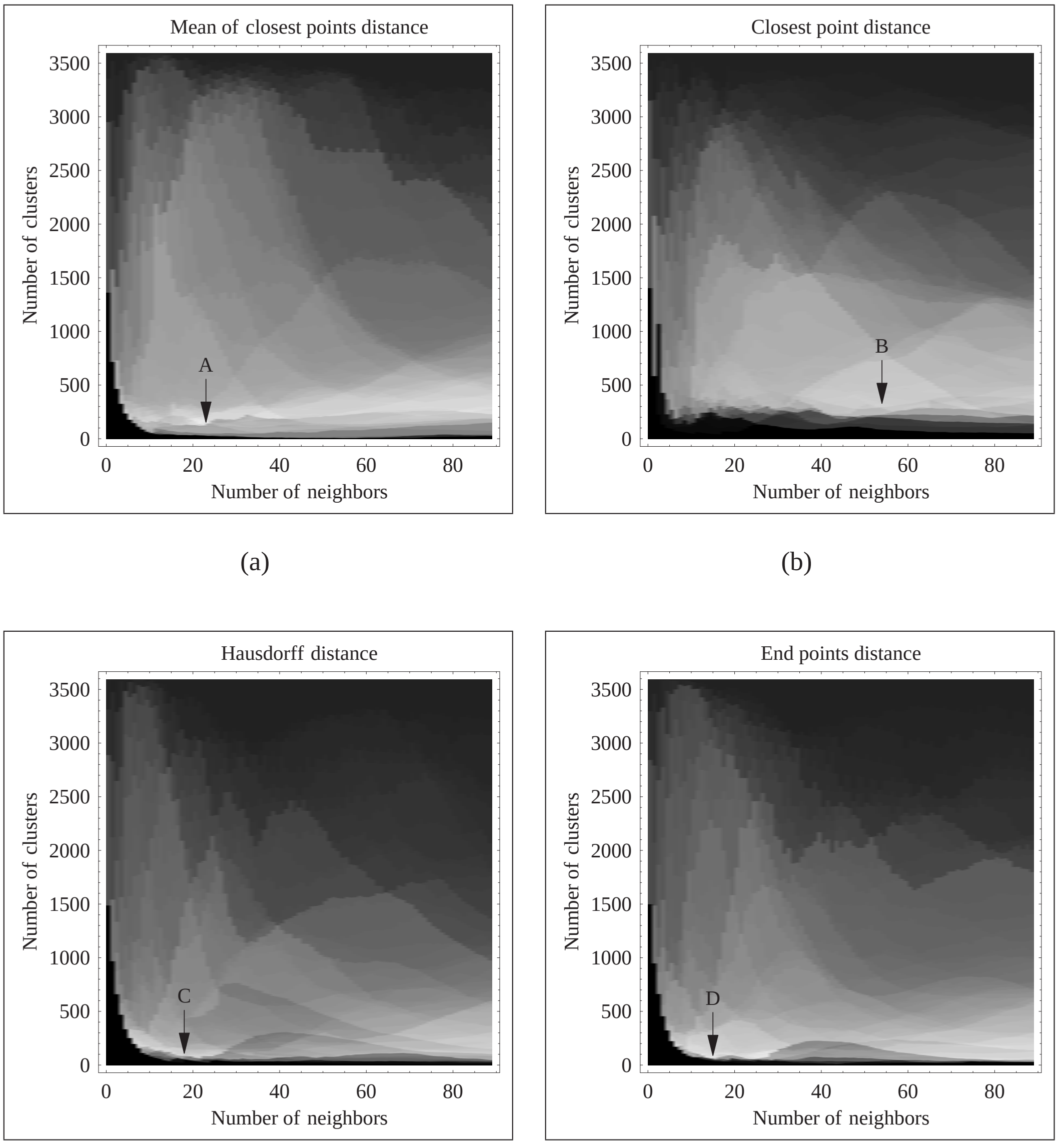
5.3.1 Single data set

The shared nearest neighbor algorithm has two parameters: the number of neighbors and the edge threshold. In general, an increased edge threshold results in an increased number of clusters. By fixing the number of neighbors and varying the edge threshold from 0 to a certain maximum value, every possible clustering for that particular number of neighbors can be obtained.

Figure 5.4 shows density plots for the four proximity measures. Each plot has the number of neighbors on the x-axis, the number of clusters on the y-axis and the value of the WNAR index represented as a grey value: black corresponds to a value of 0 and white to a value of 1. The arrows indicate the optimal clusterings which are shown in figure 5.5.

A number of observations can be made about the density plots. First of all, the highest values are found around the 50 to 250 clusters. Clusterings with less than 50 clusters tend to be incorrect, and clusterings with more than 250 clusters tend to be incomplete. This can be seen in the plots: the grey level starts black for a low number of clusters and then increases rapidly to the highest grey level before gradually fading to black again. This is actually similar to the graphs of the hierarchical clustering methods, in which the curve rises substantially near the beginning, reaches an optimum, and then gradually decreases again. The graph in figure 5.6 illustrates this: it is obtained with the shared nearest neighbor algorithm in combination with the mean of closest points measure. It has the number of clusters on the horizontal axis and the value of the WNAR index with $\alpha = 0.75$ on the vertical axis. The number of neighbors has been set to 23 (black curve) and 85 (red curve). As can be observed, this graph looks similar to the graphs in figure 5.1.

Secondly, in the plots of the mean of closest points measure, the end points measure and the Hausdorff measure the highest grey levels appear between the 10 and 25 neighbors. For the closest point distance on the other hand, the highest grey levels appear around the 50 neighbors.



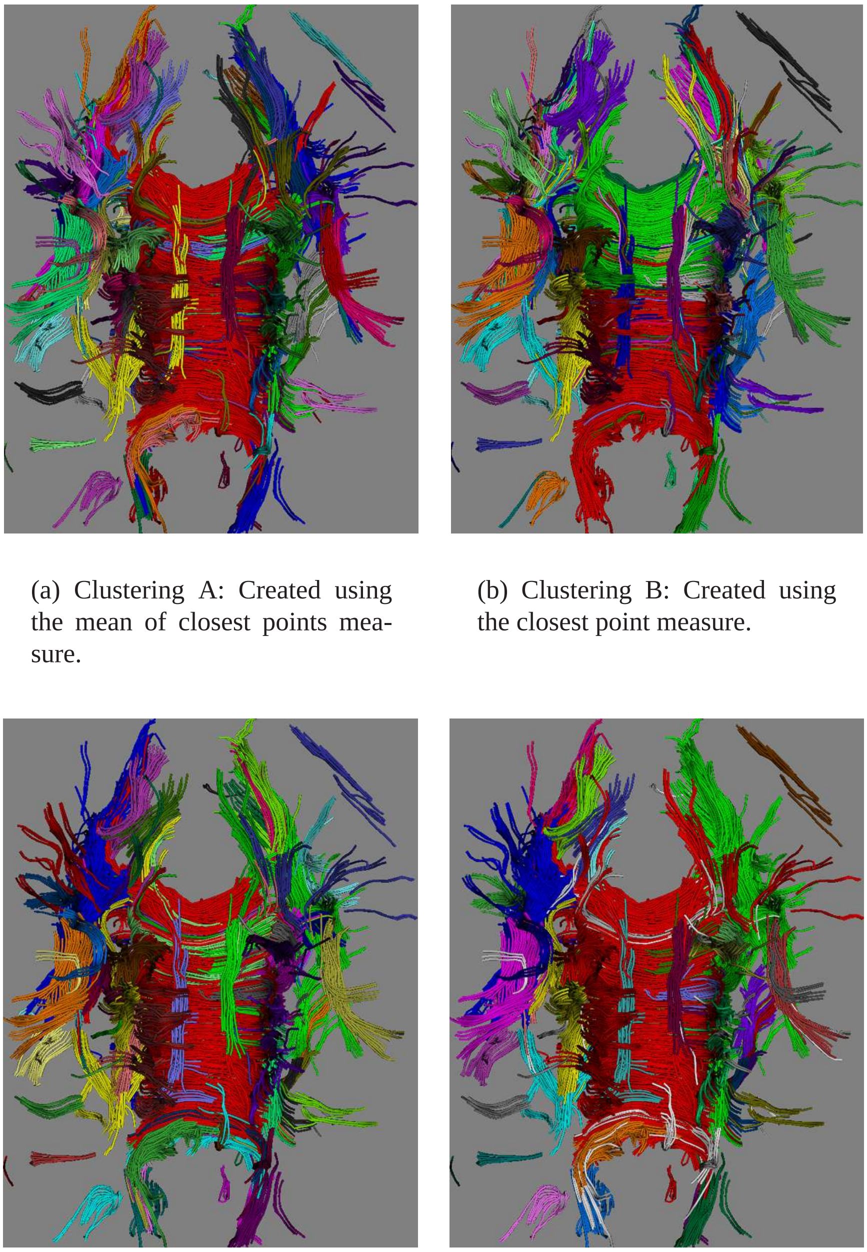
(a)

(b)

(c)

(d)

Figure 5.4: Density plots of the shared nearest neighbor algorithm results.



(a) Clustering A: Created using the mean of closest points measure.

(b) Clustering B: Created using the closest point measure.

(c) Clustering C: Created using the Hausdorff measure.

(d) Clustering D: Created using the end points measure.

Figure 5.5: Shared nearest neighbor clusterings.

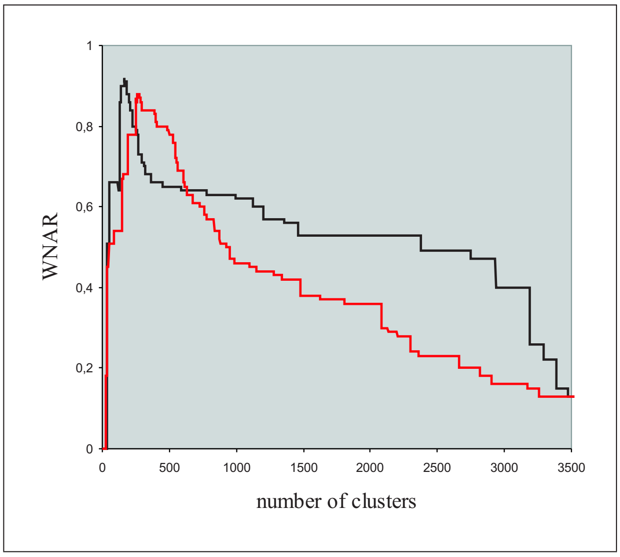


Figure 5.6: Graph for the mean of closest of points measure in combination with the shared nearest neighbor algorithm with 23 neighbors (black curve) and 85 neighbors (red curve).

Table 5.4 shows the shared nearest neighbor results for a single data set. For each proximity measure the highest reached WNAR index is given. Also, the number of neighbors k, the edge threshold τ and the number of clusters n for the optimal clustering are given.

Proximity measure	WNAR	k	τ	n
Mean of closest points	**0.93**	23	2667	145
Closest point	0.82	54	42,065	320
Hausdorff	0.87	18	863	100
End points	0.92	15	329	79

Table 5.4: Results of the shared nearest neighbors algorithm for the first data set.

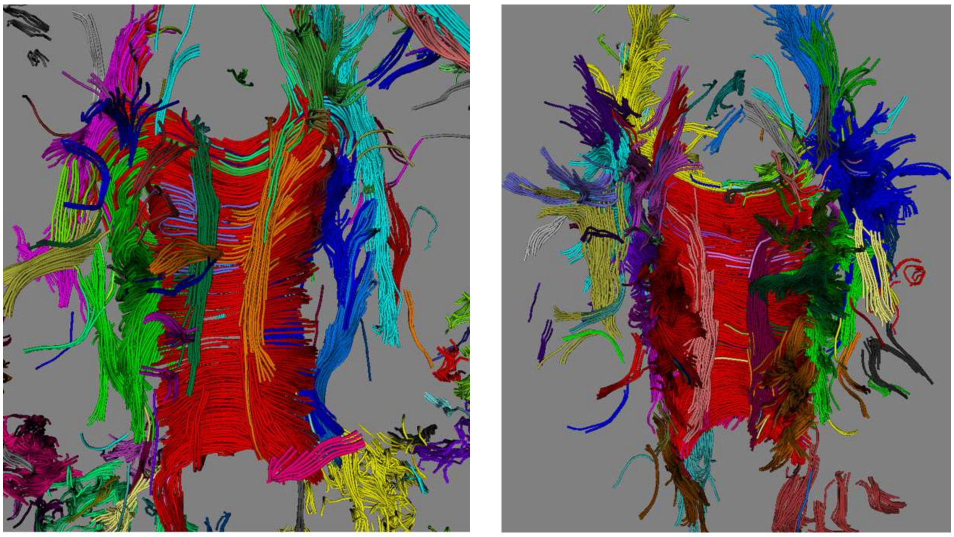
The mean of closest points distance achieves the highest value for the WNAR index. The clustering created using the end points measure is almost as good according to the WNAR index. Noteworthy is the high number of neighbors for the optimal clustering of the closest point measure. The high number of clusters indicates that it is more incomplete than the other optimal clusterings of the other measures. This is visually confirmed in figure 5.5b in which can be seen that the corpus callosum is subdivided, while it is complete in other three clusterings (the large red cluster in figures 5.5a, 5.5c and 5.5d).

5.3.2 Multiple data sets

Table 5.5 shows results for all three data sets. For each proximity measure the highest value for the WNAR index is given. Additionally, the number of neighbors k and the edge threshold τ with which the best clustering was obtained is also shown. Figure 5.7 shows the optimal clusterings of the mean of closest points measure.

Proximity	Data set 1	Data set 2			Data set 3			Avg
measure	WNAR	WNAR	k	τ	WNAR	k	τ	WNAR
Mean of closest	**0.93**	**1.00**	9	0	**0.91**	79	136,748	**0.95**
Closest points	0.82	0.83	89	203,631	0.86	35	8,994	0.84
Hausdorff	0.87	0.99	16	495	0.89	88	177,407	0.92
End points	0.92	0.97	10	9	0.92	88	183,567	0.94

Table 5.5: Results of the shared nearest neighbors algorithm for multiple data sets.



(a) Clustering of the second data set.

(b) Clustering of the third data set.

Figure 5.7: Shared nearest neighbor clusterings created with the mean of closest points distance for the second and third data set.

The shared nearest neighbor algorithm seems to be able to find both the small and the large bundles of the manual classification. Indeed, a visual inspection reveals that the clusterings produced by the shared nearest neighbor algorithm are very similar to hierarchical single-link clusterings. This is reflected in the scores of the WNAR index which are also similar.

The choice of proximity measure seems to have less influence, although the clusterings produced with the closest point distance are given somewhat lower values by the WNAR index.

The difficulty with the shared nearest neighbor algorithm is choosing appropriate values for the number of neighbors and the edge threshold. Noticeable is the apparent lack of a relation between the number of neighbors and the optimal value for the WNAR index. For instance, using the mean of closest points measure, the optimal clustering for the first data set is found with 23 neighbors, for the second data set with 9 neighbors and for the third data set with 79 neighbors.

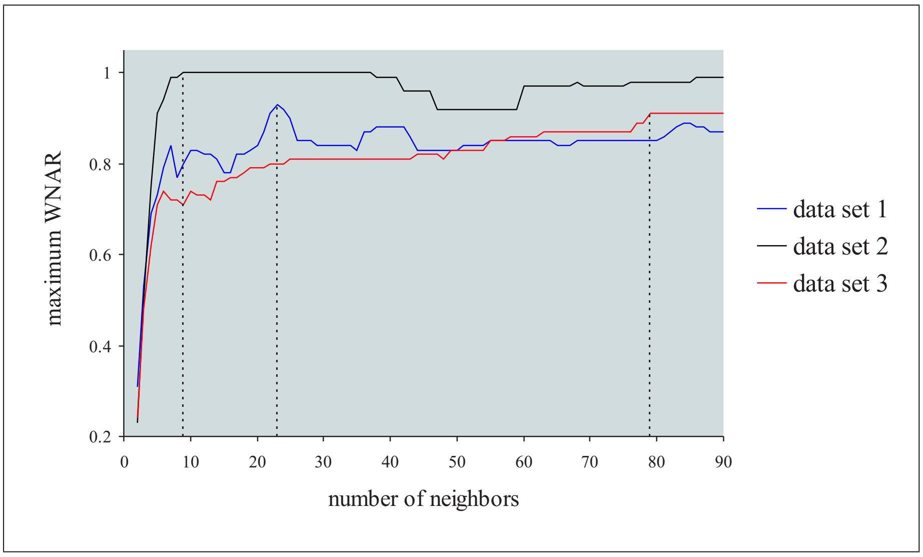


Figure 5.8: Optimal WNAR values for the shared nearest neighbor algorithm.

The graph in figure 5.8 shows how the maximum value of the WNAR index fluctuates. This graph has been created using the mean of closest points measure. It has the number of neighbors on the horizontal axis and the maximum value of the WNAR index for a specific number of neighbors on the vertical axis. The dotted lines indicate for each data set at what number of neighbors the optimal value is first achieved. As can be seen, there is no number of neighbors at which all data sets achieve their optimal value of the WNAR index. If we would have to pick a single number of neighbors for all three data sets, then the best choice seems to be

85 neighbors, at which the maximum values for the WNAR index are 0.89, 0.98 and 0.91, for the first, second and third data set, respectively.

A related problem is setting the edge threshold. When a manual classification is available, an exhaustive search can find the optimal edge threshold for a particular number of neighbors. Without such an aid however, the number of possible values for the edge threshold is very large, especially if the number of neighbors is very high. A possibility would be to set the desired number of clusters instead of the edge threshold. The algorithm could then search for an edge threshold that produces the clustering with the specified number of clusters, although it is not guaranteed that this clustering exists.

5.4 Evaluation

Table 5.6 gives for each data set the average WNAR values for the optimal clusterings. For the hierarchical clustering algorithm, all optimal clusterings were obtained with the single-link method combined with mean of closest point measure. For the shared nearest neighbor algorithm the optimal clusterings were obtained with the end points measure and the mean of closest point measure. So, the mean of closest point distance seems to be the best choice for measuring proximity between fibers, although the difference with the end points and Hausdorff distance is small, in particular when combined with the shared nearest neighbor algorithm. The closest point distance performs less well, especially in combination with the single-link method.

Proximity measure	HSL	HWA	HCL	SNN
Mean of closest points	**0.95**	0.86	0.82	**0.95**
Closest point	0.49	0.79	0.75	0.84
Hausdorff	0.89	0.80	0.72	0.92
End points	0.87	0.79	0.76	0.94

Table 5.6: Summary of the results.

As for clustering methods, the difference between the hierarchical single-link method and shared nearest neighbor method is minimal. A larger experiment with more data sets is necessary to see if there is really no difference in clustering quality between these two algorithms. If we look from a practical point of view then the hierarchical clustering algorithm seems somewhat more user friendly for our purposes: specifying the number of clusters is more intuitive than setting the number of neighbors and the edge threshold.

The results of the experiments presented in this chapter can be seen as a demonstration of the techniques described in the previous chapters. Due to time constraints, we had to restrict ourselves to a limited number of data sets, proximity measures and clustering methods.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This study has presented techniques for the clustering of brain fibers. The goal was to overcome the visual cluttering that occurred when doing fiber tracking with seeding throughout the whole volume.

We identify the following four contributions:

- The first contribution is the application of the shared nearest neighbor clustering algorithm in the context of fiber clustering. We used this algorithm because it can find clusters of different sizes and shapes in data that contains noise and outliers.

- The second contribution is a framework to evaluate fiber clustering methods. Our approach is based on the manual classification of the fibers in a number of bundles that correspond to anatomical structures. By comparing the manually defined bundles to the automatically created clusters we can get an estimation of the cluster quality.

- The third contribution is a new index to validate the fiber clusters based on the preferences of physicians. We created the WNAR index after we found that the indices available in literature are not suited to the task of fiber clustering. In particular, the existing indices do not address the following:

  - Bundles of the manual classification should be weighed equally, regardless of the number of fibers. A bundle may contain few fibers, but this does not mean that it is less important. On the contrary, because small bundles are often concealed by large bundles, it is essential that these small bundles are visually different.

  - Physicians prefer correctness above completeness, because a correct clustering is visually more appealing than a complete clustering. In an incorrect clustering, fibers belonging to different anatomical bundles are clustered together, which makes it difficult to distinguish between bundles.

- The final contribution is the comparison of different clustering methods with the new index. We demonstrated how the validation and clustering techniques can be used on DTI data sets of human brains. We compared the results of the shared nearest neighbor algorithm to results of the hierarchical clustering method used by another research group. Both algorithms performed equally well on the data sets that we selected for the experiments, but the shared nearest neighbor algorithm has multiple parameters which makes finding the optimal clustering difficult. Furthermore, we found that the mean of closest points distance measure gives a good approximation of the distance between a pair of fibers.

6.2 Future work

During the course of this project we discovered a number of areas which deserve further investigation. Here is a list of future research:

- Increase the number of bundles that are included in the manual classification. The current manual classification only contains six anatomical structures, which results in a large number of unclassified fibers that cannot be used for validation. More bundles means that more fibers can be classified. A more complete manual classification enables a more accurate assessment of the cluster results.

- Examine the effect of the fiber tracking parameters. These parameters determine to a large extent the quantity and quality of the produced fibers. For instance, a more challenging set of fibers can be created by choosing a lower minimum anisotropy.

- Conduct a larger experiment with more data sets. Our experiment has been conducted on a limited number of data sets, and can therefore not give definitive answers.

- Cluster fibers from the heart or other muscle tissues. It would be interesting to examine how the cluster methods perform on non-brain fibers.
- Develop more sophisticated proximity measures. Currently, only the fiber point coordinates are used and the information of the original tensor is largely ignored. For instance, the directions of the eigenvectors could also be used to get an indication of similarity.

Bibliography

[1] P.J. Basser and C. Pierpaoli. Microstructural and physiological features of tissues elucidated by quantitative-diffusion-tensor mri. *Journal of Magnetic Resonance*, 111(3):209–219, June 1996.

[2] Guus Berenschot. Visualization of diffusion tensor imaging. Master's thesis, Eindhoven University of Technology, 2003.

[3] Denis Le Bihan. Looking into the functional architecture of the brain with diffusion mri. *Nat Rev Neurosci*, 4(6):469–80, 2003.

[4] K. W. Bowyer. Chapter 10: Validation of medical image analysis techniques. In *Handbook of Medical Imaging volume 2*, pages 567–607. SPIE-International Society, 2000.

[5] A. Brun, H. Knutsson, H. J. Park, M. E. Shenton, and C.-F. Westin. Clustering fiber tracts using normalized cuts. In *Seventh International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'04)*, Lecture Notes in Computer Science, pages 368–375, 2004.

[6] Anders Brun, Hae-Jeong Park, Hans Knutsson, and Carl-Fredrik Westin. Coloring of DT-MRI fiber traces using laplacian eigenmaps. In *Computer Aided Systems Theory (EUROCAST'03), Lecture Notes in Computer Science 2809*, pages 564–572. Springer Verlag, February 24–28 2003.

[7] M. Catani, R. J. Howard, S. Pajevic, and D. K. Jones. Virtual in vivo interactive dissection of white matter fasciculi in the human brain. *NeuroImage*, 17:77–94, 2002.

[8] Isabelle Corouge, Sylvain Gouttard, and Guido Gerig. Towards a shape model of white matter fiber bundles using diffusion tensor MRI. In *International Symposium on Biomedical Imaging*, pages 344–347, 2004.

[9] Zhaohua Ding, John C. Gore, and Adam W. Anderson. Case study: reconstruction, visualization and quantification of neuronal fiber pathways. In *Proceedings of the conference on Visualization '01*, pages 453–456. IEEE Computer Society, 2001.

[10] Zhaohua Ding, John C. Gore, and Adam W. Anderson. Classification and quantification of neuronal fiber pathways using diffusion tensor mri. *Magnetic Resonance in Medicine*, 49:716–721, 2003.

[11] R. K. Dodd. A new approach to the visualization of tensor fields. *Graph. Models Image Process.*, 60(4):286–303, 1998.

[12] Levent Ertöz, Michael Steinbach, and Vipin Kumar. Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In *Proceedings of the Third SIAM International Conference on Data Mining*, 2003.

[13] B. J. Jellison et al. Diffusion tensor imaging of cerebral white matter: a pictorial review of physics, fiber tract anatomy, and tumor imaging patterns. *AJNR Am J Neuroradiol*, 25(3):356–369, 2004.

[14] Henry Gray. *Anatomy of the human body. 20th ed., thoroughly rev. and re-edited by Warren H. Lewis.* Philadelphia: Lea & Febiger, 1918; Bartleby.com, 2000.

[15] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.

[16] N. Lori J. S. Shimony, A. Z. Snyder and T. E. Conturo. Automated fuzzy clustering of neuronal pathways in diffusion tensor tracking. In *Proc. Intl. Soc. Mag. Reson. Med. 10*, May 2002.

[17] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

[18] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[19] G. L. Kindlmann. Superquadric tensor glyphs. In *Proceedings IEEE TVCG/EG Symposium on Visualization 2004*, page (accepted), May 2004.

[20] G. L. Kindlmann and D. M. Weinstein. Hue-balls and lit-tensors for direct volume rendering of diffusion tensor fields. In *IEEE Visualization '99*, pages 183–190, 1999.

[21] G. W. Milligan and M. C. Cooper. A study of the comparability of external criteria for hierarchical cluster analysis. *Multivariate Behavioral Research*, 21:441–458, 1986.

[22] S. Mori and P. C. M. van Zijl. Fiber tracking: principles and practices - a technical review. *NMR Biomed*, 15:468–80, 2002.

[23] A. Vilanova, G. Berenschot, and C. van Pul. DTI visualization with streamsurfaces and evenly-spaced volume seeding. In *VisSym '04 Joint Eurographics - I.E.E.E. T.C.V.G. Symposium on Visualization, Conference Proceedings*, pages 173–182, 2004.

[24] S. Wakana, H. Jiang, L. M. Nagae-Poetscher, P. C. M. van Zijl, and S. Mori. Fiber tractbased atlas of human white matter anatomy. *Radiology*, 230:77–87, 2004.

[25] C. F. Westin, S. Peled, H. Gudbjartsson, R. Kikinis, and F. A. Jolesz. Geometrical diffusion measures for MRI from tensor basis analysis. In *ISMRM '97*, page 1742, Vancouver Canada, April 1997.

[26] S. Zhang, C. Demiralp, and D. H. Laidlaw. Visualizing diffusion tensor MR images using streamtubes and streamsurfaces. *IEEE Transactions on Visualization and Computer Graphics*, 9(4):454–462, October 2003.

[27] S. Zhang and D. H. Laidlaw. Hierarchical clustering of streamtubes. Technical Report CS-02-18, Brown University Computer Science Department, August 2002.

[28] S. Zhang and D. H. Laidlaw. DTI fiber clustering in the whole brain. IEEE Visualization 2004 Poster Compendium, October 2004.

[29] S. Zhang and D. H. Laidlaw. DTI fiber clustering and cross-subject cluster analysis. In *Proceedings of ISMRM*, Miami, FL, May 2005. inreview.

[30] X. Zheng and A. Pang. Volume deformation for tensor visualization. In *Proceedings of IEEE Visualization*, pages 379–386, 2002.

[31] X. Zheng and A. Pang. Topological lines in 3d tensor fields. In *IEEE TCVG Symposium on Visualization*, 2004.

Appendix A

Derivation of the normalized adjusted rand index

This appendix shows the derivation of the Normalized Adjusted Rand (NAR) index.

We start with the definition of the Adjusted Rand index in terms of the normalized contingency table:

$$\frac{a' - (m1'm2')/\binom{Rk}{2}}{(m1' + m2')/2 - (m1'm2')/\binom{Rk}{2}}.$$

Then, we substitute a', $m1'$, $m2'$:

$$\frac{\sum_{i=1}^{R}\sum_{j=1}^{S}\binom{k\frac{n_{ij}}{u_i}}{2} - \left(R\binom{k}{2}\sum_{j=1}^{S}\binom{\sum_{i=1}^{R}k\frac{n_{ij}}{u_i}}{2}\right)/\binom{Rk}{2}}{\left(R\binom{k}{2} + \sum_{j=1}^{S}\binom{\sum_{i=1}^{R}k\frac{n_{ij}}{u_i}}{2}\right)/2 - \left(R\binom{k}{2}\sum_{j=1}^{S}\binom{\sum_{i=1}^{R}k\frac{n_{ij}}{u_i}}{2}\right)/\binom{Rk}{2}}.$$

Next, we write out the binomials and simplify the result to the form:

$$\frac{kx + k^2x_2 + k^3x_3}{ky + k^2y_2 + k^3y_3}$$

with

$$
\begin{aligned}
x_1 &= -2\sum_{j=1}^{S}\sum_{i=1}^{R}\frac{n_{ij}}{u_i} - \sum_{i=1}^{R}\sum_{j=1}^{S}\frac{n_{ij}}{2u_i} \\
x_2 &= 2\sum_{j=1}^{S}\sum_{i=1}^{R}\frac{n_{ij}}{u_i} + 2\sum_{j=1}^{S}\left(\sum_{i=1}^{R}\frac{n_{ij}}{u_i}\right)^2 - 4\sum_{i=1}^{R}\sum_{j=1}^{S}\frac{n_{ij}^2}{2u_i^2} \\
x_3 &= -2\sum_{j=1}^{S}\left(\sum_{i=1}^{R}\frac{n_{ij}}{u_i}\right)^2 + 2R\sum_{i=1}^{R}\sum_{j=1}^{S}\frac{n_{ij}^2}{u_i^2} \\
y_1 &= R - \sum_{j=1}^{S}\sum_{i=1}^{R}\frac{n_{ij}}{u_i} \\
y_2 &= -R - R^2 + (2-R)\sum_{j=1}^{S}\sum_{i=1}^{R}\frac{n_{ij}}{u_i} + \sum_{j=1}^{S}\left(\sum_{i=1}^{R}\frac{n_{ij}}{u_i}\right)^2 \\
y_3 &= R^2 + (-2+R)\sum_{j=1}^{S}\left(\sum_{i=1}^{R}\frac{n_{ij}}{u_i}\right)^2 .
\end{aligned}
$$

Now, if we take k to infinity only the $\frac{x_3}{y_3}$ term remains:

$$
\frac{-2\sum_{j=1}^{S}\left(\sum_{i=1}^{R}\frac{n_{ij}}{u_i}\right)^2 + 2R\sum_{i=1}^{R}\sum_{j=1}^{S}\frac{n_{ij}^2}{u_i^2}}{R^2 + (-2+R)\sum_{j=1}^{S}\left(\sum_{i=1}^{R}\frac{n_{ij}}{u_i}\right)^2}.
$$

This can be rewritten as:

$$
\frac{2f - 2Rg}{(-2+R)f + R^2}
$$

with

$$f = \sum_{j=1}^{S} \left(\sum_{i=1}^{R} \frac{n_{ij}}{u_i} \right)^2$$

$$g = \sum_{i=1}^{R} \sum_{j=1}^{S} \frac{n_{ij}^2}{u_i^2}.$$

Appendix B

Derivation of the weighted normalized adjusted rand index

This appendix shows the derivation of the Weighted Normalized Adjusted Rand (WNAR) index.

The WNAR index is the Weighted Normalized Rand (WNR) index adjusted for chance agreement:

$$\frac{WNR - E(WNR)}{1 - E(WNR)}.$$

First, we substitute WNR and $E(WNR)$:

$$\frac{\left(1 - \frac{b'}{M'} - \frac{c'}{M'}\right) - \left(1 - 2(1-\alpha)\frac{m1'(M'-m2')}{M'^2} - 2\alpha\frac{m2'(M'-m1')}{M'^2}\right)}{1 - \left(1 - 2(1-\alpha)\frac{m1'(M'-m2')}{M'^2} - 2\alpha\frac{m2'(M'-m1')}{M'^2}\right)}.$$

Then, we substitute b', c', $m1'$, $m2'$ and M' and write out the binomials of the result:

$$\frac{-2\left(-(-1+k)\sum_{j=1}^{S}\frac{1}{2}\left(-1+\sum_{i=1}^{R}\frac{kn_{ij}}{u_i}\right)\sum_{i=1}^{R}\frac{kn_{ij}}{u_i} + (-1+kR)\sum_{i=1}^{R}\sum_{j=1}^{S}\frac{kn_{ij}\left(-1+\frac{kn_{ij}}{u_i}\right)}{2u_i}\right)}{(-1+k)\,kR\,(-1+kR)\,(-1+\alpha) + 2\,(-1+k+\alpha-kR\alpha)\sum_{j=1}^{S}\frac{1}{2}\left(-1+\sum_{i=1}^{R}\frac{kn_{ij}}{u_i}\right)\sum_{i=1}^{R}\frac{kn_{ij}}{u_i}}.$$

Next, we simplify the last equation to the form:

$$\frac{kx + k^2 x_2 + k^3 x_3}{ky + k^2 y_2 + k^3 y_3}$$

with

$$\begin{aligned}
x_1 &= \sum_{j=1}^{S}\sum_{i=1}^{R}\frac{n_{ij}}{u_i} + 2\sum_{i=1}^{R}\sum_{j=1}^{S}\frac{n_{ij}}{2u_i}\\
x_2 &= -\sum_{j=1}^{S}\sum_{i=1}^{R}\frac{n_{ij}}{u_i} - \sum_{j=1}^{S}\left(\sum_{i=1}^{R}\frac{n_{ij}}{u_i}\right)^2 + \sum_{i=1}^{R}\sum_{j=1}^{S}\frac{n_{ij}^2}{u_i^2}\\
x_3 &= \sum_{j=1}^{S}\left(\sum_{i=1}^{R}\frac{n_{ij}}{u_i}\right)^2 - R\sum_{i=1}^{R}\sum_{j=1}^{S}\frac{n_{ij}^2}{u_i^2}\\
y_1 &= -R + R\alpha + \sum_{j=1}^{S}\sum_{i=1}^{R}\frac{n_{ij}}{u_i} - \alpha\sum_{j=1}^{S}\sum_{i=1}^{R}\frac{n_{ij}}{u_i}\\
y_2 &= R + R^2 - R\alpha - R^2\alpha - (1 - R\alpha)\sum_{j=1}^{S}\sum_{i=1}^{R}\frac{n_{ij}}{u_i} - (1-\alpha)\sum_{j=1}^{S}\left(\sum_{i=1}^{R}\frac{n_{ij}}{u_i}\right)^2\\
y_3 &= -R^2 + R^2\alpha + (1 - R\alpha)\ \sum_{j=1}^{S}\left(\sum_{i=1}^{R}\frac{n_{ij}}{u_i}\right)^2 .
\end{aligned}$$

Now, if we take k to infinity only the $\frac{x_3}{y_3}$ term remains:

$$\frac{\sum_{j=1}^{S}\left(\sum_{i=1}^{R}\frac{n_{ij}}{u_i}\right)^2 - 2R\sum_{i=1}^{R}\sum_{j=1}^{S}\frac{n_{ij}^2}{2u_i^2}}{-R^2 + R^2\alpha + (1 - R\alpha)\ \sum_{j=1}^{S}\left(\sum_{i=1}^{R}\frac{n_{ij}}{u_i}\right)^2}.$$

This can be rewritten as:

$$\frac{f - Rg}{(1 - R\alpha)\, f - R^2 + R^2\alpha}$$

with

$$f = \sum_{j=1}^{S} \left(\sum_{i=1}^{R} \frac{n_{ij}}{u_i} \right)^2$$

$$g = \sum_{i=1}^{R} \sum_{j=1}^{S} \frac{n_{ij}^2}{u_i^2}.$$

Appendix C

Implementation

This appendix shows the design of the most important classes that were implemented in the DTI Tool.

The DTI Tool was originally created by Berenschot [2] using the visualization toolkit (VTK). VTK is an open source library of C++ classes that can be used to visualize all kinds of data. Data is processed by building a pipeline of filters that create or modify the data.

The filter responsible for creating fibers is CStreamline[1], which is a subclass of vtkPolyDataToPolyDataFilter, which in turn is a standard VTK class for processing polygon data. Each fiber is represented as an ordered list of 3D points. Figure C.1 shows how the fibers, originating from the CStreamline class, flow through the new filters that were built for classification, clustering and validation. Figure C.2 shows the inheritance diagram of these new classes. Note that the CStreamline class was already part of the DTI Tool.

Here is an description of the classes that we added to the DTI Tool:

- CClassifyFiberFilter classifies the fibers according through which regions (ROI's) they pass. The regions, which are represented as 2D polygons, are loaded from a file. The bundle id's are added as attributes to the fibers.

- CClusterFilter clusters the fibers into groups. A distance matrix is built by using a certain proximity function. CClusterFilter is abstract; subclasses provide the actual implementations of the clustering algorithms:

  - CHierarchicalClusterFilter implements the hierarchical clustering algorithm. The number of clusters is passed as a parameter.

[1] In VTK, fibers are called streamlines.

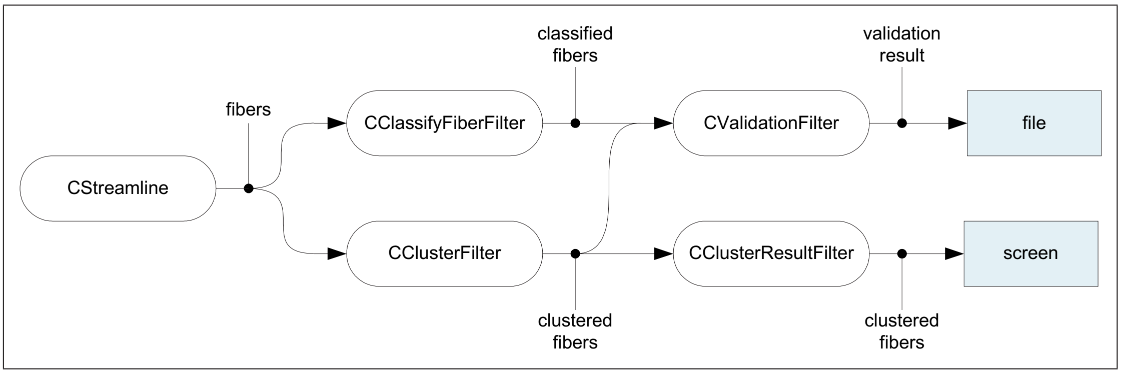


Figure C.1: Dataflow diagram.

  - CSharedNearestNeighborClusterFilter implements the shared nearest neighbors algorithm. The parameters are the number of neighbors and the edge threshold.

- CClusterResultFilter receives the fiber clusters and prepares them for visualization. Preparation includes selection and coloring using a look-up-table. Individual clusters can be selected based on properties such as size or cluster id.

- CValidationFilter class compares two partitions of fibers. Subclasses provide an actual implementation of the validation algorithms:

  - CExternalIndexFilter calculates the values of the various external indices.
  - CROCCurveFilter produces output from which ROC Curves can be drawn.

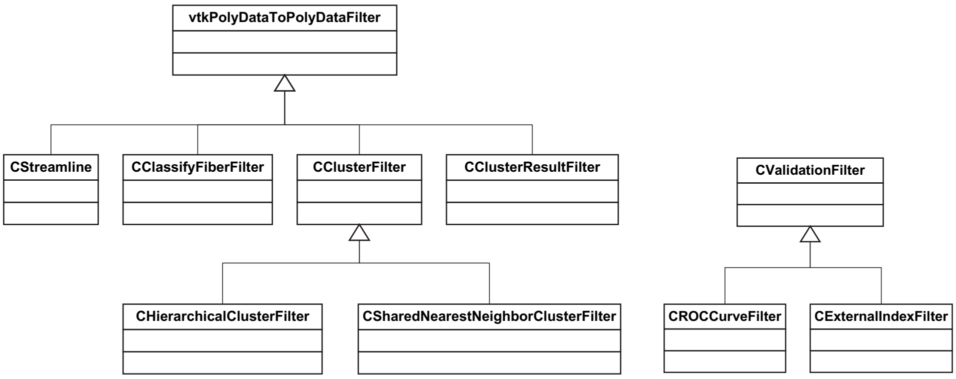


Figure C.2: Inheritance diagrams.