DTI visualization with streamsurfaces and evenly-spaced volume seeding

Vilanova Bartroli, A.; Berenschot, G.; van Pul, C.

Published in:
Proceedings of the Joint Eurographics - IEEE TCVG Symposium on Visualization (VisSym04), 19-21 May 2004, Konstanz, Germany

Published: 01/01/2004

Document Version
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Citation for published version (APA):
Vilanova, A., Berenschot, G., & Pul, van, C. (2004). DTI visualization with streamsurfaces and evenly-spaced volume seeding. In O. Deussen, C. Hanssen, & D. A. Keim (Eds.), Proceedings of the Joint Eurographics - IEEE TCVG Symposium on Visualization (VisSym04), 19-21 May 2004, Konstanz, Germany. (pp. 173-182). Germany, Konstanz: The Eurographics Association.

Joint EUROGRAPHICS - IEEE TCVG Symposium on Visualization (2004)
O. Deussen, C. Hansen, D.A. Keim, D. Saupe (Editors)

DTI Visualization with Streamsurfaces and Evenly-Spaced Volume Seeding

A. Vilanova[1†] , G. Berenschot[1] and C. van Pul[23]

[1] Department of Biomedical Engineering [2]Department of Applied Physics
Technische Universiteit Eindhoven
[3] Maxima Medical Center, Veldhoven

Abstract

Experimental evidence has shown that water diffusion is anisotropic in organized tissues such as white matter or muscles. Diffusion Tensor Imaging is a non-invasive MR technique that measures water diffusion. DTI is used to visualize linear structures such as fibers. In this paper, we present a visualization tool for DTI data. A new algorithm to visualize linear structures in areas of crossing or converging fibers is presented. Usually the user defines an area from where the fibers are generated. In this way, the user can miss part of the information, if the area is not correctly defined. We present a method to visualize the structures in the whole volume with an evenly-spaced distance between them. Some results obtained by our partners using the DTI tool will be presented.

Categories and Subject Descriptors (according to ACM CCS): J.3 [Comp. Appl.]: Medical information systems

1. Introduction

Magnetic Resonance (MR) Diffusion Tensor Imaging (DTI) is a recent MR acquisition technique that measures the random motion of water (i.e., diffusion) in tissue [BP96]. Experimental evidence has shown that the water diffusion is anisotropic in organized tissues such as white matter or muscles. In white matter, the anisotropy is caused by the parallel organization of the axons and diffusion is restricted perpendicular to the fibers by membranes, and also by myelin. The diffusion anisotropy gives an indication of the underlying structure of the tissue. DTI is used as a research tool to study, for example, brain strokes, multiple sclerosis, the muscle structure and the development of the brain.

The tool presented in this paper is used at the Maxima Medical Center in Veldhoven (MMC) to study neonates that suffer from hypoxic ischemic brain damage. These neonates have brain injuries caused by a lack of oxygen and nutrients, due to problems with the blood flow. DTI provides information about damaged regions, and the structure and the development of the neonatal brain. In the developing brain, the axons are in the process of becoming myelinated. Therefore, the anisotropy depends on the development phase. The BioMedical NMR group at Eindhoven University of Technology (TU/e) uses this tool for animal studies and studies of the functional properties of muscles. It is proven that the spa-

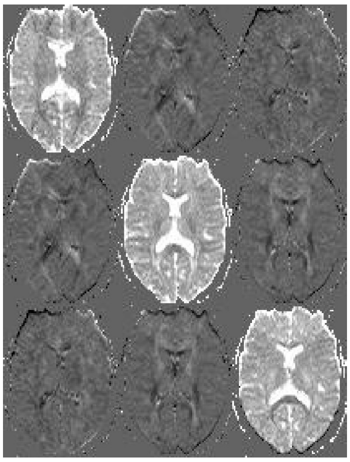


Figure 1: *The components of the diffusion tensor of a transversal brain slice from a healthy person.*

[†] email:a.vilanova@tue.nl, http://www.bmi2.bmt.tue.nl

tial structure of muscle tissue strongly determines its functional properties (e.g., the distribution of stresses and strains largely depends on the orientation of muscle fibers).

Diffusion is represented by a second order tensor (i.e., a 3x3 symmetrical matrix). The visualization of tensor data is a difficult task. Visualizing the elements of the tensors separately gives little useful information (see figure 1). Several techniques have been presented in the last years to visualize DTI data. Most of these techniques either require a simplification of the tensor field to one scalar field or vector field, or just the local information is extracted.

The most investigated technique is the so-called streamline tracing or fiber tracking. This technique simplifies the DTI data to a vector field of the main anisotropy direction. Then particle paths in this vector field are traced. The paths are associated with the linear structures in the DTI data (e.g., fibers). However, in areas where the main diffusion direction is not defined, the streamline tracing fails. For example, when two fiber bundles cross, the diffusion is restricted to the plane defined by the two bundles. Any vector in the plane is a main anisotropy direction. In this paper, we present a new method that extends streamline tracing to surface tracing when the diffusion is restricted to a plane.

Another problem is the initialization of the particle positions (i.e., seed points). Usually the seed points are defined by the user. Therefore, some information can be missed when abnormalities are present in the data. An extension of the seeding technique presented by Jobard et al. [JL97] has been done in order to show the information in the data without the need of manual seeding.

This paper is organized as follows. In the next section, a short explanation of DTI data acquisition is presented. In section 3, the previous work in DTI visualization and their current open problems are described. Section 4 presents the surface tracing. In section 5, we present the extensions to Jobard et al.'s algorithm [JL97] . In section 6, a short explanation of the DTI tool is presented. Finally, we present experimental results with different data sets, and conclusions and future work.

2. DTI data

Diffusion is the result of random motion of molecules that is driven by internal thermal energy (i.e., Brownian motion). The mobility of the molecules can be characterized by a physical constant, the intrinsic diffusion coefficient, D, with unit m^2/s. The structure of living tissue poses physical barriers (e.g., membranes) and restricts diffusion. Therefore, the diffusion is a directional-dependent quantity. This diffusion is usually represented by a second order tensor **D**.

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

D is a real Hermitian positive definite matrix. This means that the matrix is symmetric and its eigenvalues are real and positive. Eigenanalysis is usually applied to the tensor in order to find the principal diffusivity directions (i.e., eigenvectors $\vec{e_i}$) and the corresponding diffusion coefficients (i.e., eigenvalues λ_i). In the remainder of the paper, we assume that the eigenvalues and eigenverctors are ordered as $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$.

Trilinear interpolation on each component of **D** is used to reconstruct the continuous tensor field of the DTI data. It is an open problem which interpolation technique in tensor data is most adequate (e.g., componentwise or eigenvector and eigenvalues). Image processing techniques for noise removal in DTI are also subject of current research (e.g., Zhukov et al. [ZB02]). These topics will not be addressed in this paper. It is assumed that these techniques could be applied as preprocessing step to our data. In the next section, we will describe different methods that have been used to visualize the information of DTI data.

3. Background and previous work

Visualization of high dimensional data such as tensor data is a challenging problem. Several techniques exist that have been presented in the previous years to visualize second order tensor data, and to visualize diffusion tensor imaging data in particular. We have classified the techniques in three categories: anisotropy index mapping, tensor glyphs and vector field visualizations.

3.1. Anisotropy index mapping

This group of techniques consists of simplifying the six dimensional data to scalar metrics that are usually called anisotropy indices. These indices are invariant under rotation and scaling and they give information about the anisotropy of the diffusion tensor. The most commonly used indices are the fractional anisotropy (FA) and the relative anisotropy (RA) (see Bihan et al. [LBMP*01]).

Westin et al. [WMK*99] divide diffusion anisotropy into three basic geometric shapes depending on the rank of the tensor (see figure 2). In the linear case, diffusion is mainly in the direction corresponding to the largest eigenvalue (i.e., $\lambda_1 \gg \lambda_2 \approx \lambda_3$). In the planar case, diffusion is restricted to a plane spanned by the two eigenvectors corresponding to the two largest eigenvalues (i.e., $\lambda_1 \approx \lambda_2 \gg \lambda_3$). In the spherical case, there is isotropic diffusion (i.e., $\lambda_1 \approx \lambda_2 \approx \lambda_3$). According to this, the linear C_l, planar C_p and isotropic C_s indices are defined by

$$C_l = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3} \quad C_p = \frac{2(\lambda_2 - \lambda_3)}{\lambda_1 + \lambda_2 + \lambda_3} \tag{1}$$
$$C_s = \frac{3\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \quad C_s + C_p + C_l = 1$$

These indices fall within the range $[0,1]$ and form barycentric coordinates.

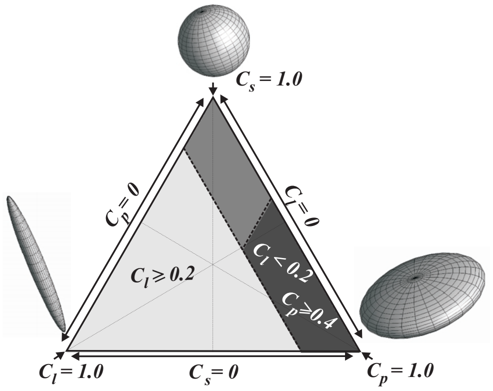


Figure 2: *Barycentric space defined by the anisotropy indices C_l, C_p and C_s and the corresponding shapes.*

3.2. Tensor glyphs

Simplifying the tensor to a scalar shows just part of the information contained in the tensor. A way to show the six dimensional information of second order tensor field is by using glyphs or icons. In the case of DTI, the use of an ellipsoid as glyph is the most intuitive representation [PB96]. The axes of the ellipsoid correspond to the eigenvectors and the lengths of the axes correspond to the eigenvalues. Worth et al. [WMWJ*98] used cuboids determined by the eigenvalues and eigenvectors. The advantage of this method compared to the ellipsoids is that it is easier to see the direction of the two less significant eigenvectors.

These methods have the advantage that they visualize all or most of the information of the tensor. However it is difficult to extract global information from them (e.g., fiber bundles). Furthermore, if the glyphs are visualized in 3D, the image becomes overloaded and cluttered.

3.3. Vector field visualization

The tensor field can also be simplified to a vector field defined by the main eigenvector, $\vec{e}_1$. This simplification is based on the assumption that in the areas of linear anisotropy, $\vec{e}_1$ defines the direction of linear structures. The sign of $\vec{e}_1$ has no meaning.

One commonly used method to visualize DTI is to map $\vec{e}_1$ to color, e.g., directly using the $\vec{e}_1$ components for the RGB channel. This color is weighted by an anisotropy index to reduce the visualization of isotropic areas. Instead of $\vec{e}_1$, Kindlmann et al. [KW99] use a user-defined vector multiplied by the tensor. The difference between the user defined and the resulting vector gives more information about the anisotropy of the tensor.

Other methods have been proposed to visualize the global information of 2D as well as 3D vector fields [PVH*02]. There are well established 2D vector-fields visualization methods, see van Wijk [vW02]. Although 2D techniques have been extended to 3D, the visualization of 3D vector fields is still an unsolved problem due to the extra challenge of the cluttering and the computational costs.

DTI data is essentially 3D. The most commonly used technique to visualize DTI data is streamline tracing, also called fiber tracking [MvZ02]. Streamlines in 3D can be easily visualized for regions which avoids the problems of cluttering. Furthermore there is a direct analogy between the streamlines and the fibers. The streamline tracing is based on solving the following equation:

$$p(t) = \int_0^t \vec{v}(s)ds \qquad (2)$$

where $p(t)$ is the generated streamline and $\vec{v}$ corresponds to the vector field generated from $\vec{e}_1$.

The streamline techniques have three main steps: definition of initial tracking points (i.e., seed points), integration, and the definition of stopping criteria.

Seed points are usually defined by the user who specifies one or more regions of interest (ROI). The interior of the ROI is sampled and the samples are used as seed points.

Numerical integration is used to solve equation 2. Several numerical schemes can be used, such as Euler forward, and second or fourth order Runge-Kutta. In this paper, second order Runge-Kutta is used.

Stopping criteria avoid that the streamline is being traced in areas where the vector field is not robustly defined. In areas of isotropic or planar diffusion, the value of $\vec{e}_1$ can be considered random. Therefore, $\vec{e}_1$ has no meaning concerning the underlying structure. The user usually can define a threshold based on the anisotropy indices (i.e., FA, RA or C_l) to define the areas where the vector field is defined. The value of this threshold depends on the data acquisition protocol and the nature of the object that is being scanned. Other criteria are also being used such as the curviness or length of the streamline.

An extension to streamlines for second order tensor fields are hyperstreamlines [DH93]. Hyperstreamlines use all eigenvalues and eigenvectors. A streamline defines the axis of a generalized cylinder whose cross-section perpendicular to the axis is an ellipse defined by $\vec{e}_2$ and $\vec{e}_3$, and λ_2 and λ_3 respectively.

Streamline tracing techniques for DTI have several problems. Linear structures are also present in areas with nonlinear diffusion (see figure 3). Most of the DTI tracing algorithms just consider the areas where the vector field is defined robustly. Several authors proposed methods to trace within areas of isotropic or planar diffusion following the most probable diffusion direction (e.g., Weinstein et al. [WKL99]). These methods use the incoming direction of the streamline together with the information of the tensor to determine the next tracing point according to some heuristics.

Another problem of streamlines is seeding. The seed points are usually defined by the user. In a healthy person where the anatomy is known, the users can reasonably guess where the interesting bundles are. However, in patients, there is no real clue about the possible underlying structure and the seeding by the user can miss important structures. On the other hand, defining the seed points to cover the whole volume can be computationally expensive if done carelessly. Furthermore, the image gets cluttered and therefore it is difficult to extract useful information.

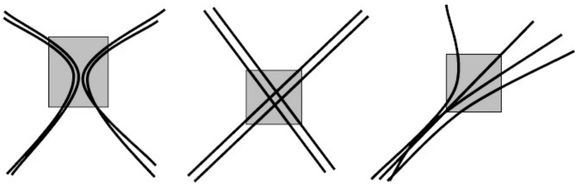


Figure 3: *Examples of regions where fibers are present but the diffusion tensor is planar: (left) kissing fibers, (middle) two fiber bundles crossing and (right) divergence/convergence of fibers. Regions that are marked grey will have a tensor with planar diffusion.*

4. Streamsurface tracing

Linear structures can also be present in areas with non-linear anisotropy, for example, due to crossing fibers. The anisotropy will be mainly planar when two fiber bundles are crossing and fibers converge or diverge (see figure 3). Figure 3 shows that the tensor information is not enough to solve the underlying linear structure. Our approach is based on reconstructing the structure defined by the tensor. It is left up to the user to decide which would be the most probable underlying linear structure.

Planar anisotropy indicates planar structure and therefore a surface is traced. In our method, when a streamline enters an area of planar anisotropy, a surface is traced until an isotropic or linearly-anisotropic area is found. A method for surface tracing similar to the one presented by Zhang et al. [ZDL03] is introduced in this section. The surfaces generated by the algorithm will be called streamsurfaces. Streamsurfaces are used to visualize linear structures in areas such as crossing fibers. The streamsurfaces will be topologically equivalent to a disk with holes.

Streamsurface tracing consists of the same main steps as streamline tracing: seeding, surface integration and stopping criteria. The seeding can be done as described in section 3.3.

In order to define the areas of linear and planar anisotropy, the barycentric-coordinates are used. Two thresholds will be defined T_{C_l} for C_l, and T_{C_p} for C_p. If $C_l \geq T_{C_l}$, then streamlines will be traced (see figure 2 where $C_l \geq T_{C_l} = 0.2$). Else if $C_p \geq T_{C_p}$ then surfaces will be traced (see figure 2 where $C_p \geq T_{C_p} = 0.4$ and $C_l < T_{C_l} = 0.2$). T_{C_l} and T_{C_p} are defined by the user. In the rest of this section we will describe how the surface generation is done.

Given a mesh in a plane, the algorithm will fit this mesh to the surface structure defined by the DTI data. The mesh will have the topology of a triangular mesh, and initially will consist of equilateral triangles. This is chosen in order to have a mesh as regular as possible. Each vertex p will have six neighbors at equal distance of p. These neighbors will form an hexagon around p (see figure 4c). The geometry of the mesh is an array G with point coordinates. The topology or connectivity of the mesh is stored in a table T. Each entry in T corresponds to the point at the same index position in G. Each entry has six table indices which correspond to the six possible neighbors or edges of the point. The neighbors are ordered in a clockwise order around the entry point. An edge appears several times in T. These redundancies are used to keep track of the already calculated points. A point can have an entry in the topology table T although no geometry is associated to it. The geometry, the indexing and the mesh topology are generated on the fly.

The description of the algorithm will be divided in two parts. First the algorithm for the indexing and the generation of the topology of the mesh will be described. Secondly, we present how the streamsurface points are calculated. In the implementation these two steps are combined, but they are separated here for clarity.

4.1. Topology generation

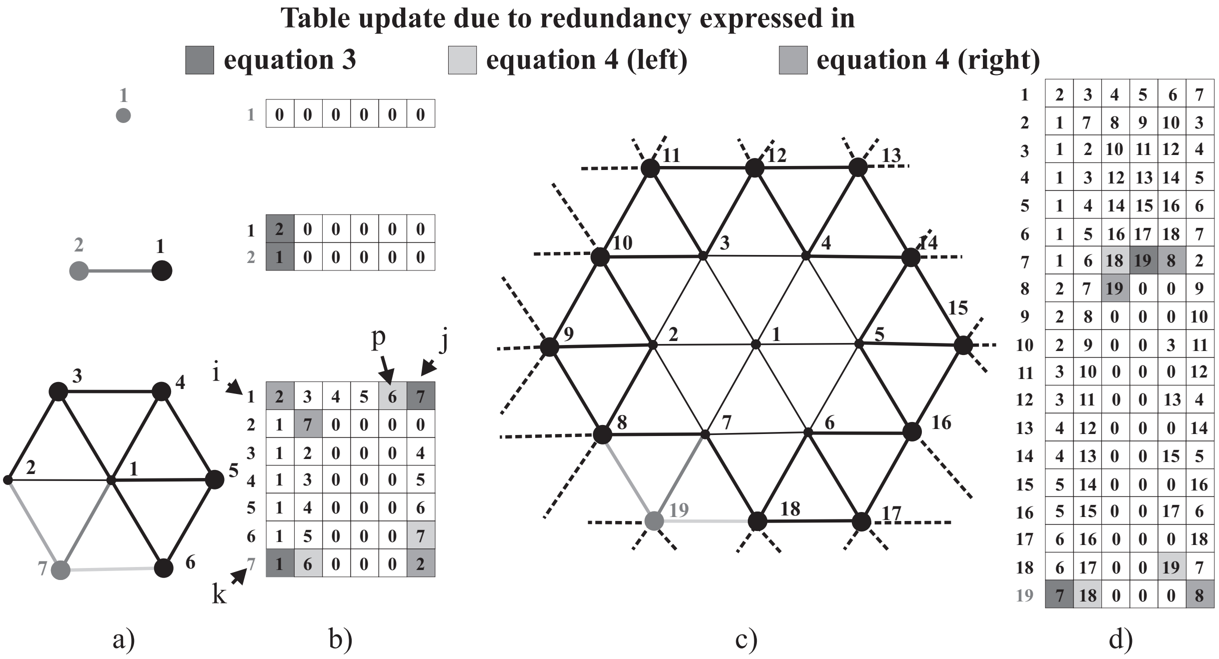
The indexing is chosen such that it forms a spiral around the first point, see figure 4c. The indices are also the order in which the points are being treated.

The first point that initiates the surface tracing gets index number one. Its entry in the topology table T gets zeros in all positions or neighbors. Zero indicates that the corresponding neighbor point has not been treated (see figure 4a and 4b). After the first initialization, the process is the same for all points. The first position with value 0 is searched in table T, sorted first by point index and then by neighbors. A new entry in table T is created for the new point, and T is updated to reflect the connectivity of this point. The redundancies in table T can be defined by the following conditions. All the additions and subtractions are done modulo 6. Variable i, k, and p are point indices of table T, and j, m, and $l \in [0,5]$ are the six neighbors of a point:

$$T[i,j] = k \wedge k \neq 0 \Rightarrow \exists m \;\; T[k,m] = i \tag{3}$$

$$\begin{aligned} T[i,j] \; &= k \wedge T[i,j-1] = p \wedge k \neq 0 \wedge p \neq 0 \Rightarrow \\ \Rightarrow \quad & \exists l \; (T[p,l] = i \wedge T[p,l-1] = k) \wedge \\ & \wedge \exists m \; (T[k,m] = i \wedge T[k,m+1] = p) \end{aligned} \tag{4}$$

Figure 4b and 4d show the status of table T with its corresponding initial mesh 4a and 4c after the insertion of the points 1,2,7 and 19. The updates of T to fulfill equations 3 and 4 are also illustrated. Notice that the topology of



1	0	0	0	0	0	0

1	2	0	0	0	0	0
2	1	0	0	0	0	0

1	2	3	4	5	6	7
2	1	7	0	0	0	0
3	1	2	0	0	0	4
4	1	3	0	0	0	5
5	1	4	0	0	0	6
6	1	5	0	0	0	7
7	1	6	0	0	0	2

1	2	3	4	5	6	7
2	1	7	8	9	10	3
3	1	2	10	11	12	4
4	1	3	12	13	14	5
5	1	4	14	15	16	6
6	1	5	16	17	18	7
7	1	6	18	19	8	2
8	2	7	19	0	0	9
9	2	8	0	0	0	10
10	2	9	0	0	3	11
11	3	10	0	0	0	12
12	3	11	0	0	13	4
13	4	12	0	0	0	14
14	4	13	0	0	15	5
15	5	14	0	0	0	16
16	5	15	0	0	17	6
17	6	16	0	0	0	18
18	6	17	0	0	19	7
19	7	18	0	0	0	8

Figure 4: *Table T and its corresponding meshes at different iterations, inserting point 1,2,7, and 19.*

the mesh is independent of the geometry. It could be precalculated if the dimensions of the mesh would be known.

4.2. Geometry generation

In this section, we describe how the mesh is fitted to the streamsurface. In a point p with planar anisotropy, the two major eigenvectors define a plane. This means that Brownian motion of molecules is mainly restricted to movements within the plane. This plane can be seen as the tangent plane at point p of a streamsurface $S(u,v)$. This surface represents all the equally possible paths that a molecule could follow. A curve in this surface is defined by $C(t) = S(f(t), g(t))$, where $f(t)$ and $g(t)$ can be any differentiable function. Extension of equation 2 to surface curves can be done by:

$$C(t) = \int_0^t \vec{v_c}(s)ds \qquad (5)$$

$$\vec{v_c}(t) = f'(t)\vec{S_u}(f(t),g(t)) + g'(t)\vec{S_v}(f(t),g(t))$$

$\vec{S_u}$ and $\vec{S_v}$ are the partial derivatives of S. $\vec{S_u}$ and $\vec{S_v}$ are two vectors in the tangent plane. Therefore $\vec{v_c}(t)$ is also a vector in the tangent plane. This is valid for any differentiable parametrization of the surface.

There is no need to find a parametrization of the surface to find the surface itself. Therefore, $\vec{v_c}(t)$ can be defined implicitly by ensuring that $\vec{v_c}(t)$ is on the tangent plane of the surface at point $C(t)$. In a plane, we have an infinite number of vectors that can be chosen. A criterion has to be applied to choose one $\vec{v_c}(t)$. Using $\vec{v_c}(t)$, equation 5 is solved numerically by using second order Runge-Kutta. The criterion used to define $\vec{v_c}(t)$ will be by minimizing the angle between two consecutive vectors. This will be a local curvature minimization. $\vec{v_c}(t+\Delta t)$ is the projection of $\vec{v_c}(t)$ to the tangent plane at $C(t+\Delta t)$. In this way $\vec{v_c}(t+\Delta t)$ will be the vector in the tangent plane at $C(t+\Delta t)$ with the smallest angle with $\vec{v_c}(t)$. The pseudocode of the geometry calculation would be as follows assuming that T has already been generated:

```
proc CalculateGeometry(
     in : Topology Table $T$,
     in : Point $seed$,
  in out: Queue seed points $SeedPoints$,
     out: Geometry Table $G$ )
  integer $i$:= 1
  Queue $PointsToTreat$
  Vector Array $\vec{V_c}$
  Point $p$
  $G$[1]:= $seed$
  Queue 1 in $PointsToTreat$
  while $PointsToTreat$ is not empty do
    $i$:= Get and delete head $PointsToTreat$
    $\vec{V_c}$ := vectors in the tangent plane at $G$[$i$]
    for each $j \in [0,5]$ and $T$[$i,j$] not computed do
       $p$ := Integrate $C(t)$ given $G$[$i$] and $\vec{V_c}$[$j$]
       $G$[$T$[$i,j$]]:= $p$
       if$p$ has planar anisotropy
         Queue $T$[$i,j$] in $PointsToTreat$
       else if $p$ has linear anisotropy
         Queue $p$ in $SeedPoints$
       end
    end
  end
```

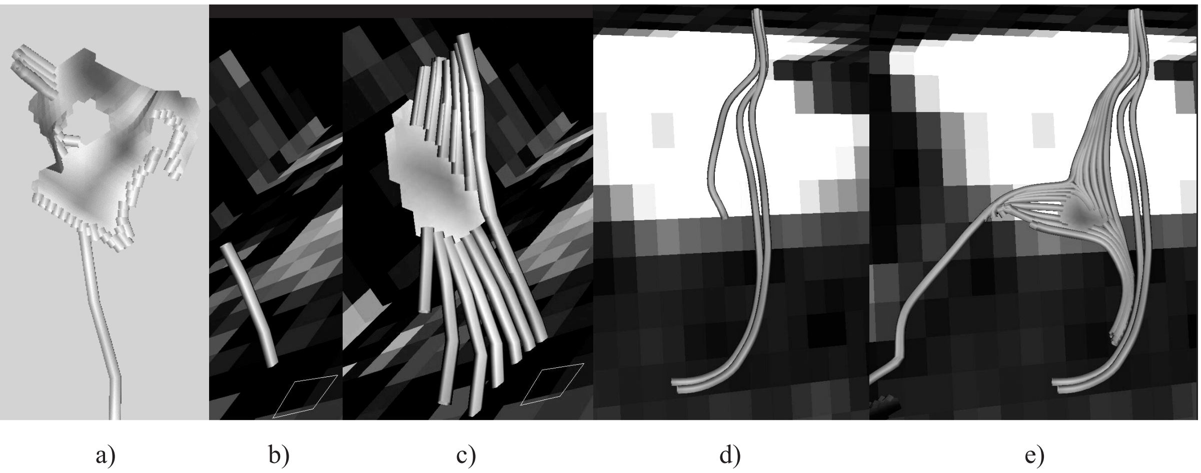


Figure 5: *Streamlines and streamsurfaces traced in a healthy adult brain data set with* $T_{C_l} = 0.2$ *and* $T_{C_p} = 0.4$*:* ***a)*** *streamsurface with a hole generated at the end of a streamline given one seed point.* ***b)*** *One streamline traced given a seed point* ***c)*** *the same seed point as* ***b*** *but tracing streamsurfaces and showing the possible prolongation of the fiber bundle* ***d)*** *streamlines using three seed points* ***e)*** *the same seed points as* ***d*** *with tracing streamsurfaces and showing the possible prolongation of the fiber bundle.*

The points with planar anisotropy, whose neighbors' geometry still has to be calculated, are in queue *PointsToTreat*. Position 1 in table G gets the value of the seed point, p. Index 1 is queued in *PointsToTreat*. The loop of the algorithm will go on until no points are left in *PointsToTreat*. The first element of *PointsToTreat* is assigned to i and deleted from the queue.

Each point i has six possible neighbors or edges $j \in [0,5]$. For each unknown neighbor a curve $C[j](t)$ is integrated until a defined edge length L is achieved. The value of $C[j](L)$ corresponds to the geometry, p, of point $T[i,j]$. The initial position of each edge is $C[j](0) = G[i]$. The corresponding initial direction of the edge $\vec{v}_c[j](0)$ needs to be calculated. Six initial directions are defined, $\vec{V}_c[j]$, in the tangent plane at $G[i]$. The angle between any two consecutive vectors in $\vec{V}_c[j]$ is define as constant as possible. The points whose geometry is known will have at least one neighbor whose geometry is also known. The initial vectors, $\vec{V}_c[j]$, of the unknown neighbors will be ordered following clockwise order from the already known ones.

If p has planar anisotropy, then $T[i,j]$ is queued in *PointsToTreat*. If p has linear anisotropy, p is added to the seed points to be treated. Notice that there are points which will be in the topology but will not have geometry assigned, so points which have not been computed. These points can be inside the surface, and will appear as holes (see figure 5a).

In this way, we fit the initial mesh of equilateral triangles to the surface. The mesh that is finally rendered, is generated by going through the table T and just generating those triangles between neighbors that have been computed and whose indices are bigger than the current index of the table. In this way the redundancies of T are ignored in the final mesh.

The presented algorithm will result in a surface which is topologically equivalent to a disk with holes. In the algorithm described by Zhang et al. [ZDL03] the generated surface is topologically equivalent to a disk, and holes are not taken into account although they can often be present in the data. This leads to infinite cycles in Zhang et al.'s algorithm, if no extra tests (e.g., distance to other surface points) and processing is done. In our algorithm, $\vec{v}_c(t)$ is defined such that the local curvature of $C(t)$ is minimized, but not constrained to a planar curve.

4.3. Streamline and streamsurface tracing

Differently from Zhang et al. [ZDL03], the streamsurfaces are combined with streamlines and not calculated independently. A queue with seed points, *SeedPoints*, is used. If during streamsurface tracing new streamline seed points are found, they are queued to *SeedPoints*. Such a seed point can generate a new streamline that can end again in a planar anisotropy point. This point with planar anisotropy will be queued to *SeedPoints*, and will initialize a surface tracing in another iteration. This procedure stops when *SeedPoints* is empty.

Following the previous explanation, the same streamsurface can be traced more than once, producing an infinite loop. In order to avoid this, a minimal distance d_s between streamsurfaces is defined. This distance indicates how close two streamsurfaces must be to consider them to be equal. A seed point for streamsurface tracing will not be valid if the distance to any already generated surface point is smaller than d_s. A more detailed description of how this distance is computed is given in section 5.

As said before, the sign of the eigenvectors of a tensor has no meaning. We always define the sign such that it is coherent with the incoming structure. Therefore, a streamline will not turn more than 90 degrees. An angle α can be defined by the user. If a surface is initialized by an incoming streamline, the streamlines, that start at the streamsurface which have an angle with the incoming streamline smaller or equal than α will be traced.

Figure 5 shows several examples of tracing streamlines with and without streamsurfaces. It shows that streamlines without streamsurfaces can underestimate the length of the fibers and miss part of the information of the fiber structure.

5. Evenly-spaced volume seeding

A user-defined seeding is a biased way to show the data. If the user does not define a good ROI some interesting structures might be missed. This becomes really a problem in cases where the underlying structure is not well known. Zhang et al. [ZDL03] propose to regularly sample the volume, and use each sample as seed point. Afterwards, a selection is made according to a distance criterion. The sampling distance should be at least in the order of the voxel distance. It is obvious that this can be extremely computationally expensive.

Our approach is based on the seeding method presented by Jobard et al. [JL97]. Their paper concentrates on streamline placement for visualizing 2D steady flow. We extend the method to 3D and to include streamsurfaces. Initially we will introduce the algorithm for streamlines and afterwards for streamsurfaces.

The goal of our method is to produce long and evenly-spaced streamlines in a single pass. The user defines how dense the visualization of the streamlines must be. This is expressed by the minimal distance between streamlines d_l.

The algorithm is as follows. An initial streamline is traced. Then a seed point at minimal distance d_l from all existing streamlines is chosen for the next streamline. Then this streamline is integrated until it goes too close to another streamline or it stops due to the stopping criteria. The process is continued until no more seed points are found.

5.1. Distance test

The density of the DTI visualization is defined by d_l. The density is controlled by not allowing that any point is used as seed point, if it is closer than d_l to any other streamline. The computations will be just approximated by not calculating the distance to the streamline itself, but to the sampled points of the streamlines. This approximation is valid, if the distance between neighboring sample points of the streamline is smaller than d_l

A distance volume is created to accelerate the distance test. The DTI domain is divided in cells of dimensions d_l. Each cell contains a list of the streamline points that are inside the cell. For each sample point it is checked whether it is at a distance larger than d_l. This is done by identifying the cell of the sample point in the distance volume, and checking the distance between itself and the points contained in its own cell and neighboring cells.

Like Jobard et al., a distance d_c smaller than d_l is defined by the minimal distance allowed between two streamlines, while d_l is used for the seed points. This produces longer streamlines which gives a better visual effect. In figure 6a and b the effect of using d_c is shown with $d_c = \frac{1}{2}d_l$.

If the streamlines are shown as tubes, the thickness of the tube can be made dependent on the distance to the other streamlines. This attenuates that the distance between streamlines is not constant due to using d_c. The distance d is the smallest distance of the streamline point with the other streamlines, and the maximum radius of the tube is r. The thickness of the tube at a position in the streamline is $\frac{d-d_c}{d_l-d_c}r$. This will be applied if $d < d_l$. This is also useful to identify streamlines which are stopped due to their distance to the other streamlines. Comparing figure 6b and 6c, the effect of using distance dependent tube thickness can be seen.

5.2. Seed selection

In order to fill the space correctly, it is necessary that the seed points are chosen with care. The algorithm starts tracing the streamline for the voxel with maximal linear anisotropy value. Once the streamline has been created, its points are added to the distance volume. The next seed point is calculated at distance d_l from the streamline. For each streamline point, six seed points are calculated. These points lie in the orthogonal plane of the streamline curve. The points define a regular hexagon of length d_l. The generated streamlines are queued in order of generation. Once all the possible seed points at distance d_l have been treated, the same procedure is done for the next streamline in the queue. Notice that a seed point will not be used if it is at a distance smaller than d_l from any streamline. This procedure continues until no streamline is left in the queue.

The vector field generated by $\vec{e}_1$ and the stopping criteria does not fill the whole domain of the DTI data set. Therefore, the voxel positions will also need to be considered for seeding. Once the queue of streamlines is empty, the next voxel position will be considered for seeding. If their distance to any existent streamline is larger than d_l, then a streamline will be traced and the procedure will start again. The process will end when no voxel position is left.

5.3. Evenly-spaced streamsurfaces

Streamsurfaces will be treated in a similar way. A distance d_s which indicates the minimum distance between surfaces is defined. A surface distance volume is generated whose

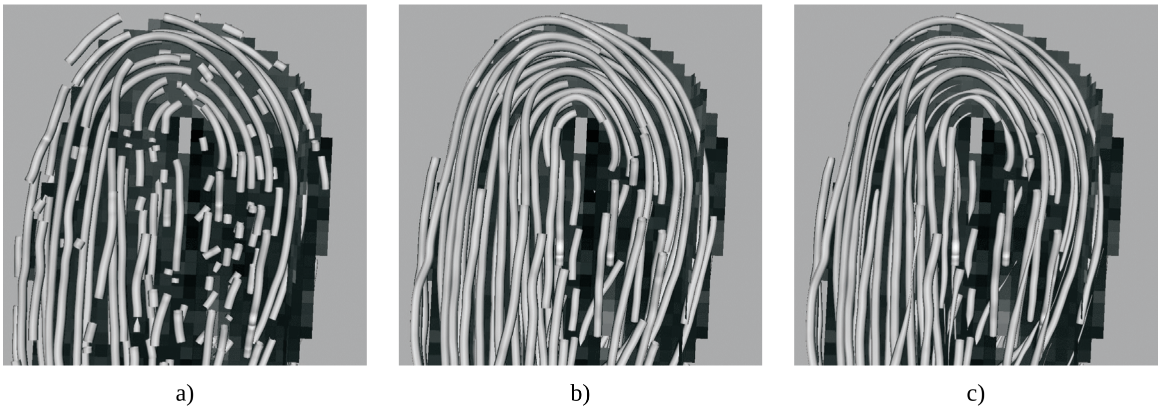


Figure 6: *Evenly-spaced volume seeding in a pig optic nerve (64x64x128) DTI data set. The pig optic nerve has been extracted and twisted before acquisition.* ***a)*** *d_c is not used,* ***b) a*** *$d_c = \frac{1}{2}d_l$,* ***c)*** *is as* ***b*** *but adding distance-dependent tube thickness.*

cell dimensions are d_s. The distance test will be done in the same way as with streamlines. The seed generation for streamsurfaces is also the same as with streamlines. The only difference is that given a surface point, two seed points will be generated, at $\pm d_s$ distance following the normal of the streamsurface at that surface point.

If the linear anisotropic domain is filled with streamlines, the use of surfaces is less useful to detect crossings or convergence/divergence of fibers. The connections can be deduced by the empty space between the streamlines. However, showing the planar structures of the DTI shows structural information of the data that might give further insight. Figure 9a (see color section) shows the evenly-spaced volume seeding in a healthy adult brain for streamlines and streamsurfaces.

6. DTI tool

A visualization tool for DTI data, called DTI tool, has been developed by us. This tool is implemented using C++, VTK and fltk. This tool has implemented the visualization techniques presented in this paper together with several of the methods described in section 3.

Anisotropy index mapping: Anisotropy indices, *FA*, *RA*, and the barycentric coordinates have been implemented. A hue color map is used to visualize the values of the indices.

Tensor glyphs A 2D ROI is defined by the user in a cutting plane. The ROI is sampled uniformly. Glyphs for all sampled points are shown. The shapes of the glyphs can be chosen between ellipsoids or cuboids.

Vector field visualization: The mapping of $\vec{e_1}$ to a continuous mapping from direction to color has been implemented, see the streamlines in figure 9(a-d)(see color section).

For global information, streamline tracing has been implemented. The resulting streamlines can be shown as lines, tubes or hyperstreamlines. The shape of the cross-section of the hyperstreamlines can be an ellipsoid, a quad or a cross. The last two allow to follow the rotation of $\vec{e_2}$ and $\vec{e_3}$ more easily.

Interactive seeding can be applied in different ways: single fiber, painting and ROI. Single fiber is based on a point defined by the user in an orthogonal plane, and a streamline is immediately traced at this point. If the user points to another position, the previous streamline is deleted and a new one is created at the current position. The painting interaction is the same as the single fiber case, but then the streamlines are not deleted. A 2D ROI can be defined in one of the cutting planes and then its interior is sampled uniformly. Each sample is used as seed point. Single fiber and painting interactions are meant for interactively exploring the data and getting fast feed-back. The ROI is used when the region to be inspected is already known.

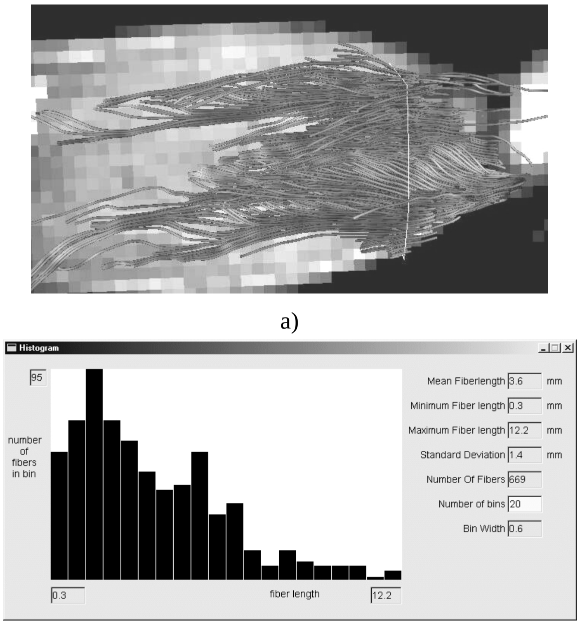
In the DTI tool, the evenly-spaced volume seeding presented in section 5 is also implemented. This method is used to get a global view of the distribution of the fibers. Due to cluttering, it is difficult to use it for a local or focussed study. Once an interesting region has been detected, other seeding methods such as ROI are used to avoid the cluttering.

Streamsurfaces are also an option for all types of seeding in the DTI tool (see section 4).

A histogram of the lengths of the streamlines is calculated. It is not yet clear that this will be useful information for brain studies. However for the muscle, there are expectations that it can give an indication of the quality of the muscle. Figure 7 shows the tracing of muscle fibers in a mouse leg together with its histogram.

7. Results

This project was a collaboration with the MMC and the BioMedical NMR group at the TU/e. In this section, we will



a)

Histogram

Mean Fiberlength 3.6 mm

Minimum Fiber length 0.3 mm

Maximum Fiber length 12.2 mm

Standard Deviation 1.4 mm

Number Of Fibers 669

Number of bins 20

Bin Width 0.6

b)

Figure 7: ***a**) Streamlines traced in a data set of a mouse leg 64x64x128. **b**) Histogram of fibers of generated in **a**.*

present several studies that have been done using the DTI tool.

Figure 8 and figure 9 (see color section) show different visualizations of brain data sets. Data sets of adults have a resolution of 128x128x30 with a voxel size of 1.8x1.8x3 mm. In the case of neonates the resolution is 128x128x20 and the voxel size of 1.6x1.6x3 mm.

The evenly-spaced volume seeding has been used to generate the fibers in figure 9a to 9d. The fibers are color coded by mapping $\vec{e}_1$ to color. Figure 9a corresponds to a data set of a healthy adult. The main bundles of the fiber structure can be recognized, such as, the corpus callosum in red going from left to right, which is the connection between the hemispheres of the cerebrum, and the superior longitudinal fibers in green. In this figure, the streamsurfaces are also shown. Figure 9b shows the fibers that correspond to a data set of a neonate that was prematurely born for 26 weeks and scanned after 6 weeks. Neonates have a low anisotropic diffusion since the myelination is in development. Due to the smaller voxel size than used in adults, the neonatal DTI data is more noisy. Several fiber structures are visible in figure 9b (e.g., corona radiata in blue). However, the corpus callosum is not visible. The arrow indicates where the fibers are missing. The corpus callosum is one of the fiber structures with high anisotropy and it is missing in this patient. Further investigation of all MR images of this patient confirmed that this patient does not have a corpus callosum. In figure 9c and 9d two adults with a tumor were scanned. The arrows indicate where the tumors are. The fibers in figure 9c have been pushed to the left due to the tumor. However, in figure 9d it seems that the fibers have been destroyed, since the structure around the tumor is not moved, but in the tumor area no fibers are present.

figure	seconds	figure	seconds	figure	seconds
8a	0.51	8b	0.56	9a	55.73
9b	19.59	9c	88.8	9d	70.8

Table 1: *Computation times for the figures of this paper calculated in a 1.7 GHz PentiumIII with 1Gb RAM.*

In figure 8a and 8b, a data set of a neonate who suffered from hypoxic ischemic injury is visualized. In both images the fibers that indicate the corpus callosum have been traced. In figure 8b streamsurfaces are also traced. More structures are visible due to the extension of the streamsurfaces. In a first evaluation, no fiber damage can be observed.

The generation of streamlines and streamsurfaces using interaction techniques such as single fiber or painting, result in real-time frame rates. However, the computation times for ROI and evenly-spaced volume seeding are higher and depend on the data set and the parameters settings. In table 1, these times are shown for figure 8 and 9. Although these are not interactive times, the computation time is reasonable. Once the fibers have been generated they can be inspected at interactive frame rates.

8. Conclusions and future work

In this paper, a visualization tool for DTI data has been introduced. Streamsurfaces have been introduced to alleviate a source of error of streamline tracing by showing areas with planar anisotropy. A new algorithm has been presented to generate streamsurfaces. We also have presented a technique to fill the space regularly with streamlines and streamsurfaces based on the algorithm of Jobard et al. [JL97]. This method allows to obtain a general view of the data at different levels of streamlines density, defined by the minimum distance between streamlines or streamsurfaces. All linear and planar structures in the DTI volume are shown in one visualization. A problem that should be addressed as future work is how to avoid cluttering that usually appears in images where all space is sampled.

DTI data of the neonatal brain is noisy. Therefore, a further study in noise removal and interpolation techniques is necessary. We presented the histogram of fiber lengths as a possible quantitative measures for studies. Other measures need to be evaluated in the future.

The DTI tool proves to be useful for different examinations of human brains, in neonates and adults, and different animal studies.

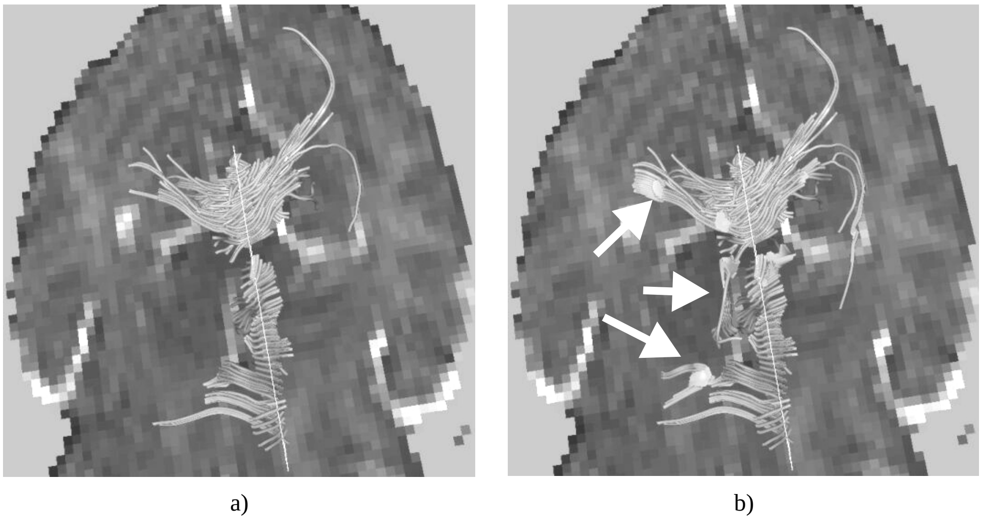


Figure 8: *Streamlines in the corpus callosum of a neonate that suffered hypoxic ischemic injury* ***b)*** *same seed points as* ***a)*** *but streamsurfaces are traced.*

Acknowledgements

We thank the BioMedical NMR group at the TU/e and Maxima Medical Center in Veldhoven for the successful collaboration and for providing the data sets and evaluations used in this paper. We thank Huub van de Wetering for his valuable comments.

References

[BP96] BASSER P., PIERPAOLI C.: Microstructural and physiological features of tissues elucidated by quantitative-diffusion-tensor mri. *J Magn Reson 111* (1996), 209–219.

[DH93] DELMARCELLE T., HESSELINK L.: Visualizing second order-tensor fields with hyperstreamlines. *IEEE Computer Graphics and Applications*, 13 (1993), 25–33.

[JL97] JOBARD B., LEFER W.: Creating evenly-spaced streamlines of arbitrary density. In *Visualization in Scientific Computing '97. Proc. of the Eurographics Workshop in Boulogne-sur-Mer, France* (1997), Springer Verlag, pp. 43–56.

[KW99] KINDLMANN G., WEINSTEIN D.: Hue-balls and lit tensors for direct volume rendering of diffusion tensor fields. In *Proc., IEEE Visualization* (1999), pp. 183–189.

[LBMP*01] LE BIHAN D., MANGIN J., POUPON C., CLARK C., PAPPATA S., MOLKO N., CHABRIAT H.: Diffusion tensor imaging: Concepts and applications. *Journal of Magnetic Resonance Imaging 13* (2001), 534–546.

[MvZ02] MORI S., VAN ZIJL P.: Fiber tracking: principles and strategies - a technical review. *NMR in Biomedicine*, 468-480 (2002).

[PB96] PIERPAOLI C., BASSER P.: Toward a quantitative assessment of diffusion anisotropy. *Magnetic resonance in Medicine 36* (1996), 893–906.

[PVH*02] POST F., VROLIJK B., HAUSER H., LARAMEE R., DOLEISCH H.: Feature extraction and visualization of flow fields. In *State-of-the-Art Proc. of EUROGRAPHICS* (2002), pp. 69–100.

[vW02] VAN WIJK J.: Image based flow visualization. *ACM SIGGRAPH 21*, 3 (2002), 745–754.

[WKL99] WEINSTEIN D., KINDLMANN G., LUNDBERG E.: Tensorlines: advection-diffusion based propagation through diffusion tensor fields. In *Proc., IEEE Visualization* (1999), pp. 249–253.

[WMK*99] WESTIN C.-F., MAIER S., KHIDHIR B., EVERETT P., JOLESZ F., KIKINIS R.: Image processing for diffusion tensor magnetic resonance imaging. In *Proc. of Second Int. Conf. on Medical Image Computing and Computer-assisted Interventions (MICCAI'99)* (1999), pp. 441–452.

[WMWJ*98] WORTH A., MAKRIS N., WEDEEN JR. V., CAVINESS V., KENNEDY D.: Fusion of MRI data for visualization of white matter bundles, 1998. http://neuro-www.mgh.harvard.edu/cma/staff/ajw/MICCAI98/MICCAI98.html.

[ZB02] ZHUKOV L., BARR A.: Oriented tensor reconstruction: tracing neural pathways form diffusion tensor mri. In *Proc., IEEE Visualization* (2002), pp. 387–394.

[ZDL03] ZHANG S., DEMIRALP C., LAIDLAW D.: Visualizing diffusion tensor MR images using streamtubes and streamsurfaces. *IEEE Transactions on Visualization and Computer Graphics 9*, 4 (2003), 454–462.

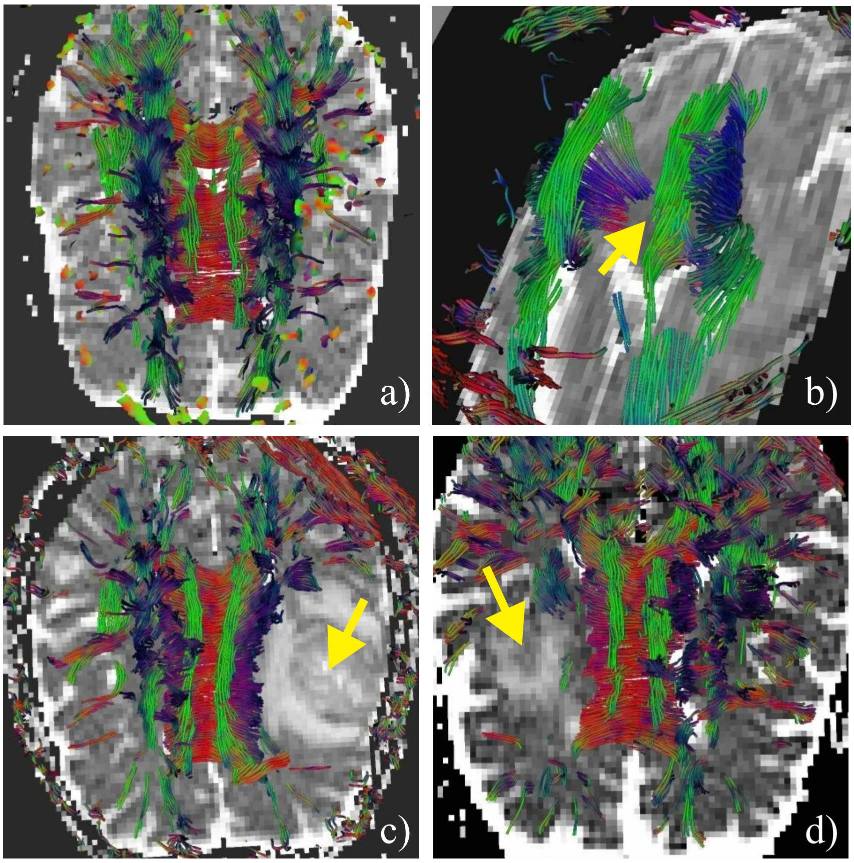


Figure 9: *Studies of fibers in the brain with different data sets.* ***a)*** *healthy adult with streamsurfaces,* ***b)*** *premature neonate missing corpus callosum (see arrow),* ***c)*** *and* ***d)*** *Show behavior of fibers in two data sets with tumors (see arrows)*