Visualization of DTI fibers using hair-rendering techniques

Peeters, T.H.J.M.; Vilanova Bartroli, A.; ter Haar Romenij, B.M.

Published in:
Proceedings of the twelfth annual conference of the Advanced School for Computing and Imaging (ASCI 2006), 14-16 june, 2006, Lommel, Belgium

Published: 01/01/2006

Citation for published version (APA):
Peeters, T. H. J. M., Vilanova, A., & Haar Romenij, ter, B. M. (2006). Visualization of DTI fibers using hair-rendering techniques. In B. P. F. Lelieveldt, & B. Haverkort (Eds.), Proceedings of the twelfth annual conference of the Advanced School for Computing and Imaging (ASCI 2006), 14-16 june, 2006, Lommel, Belgium. (pp. 66-73). Delft: Advanced School for Computing and Imaging (ASCI).

Visualization of DTI fibers using hair-rendering techniques

T.H.J.M. Peeters A. Vilanova B.M. ter Haar Romeny

Biomedical Image Analysis Group, Department of Biomedical Engineering,
Technische Universiteit Eindhoven,
W-Hoog P.O. Box 513, 5600 MB Eindhoven, The Netherlands
{T.Peeters, A.Vilanova, B.M.terHaarRomeny}@tue.nl

Keywords: Diffusion tensor imaging, computer graphics, visualization

Abstract

Diffusion Tensor Imaging (DTI) is an MRI technique that measures the diffusion of water in tissue such as white matter and muscle. From a DTI dataset, tracts representing fibers in the data can be reconstructed. Because of the vast amounts of fibers that can be reconstructed from a dataset, the visualization of these fibers is a challenging problem. In order to give the user a better understanding of the structure of the data, it is necessary to convey both the shapes of fibers, and the mutual coherency among multiple fibers and groups of fibers. Besides the fibers that were reconstructed, the local tensor properties, such as the second and third eigendirections and eigenvalues, are also of importance. We propose to use line illumination and shadowing of fibers in order to improve the perception of their structure. We also present a new method, inspired by the modeling of curled hair, for showing extra tensor properties along the fibers. This is done by showing curves that spirally wind around the actual fiber location, where the local tensor determines the parameters of that curve. We implemented the illumination, shadowing, and spiral curves, in such a way that the user can interact with the data and interactively change all parameters. The presented methods help in gaining more insight in DTI data of the brain and the heart. It is now possible to visualize more dense fiber structures using lighting and shadowing. The spiral curves help in evaluating the data where the extra tensor properties are of importance.

1 Introduction

Diffusion Tensor Imaging (DTI) is an MRI technique that measures the local diffusion of water in tissue. In water with no obstacles, water molecules move freely in all directions, thus their diffusion is *isotropic*. In tissue, the internal structure hinders the free motion of water molecules. If these structures are fibrous, the water molecules diffuse more in directions along the fibers than perpendicular to them. This causes the diffusion to be *anisotropic*. Thus, a DTI dataset can provide information on the presence and orientation of fibrous tissue.

An important application of DTI is the study of the brain, for example by visualizing white-matter tracts. DTI is the only non-invasive technique that can show these tracts *in vivo* [1]. Therefore, it is useful for, for example, brain development research and brain tumor detection. DTI can also be applied to muscle tissue and therefore it can also be used to visualize the structure of the heart muscle.

A common way to visualize DTI data is by reconstructing fibers using tractography [2]. Most existing tools for visualizing DTI fibers render them either as unshaded lines or as polygonal tubes. The use of unshaded lines gives no cues about the shape of the fibers, as is shown in figure 1(a). The use of polygonal tubes requires a very large number of polygons in order to achieve high image quality. This results in bad rendering performance. Also, neither method conveys the coherent structure of a large amount of fibers clearly.

The visualization of DTI fibers has analogies with realistic rendering of human hair. Both visualize large amounts of fibers that have particular shapes and coherencies. In DTI visualization, the fiber shapes and coherencies are important for the user because they contain vital information about the structure of a dataset. In hair rendering, the shapes of fibers and their mutual coherencies define the hairstyle, which must be conveyed to the viewer. The two most important components of realistic hair rendering are the local lighting model, and the casting of shadows from hair fibers onto each other. Both techniques are essential for creating realistic-looking images [3]. Without proper line lighting, individual fiber shapes are not ap-

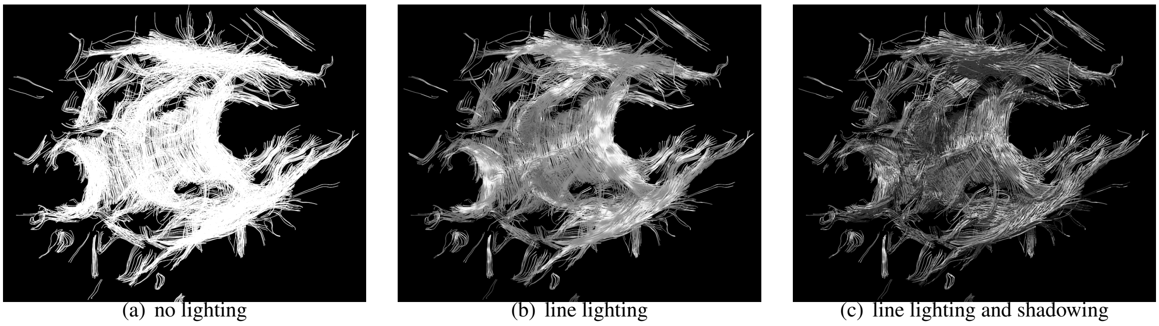

(a) no lighting (b) line lighting (c) line lighting and shadowing

Figure 1: Fibers tracked in the brain of a healthy volunteer.

parent. Without self-shadowing, the coherent structure of groups of fibers cannot be easily shown.

In scientific visualization, illuminated lines are used to acquire a better perception of shape [4, 5]. The lighting model used is similar to that used for hair rendering. However, it is not combined with self-shadows. In this paper, we show how perception of both shape and coherency of large amounts of DTI fibers can be considerably improved by applying line lighting and shadowing. We also introduce a new technique to visualize local tensor properties that makes use of the possibility to perceive fibrous structure using illumination and shadowing. This technique is inspired by the modeling of curled hair.

In section 2, we describe the input data and its properties. Section 3 contains an overview of existing DTI visualization techniques and related hair-rendering methods. The methods for visualizing DTI fibers and local tensor properties are explained in sections 4 and 5. Results are given and analyzed in section 6. Finally, in section 7, we summarize our new contributions and identify directions for future research.

2 DTI data

A DTI dataset consists of a structured grid on a volume $\mathbb{V} \subset \mathbb{R}^3$, with a *diffusion tensor* on each grid point representing the local diffusion. Each diffusion tensor is represented by a symmetric 3×3 positive definite matrix $\mathbf{D}$.

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

Trilinear interpolation on each component of $\mathbf{D}$ is used to reconstruct a continuous tensor field on $\mathbb{V}$. Other interpolation methods might be used for interpolating tensors, but that topic is beyond the scope of this paper.

Eigenanalysis of $\mathbf{D}$ gives the eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$, and the accompanying eigenvectors $\vec{e}_1, \vec{e}_2, \vec{e}_3$. The eigenvectors represent the principal diffusion directions, and the eigenvalues are the corresponding diffusion coefficients. An intuitive way of representing a diffusion tensor is with an ellipsoid that has its axes aligned with the eigenvectors of $\mathbf{D}$ and scaled by the eigenvalues.

Measure			Name
c_l	$=$	$\frac{(\lambda_1 - \lambda_2)}{3\mu_1}$	Linear diffusion
c_p	$=$	$\frac{2(\lambda_2 - \lambda_3)}{3\mu_1}$	Planar diffusion
c_s	$=$	$\frac{\lambda_3}{\mu_1}$	Spherical diffusion
FA	$=$	$\frac{3}{\sqrt{2}}\sqrt{\frac{\mu_2}{J_4}}$	Fractional anisotropy
RA	$=$	$\frac{\sqrt{\mu_2}}{\sqrt{2}\mu_1}$	Relative anisotropy

where

$$\begin{aligned} \mu_1 &= \tfrac{\lambda_1 + \lambda_2 + \lambda_3}{3} \\ \mu_2 &= \tfrac{(\lambda_1 - \lambda_2)^2 + (\lambda_2 - \lambda_3)^2 + (\lambda_1 - \lambda_3)^2}{9} \\ J_4 &= \lambda_1^2 + \lambda_2^2 + \lambda_3^2 \end{aligned}$$

Table 1: Anisotropy indices used for classifying the type of the diffusion [6, 7].

Many measures for classifying the diffusion type exist. The ones that we use are listed in table 1. We also use normalized eigenvalues λ_i^{n} defined by:

$$\lambda_i^{\mathrm{n}} = \frac{\lambda_i}{3\mu_1} \qquad i \in \{1, 2, 3\} \tag{1}$$

In section 3, we describe existing techniques for visualizing DTI data and the hair rendering techniques that we apply to DTI data in section 4.

3 Background and related work

In recent years, the visualization of DTI data has gained interest as a research topic. In this section, we give an overview of existing methods to visualize DTI data. We describe tractography and the visualization of reconstructed fibers in section 3.1. In section 3.2, we give a short overview of methods for realistic rendering of hair.

3.1 DTI Visualization

Various methods exist for visualizing DTI data [2]. One can compute scalar anisotropy indices (see table 1) from the diffusion tensors, which can be visualized by, for example, volume rendering. Less information is lost if the tensor field is simplified to the vector field defined by $\vec{e}_1$. A popular way to visualize this field is by slicing the data and applying RGB color coding. The color coding is applied by directly mapping the components of $\vec{e}_1$ to RGB color space. The resulting colors may be weighed by an anisotropy index.

A method of which the resulting visualization represents the scanned tissue in an intuitive way, is tractography or *fiber tracking*. Fiber tracking aims to reconstruct the fibrous structure that was the cause of the anisotropy measured by DTI. There are several techniques to perform this reconstruction. One way of doing this is by tracing streamlines in the vector field defined by $\vec{e}_1$. The streamlines are traced by releasing particles on seed points in the vector field. The trajectories of the particles are traced using numerical integration techniques such as first-order Euler and second-order Runge-Kutta methods. Tracing stops when a stopping criterium such as a lower bound for c_l or FA is met. The resulting streamlines represent the fibers. In this paper, we use the specific method presented by Vilanova et al. [8].

The fibers that are the output of a fiber tracking algorithm can be visualized in various ways. The simplest way to visualize them is with unshaded lines, as is shown in figure 1(a). Local tensor measures can be color-coded on the lines. However, with unshaded lines it is hard to see their actual shapes and their mutual coherencies. In order to improve the perception of shape and orientation of the fibers, the local fiber orientation can be color-coded on the lines by mapping the components of the local direction directly to RGB values. This prohibits the use of color for other purposes.

Polygonal tubes can be used to represent lines, but this results in poor image quality or bad rendering performance. Hyperstreamlines are general cylinders whose cross-sections perpendicular to the local fiber direction are ellipses whose axes and shapes are defined by $\vec{e}_2, \vec{e}_3$ and λ_2, λ_3 [9]. Hyperstreamlines show more tensor information, but do not solve the problems of representing lines by polygons. For dense sets of fibers, these techniques also result in more occlusion of information than line-based techniques.

3.2 Hair rendering

We take a different approach in reducing visual clutter and making the shapes of fibers and their coherent structures visible. We use techniques used for the realistic rendering of human hair, in order to achieve a better visualization of DTI fibers. In hair rendering, anisotropic lighting [3, 10] is used for the local shading model. This gives realistic-looking results, and conveys the fiber shapes to the viewer in an intuitive way. Some scientific visualizations use illuminated lines in order to make the shapes of the individual lines easier to interpret [5]. Our method uses the normal in the local normal plane of the fiber that maximizes the lighting intensity to do the lighting calculations. It is based on the model described by Stalling et al. [4]. Wenger et al. [11] use anisotropic lighting for volume rendering of vector-field structure and also apply it to DTI.

The large number of algorithms for rendering of shadows [12] indicates the importance of shadows for creating realistic-looking scenes. For rendering hair and fur, self-shadowing is essential to make it look real [3]. Shadow mapping is an image-space shadowing technique that is suitable for complex scenes because it does not depend on the geometric complexity of the scene [13]. However it has some problems, including aliasing on the edges of casted shadows. Therefore, many extensions to the classical shadow mapping algorithm exist, of which some were especially constructed for the rendering of complex structures such as hair [14, 15].

We apply anisotropic lighting of lines to DTI fibers in order to make the shapes of fibers better visible. We use shadow-casting of fibers onto each other to make the coherent structure of nearby fibers apparent. We also present a new method for visualizing tensor properties in a similar way that hyperstreamlines try to show information of the other components of the tensor, without using polygonal representations. We propose a method inspired by the modeling of curly hair to show more information of the tensor in section 5.

4 Light and shadow

In this section, we show how proper lighting and shadowing can be used to improve the visual perception of the structure of a DTI dataset. First, in 4.1, we describe how existing line lighting theory is implemented for use with DTI fibers. Then, in section 4.2, we explain how the standard shadow mapping technique is modified in order to give good results for dense line datasets. The algorithms given are implemented in the OpenGL Shading Language (GLSL), and run directly on the GPU. This ensures interactive rendering speeds.

4.1 Illuminated lines

If we use the Phong lighting model, the light intensity I in a point on a surface, follows the equation:

$$I = I_a + I_d + I_s = k_a + k_d(\vec{L} \cdot \vec{N}) + k_s(\vec{V} \cdot \vec{R})^n \quad (2)$$

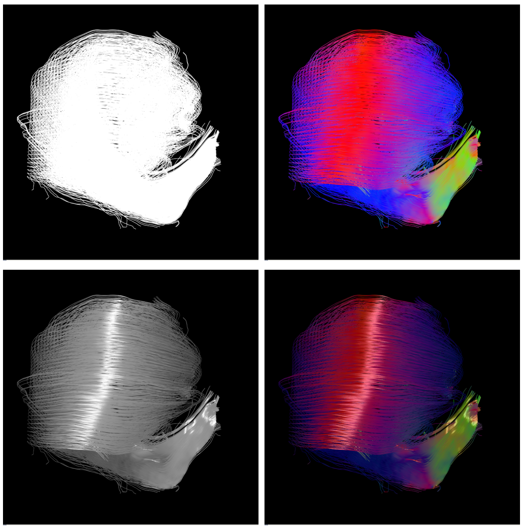


Figure 2: Fibers tracked in a folded eye nerve of a pig, which was used to create a phantom DTI dataset. Left: No coloring. Right: RGB coloring of local tangent direction. Top: No lighting. Bottom: Illuminated lines.

Here, I_a, I_d and I_s are the ambient, diffuse, and specular components of the intensity. The material-specific values of k_a, k_d, k_s and n are the ambient, diffuse and specular coefficients, and the specular component or shininess. $\vec{N}$ is the normal on the surface point. $\vec{L}$ points towards the light source, $\vec{V}$ towards the camera position, and $\vec{R}$ is the reflection of $\vec{L}$ at $\vec{N}$. Vectors $\vec{N}$, $\vec{L}$, $\vec{V}$, and $\vec{R}$ have unit length.

This model cannot be applied to illuminate lines directly, because lines do not have a single normal $\vec{N}$, but a plane of normals perpendicular to the tangent direction $\vec{T}$. This problem can be resolved by choosing for $\vec{N}$ the vector in the normal plane that maximizes $(\vec{L}\cdot\vec{N})$ and $(\vec{V}\cdot\vec{R})$ in Eq. (2). To avoid explicit calculation of the optimal $\vec{N}$, the following equations can be used [4, 10]:

$$\vec{L}\cdot\vec{N} = \sqrt{1-(\vec{L}\cdot\vec{T})^2} \tag{3}$$

$$\vec{V}\cdot\vec{R} = (\vec{L}\cdot\vec{N})\sqrt{1-(\vec{V}\cdot\vec{T})^2} - (\vec{L}\cdot\vec{T})(\vec{V}\cdot\vec{T}) \tag{4}$$

Using Eq. (3) and (4), the calculation of I in Eq. (2) can be implemented directly as a GLSL shader. Figures 1(b) and 2 show that line lighting gives better, and far more intuitive cues about the shapes of the fibers than flat shading, even if RGB color coding of the local fiber orientation is used.

4.2 Shadowing

Shadow mapping [13] is a well-known and well-researched technique. In the first render pass it renders the scene with the camera placed at the light source. The depth-values in *light-coordinates* of the rendered fragments are stored in the *shadow map*. In the second render pass, the camera is placed at the actual view position. For each fragment, the view-coordinates are converted to light-coordinates (x, y, z). The z-component is then compared to the depth-value z_s stored in the shadow map at position (x, y). If $z = z_s$ then the current fragment is visible from the light source and thus lighted. If $z > z_s$ then there is another object closer to the light source that obscures the current fragment, so it must be shadowed.

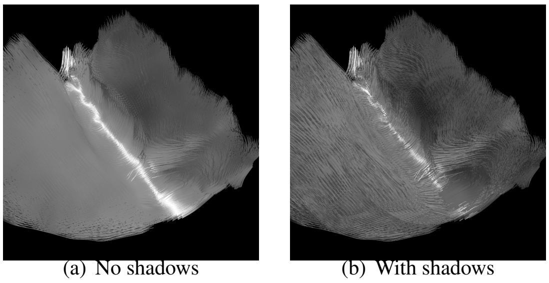
This approach has two problems. The first is the limited resolution of the shadow map combined with the computations to convert coordinates in camera space to light space. One pixel in the shadow map may represent many pixels in the image in camera space. This can cause serious aliasing artefacts. The second problem is self-shadowing of objects. Because they are the result of two different, limited-precision computations, the depth-values z and z_s will not be exactly the same. Depending on which value is larger, for each fragment there is a chance that the object casts a shadow onto itself. This problem is often dealt with by subtracting an offset d from z_s, such that the computed distance between z and z_s must be at least d for the fragment to be shadowed. This is not a solution in our case because we have very dense sets of fibers where no reasonable value can be given for d.

We tackle the aliasing problem by using a shadow map with a very high resolution. We implemented the shadow-mapping technique using Framebuffer Objects to render the shadow map to a texture with a resolution of up to 4096^2. Because in our case the light source is relatively close to the camera this sufficiently reduces the aliasing problems.

If the shadow map has a resolution that is r^2 times larger than the resolution of the image that is rendered to the screen, the shadows will become r times thinner. In order to deal with this, we simply call $glLineWidth()$ with a value of r before rendering the shadow map. We can also use a value larger than r to make lines cast thicker shadows.

The problem with lines casting shadows onto themselves is solved by not using depth-values for the shadow computations, but a unique identifier for each line. Each fiber is assigned a unique integer identifier id_f, and within a fiber, each line is assigned an identifier id_l. Together, these two values form the identification of a line which is rendered to the shadow map instead of the depth-value.

Because fragment shaders in GLSL can only output floating-point values in the range $[0, 1]$, id_f and id_l have to be converted first. There are two advantages of using this conversion over the storing of depth-values, as is done with standard shadow map-



(a) No shadows (b) With shadows

Figure 3: Fibers tracked in a part of the eye-nerve dataset.

ping.. First, we always have exactly the same input values for id_f and id_l, where the rasterization of a line on the GPU with different camera settings (one for the light source, and one for the view camera) and different output resolutions, produces different fragments with different depth-values. Second, because we can control the conversion of id_f and id_l, we can avoid having very small differences in the ID's for fibers that are close to each other. A very simple conversion is to divide id_f by the number of fibers, and id_l by the maximum number of lines per fiber.

In figures 1 and 3, the images with shadows clearly show which structures are in front and which are behind. This is the case for both fibers that are very close to each other, and for fibers that have a larger distance between them. If fibers are really close, shadowing makes their coherent structures visible. For groups of fibers that are less close to each other, shadows make it easy to estimate the distance between the fiber groups.

5 Attribute visualization

So far, we have only dealt with visualizing fibers which were constructed by tracking the direction of $\vec{e}_1$. However, other properties of tensors, such as $\vec{e}_2$ in areas with high c_p, are also of interest. In this section, we introduce a new method for visualizing the characteristics of the diffusion perpendicular to $\vec{e}_1$, inspired by the modeling of curled hair. Results for synthetic datasets are shown in figure 4. Different types of diffusion can be discerned, and in case of planar diffusion, it also shows the direction of the plane in which the most diffusion takes place. These results are explained in more detail at the end of this section. First, we describe a standard helix. Then, we generalize this helix, such that the free parameters of this curve can be used to represent the diffusion parameters. Finally, we show how these parameters are filled in, such that the curve can be used to visualize DTI fibers.

A curve of which the tangent makes a constant angle with a fixed line is called a *helix*. A helix around the z-axis has the following parametrization:

$$\begin{aligned} x &= r\cos t \\ y &= r\sin t \\ z &= ct, \end{aligned} \tag{5}$$

where r is the radius of the helix and c is a strictly positive constant defining the vertical separation of $2\pi c$ of the helix's loops. Thus, for a point $P = (0,0,z)^{\mathrm{T}}$ on the z-axis, we can acquire the corresponding point $Q = P + T_h(P)$ on the helix by translating P over:

$$\begin{aligned} T_h(P) &= \begin{pmatrix} r\cos(z/c) \\ r\sin(z/c) \\ 0 \end{pmatrix} \\ &= r\cos(z/c) * \vec{b}_1 + \\ &\quad\; r\sin(z/c) * \vec{b}_2, \end{aligned} \tag{6}$$

where $\vec{b}_i$ form the standard basis, meaning:

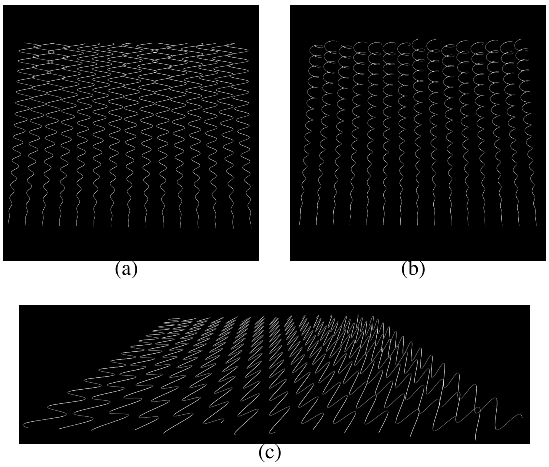
$$\begin{aligned} \vec{b}_1 &= (1,0,0)^{\mathrm{T}} \\ \vec{b}_2 &= (0,1,0)^{\mathrm{T}} \\ \vec{b}_3 &= (0,0,1)^{\mathrm{T}}. \end{aligned} \tag{7}$$

We want to construct a curve $\mathcal{C}(t)$ that spirally winds around an actual fiber $\mathcal{F}(t)$. $\mathcal{C}(t)$ must visualize the local diffusion, which is characterized by the eigenvectors and eigenvalues in the points on $\mathcal{F}(t)$. In order to achieve this, we generalize T_h to T_g. For each point $P = \mathcal{F}(t_i)$, $T_g(P)$ depends on the local tensor $\mathbf{D}_P$, and on the value of parameter t_i. Here, t_i is the value of t in P, with $0 \leq i < N$ where N is the number of points of the fiber. The value of t_i corresponds with the curve length from point 0 to i, added to a constant random value t_0 which differs for each fiber. Using T_g, we construct $\mathcal{C}(t) = \mathcal{F}(t) + T_g(\mathcal{F}(t))$, such that the maximum distance between corresponding points $P = \mathcal{F}(t_i)$ and $Q = \mathcal{C}(t_i)$ depends on the amount of diffusion in P in the direction of $Q - P$. Because the main eigendirection $\vec{e}_1$ is already represented by the tangent direction of $\mathcal{F}(t)$, we displace P only in the plane orthogonal to $\vec{e}_1$, i.e. in the space spanned by $\vec{e}_2$ and $\vec{e}_3$. This can be accomplished by using as a local basis $(\vec{e}_1, \vec{e}_2, \vec{e}_3)$ for T_g instead of the standard basis $(\vec{b}_1, \vec{b}_2, \vec{b}_3)$:

$$\begin{aligned} T_g(P) &= a_2(\mathbf{D}_P, t) * \cos(f_2(\mathbf{D}_P, t)) * \vec{e}_2 + \\ &\quad\; a_3(\mathbf{D}_P, t) * \sin(f_3(\mathbf{D}_P, t)) * \vec{e}_3 \end{aligned} \tag{8}$$

New degrees of freedom introduced in T_g are the functions a_2, a_3, f_2, and f_3. Functions a_2 and a_3 specify the amplitudes of the spiral curve $\mathcal{C}(t)$ in the directions of $\vec{e}_2$ and $\vec{e}_3$. Functions f_2 and f_3 define the frequencies of $\mathcal{C}(t)$ in those directions.

We introduce a possible definition of the degrees of freedom a_2, a_3, f_2 and f_3 to illustrate this method. In order to construct a representation that is easily understood by the viewer, we use a constant, user-defined,



(a)

(b)

(c)

Figure 4: Fibers in synthetic datasets. (a): From bottom to top c_l, changes from 1 to 0, and c_p from 0 to 1. $c_s = 0$ everywhere. (b): From bottom to top, c_l changes from 1 to 0, and c_s from 0 to 1. $c_p = 0$ everywhere. (c): $c_p = 1$. From left to right, in the plane containing the fibers, planar anisotropy changes from in-plane (horizontal) to out-of-plane (vertical).

frequency W for all curves. Thus we obtain the following definitions for f_2 and f_3:

$$f_2(\mathbf{D}_P, t) = f_3(\mathbf{D}_P, t) = W * t. \tag{9}$$

We use a_2 and a_3 to show the relative amount of diffusion in directions $\vec{e}_2$ and $\vec{e}_3$, thus we choose

$$\begin{aligned} a_2(\mathbf{D}_P, t) &= A * \lambda_2^{\mathrm{n}} \\ a_3(\mathbf{D}_P, t) &= A * \lambda_3^{\mathrm{n}}, \end{aligned} \tag{10}$$

where A is the user-defined maximum amplitude. By filling in (9) and (10) in Eq. (8), we obtain a translation function for DTI fibers T_d:

$$\begin{aligned} T_d(P) \quad = \quad & A * \lambda_2^{\mathrm{n}} * \cos(W * t) * \vec{e}_2 + \\ & A * \lambda_3^{\mathrm{n}} * \sin(W * t) * \vec{e}_3 \end{aligned} \tag{11}$$

Because the tensors are symmetric, each eigenvector may be rotated π rad, such that it points in the opposite direction. To avoid discontinuities in the constructed curve, we choose the eigenvectors in neighbouring vertices $P = \mathcal{F}(t_i)$ and $R = \mathcal{F}(t_{i+1})$ with $0 \leq i < N - 1$, such that $(\vec{e}_2(P) \cdot \vec{e}_2(R)) \geq 0$ and $(\vec{e}_3(P) \cdot \vec{e}_3(R)) \geq 0$.

By displacing all vertices of $\mathcal{F}(t)$ by $T_d(\mathcal{F}(t))$, we acquire curve $\mathcal{C}(t)$, that shows the diffusion along $\mathcal{F}(t)$ in an intuitive way. As illustrated in figure 4, different types of anisotropy can be easily discerned. In figure 4(a), from bottom to top, the diffusion changes from linear to planar. The curves change gradually from a straight line to a sine which lies completely in the plane where the diffusion occurs. When changing the camera position, this is clearly visible, see figure 4(c). The amplitudes of the sines change according to the amount of planar diffusion. In figure 4(b), from bottom to top, the diffusion changes from linear to spherical. In the spherical parts, the curves are not sines as in the planar case, but helixes. Because there is no preference for a diffusion direction there, the amplitudes of the curves are the same in all directions in the plane orthogonal to $\vec{e}_1$. Figure 4(c) shows a dataset with a constant value of c_p. From left to right, the direction of planar diffusion changes from horizontal to vertical. This is shown by always putting the curves in the diffusion plane. What we presented is a possible implementation for a_2, a_3, f_2 and f_3. Other possibilities can be explored to encode other tensor information.

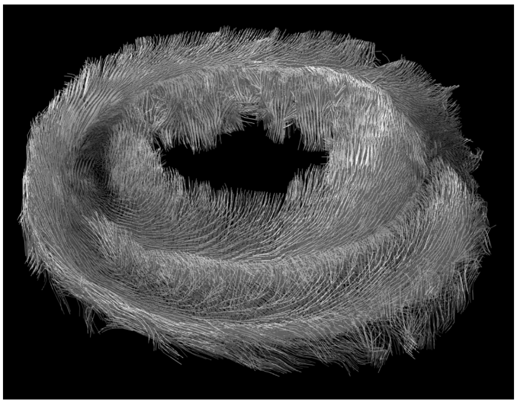


Figure 5: Short fibers tracked with seeding points in one slice of a mouse heart DTI dataset. Fibers were rendered with lighting and shadowing.

In the next section, we apply the methods we described in this section and in section 4 to different datasets.

6 Results

We applied the methods presented in this paper to different types of DTI data. Figure 1 shows fibers tracked in the brain of a healthy human volunteer. Figure 1(a) has no lighting. It is not possible to see the shapes of the fibers and their spatial relationships. RGB coloring of the local fiber orientation shows that there are several coherent groups of fibers. However, it still does not show the actual coherency among fibers or their shapes. Line illumination makes the shapes of the fibers more apparent, as is shown in figure 1(b). With shadowing, the spatial relationships between groups of fibers and among fibers that are close together, become visible. This is shown in figures 1(c) and 1(b).

Current methods of showing the varying orientation of muscle fibers in a slice of a heart include RGB color coding of $\vec{e}_1$ in that slice. Another frequently used method is to show fibers tracked from seed points that are all in one line from the outside of the heart wall to the inside. Showing these fibers vi-

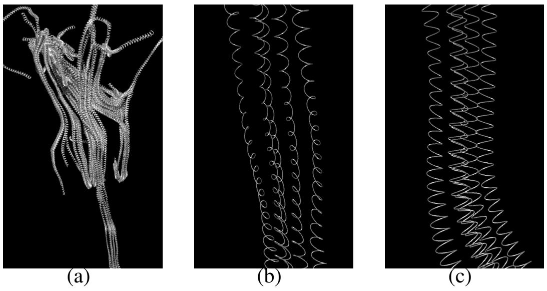


Figure 6: Fibers tracked in a region of a human brain, rendered with spiral curves in order to show extra diffusion information. (a): A few bundles of fibers. (b) and (c): Zoom in on the fibers at the bottom of (a), but both with a different camera position.

sualizes the changing fiber orientation along that chosen line. In figure 5 we present a visualization that shows more fiber information for the whole slice at once. We seeded on the whole slice and tracked short fibers. The resulting fibers are then visualized with line lighting and shadowing. In this way, a lot more data can be shown to the user at once, and the different orientations are still visible.

Figure 6 shows results of using the spiral curves for fibers tracked in a brain dataset of a healthy volunteer. When interacting with such a scene, the type of diffusion along the various fibers becomes apparent. Fibers that have highly linear diffusion have spiral curves with small amplitudes, thus they closely follow the path of the actual fiber location. Areas that are more isotropic have curves with higher amplitudes, thus the curves look more chaotic. In areas with high planar anisotropy, the curves change their path more in the planes in which the diffusion was measured than in other directions.

The fibers in figures 6(b) and 6(c) have a high c_p. This can be seen when interacting with the scene because the spiral curves have a much larger amplitude in the diffusion plane than perpendicular to that plane. Therefore, the diffusion direcion $\vec{e}_2$ can also be observed. In figure 6(b) the view-direction is close to $\vec{e}_2$, thus the amplitudes of the curves are small in this projection. In figure 6(c), the camera is rotated such that the view direction is closer to $\vec{e}_3$. This shows how the curves behave in the diffusion plane spanned by $\vec{e}_1$ and $\vec{e}_2$

Our aim was to create an interactive visualization where the user can change the possible parameters interactively. The lighting and shadowing were implemented using shaders that run on the GPU, which gives a high performance. The computations needed for the displacement of the vertices in order to obtain the curves that represent the local diffusion are done on the CPU in each render pass. Because of this, we do not need to use extra memory to store the new locations, and the user can change the parameters for the displacements interactively.

The rendering performance for several datasets is given in table 2. The measurements were made on a 3.2 GHz Pentium 4 PC with 3 GB of RAM and a GeForce 7800 GTX 256MB graphics card. The number of fibers and lines in these datasets are much larger than the amounts of fibers we could visualize formerly. This is because without the methods presented in this paper, we could not visualize such dense sets of fibers in a way that the results were interpretable by the user. We applied the spiral curve rendering to the datasets with a maximum of 248 fibers and 124102 lines. They show a framerate between 2.0 and 15 FPS. We expect to improve the render speeds drastically by implementing the translation of points P over $T_d(P)$ to run on the GPU.

7 Conclusions and future work

Our work was inspired by techniques for the rendering of human hair. With the techniques presented, it is now possible to visualize dense sets of fibers, where the structure of the fibers is more apparent. The lighting of the fibers improves the perception of their shapes. The casting of shadows by the fibers onto each other shows which fibers are in front and which are behind in an intuitive way. This also shows the coherencies among individual fibers and groups of fibers. The data-dependent curling of the fibers is a new way to visualize local diffusion properties. It can be used to distinguish the different types of diffusion (high c_l vs. high c_p vs. high c_s). In case of planar diffusion (high c_p) it also shows the direction of the plane in which the most diffusion occurs.

There are more hair-rendering techniques that can be applied to the visualization of DTI fibers. There are improved shadow mapping techniques, such as deep shadow maps [14] and opacity shadow maps [15], that were developed especially for the rendering of hair-like structures. The application of these techniques needs to be evaluated in the future.

In the current implementation the displacements of the vertices for the curl visualization are computed on the CPU as a proof of concept. These computations can be moved to vertex shaders that are executed on the GPU. This can result in an increase of performance. Also, more complex functions for $a_{2,3}$ and $f_{2,3}$ can be investigated. For example, if it is known that c_p is within a certain range for certain types of data, then a_2 and f_3 may be tuned to this range in order to show a large difference in amplitudes and frequencies for values near the upper or lower bound of the c_p range. It would also be necessary to do an extensive user study to be able to evaluate the efficiency of these curl visualization techniques.

We showed that the methods that we developed for

#fibers	total #lines	FPS flat shading	FPS line lighting	FPS light+shadow	Figure
39	8956	1152	633	26	6
160	94606	123	71	14	–
757	254934	44	25	7.0	–
3595	309876	38	22	6.4	1
9286	770068	15	8.9	2.9	3
9218	967531	12	7.1	2.5	5
1493	1006774	12	6.8	2.4	2

Table 2: Dataset size and performance in frames per seconds (FPS) for rendering with line lighting and shadow mapping. The datasets are ordered by the number of lines. Measurements were made using a 1024×768 viewport and a 4096×3072 shadow map. Datasets that do not have a figure listed are not shown in this paper.

DTI are useful for getting insight in brain and heart data. Especially for heart data, where you have a heart wall that is densely filled with fibers, our methods can show the structure of the fibers better than methods that do not use shadows and lighting. The method which we presented for visualizing extra tensor properties helps the user to distinguish between areas with different types of anisotropy. It also shows changing $\vec{e}_2$-directions in areas where the diffusion is planar.

Acknowledgments

We thank the Biomedical NMR group of the Dept. of Biomedical Engineering of the Technische Universiteit Eindhoven for the datasets.

This work was done for the Molecular Imaging of Ischemic Heart Disease project. It is supported by a BSIK grant from the Dutch Ministry of Education, Culture and Science (OCW).

References

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

[2] A. Vilanova, S. Zhang, G. Kindlmann, and D. Laidlaw. *Visualization and Processing of Tensor Fields*, chapter An Introduction to Visualization of Diffusion Tensor Imaging and its Applications. Mathematics and Visualization. Springer, 2005.

[3] J. T. Kajiya and T. L. Kay. Rendering fur with three dimensional textures. In *SIGGRAPH '89: Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, pages 271–280, New York, NY, USA, 1989. ACM Press.

[4] Detlev Stalling, Malte Zckler, and Hans-Christian Hege. Fast display of illuminated field lines. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):118–128, 1997.

[5] O. Mallo, R. Peikert, C. Sigg, and F. Sadlo. Illuminated lines revisited. In *Proceedings of IEEE Visualization 2005*, pages 19–26, October 2005.

[6] 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.

[7] Gordon Kindlmann. *Visualization and Analysis of Diffusion Tensor Fields*. PhD thesis, School of Computing, University of Utah, September 2004.

[8] A. Vilanova, G. Berenschot, and C. van Pul. DTI visualization with streamsurfaces and evenly-spaced volume seeding. In O. Deussen, C. Hansen, D.A. Keim, and D.Saupe, editors, *Joint EUROGRAPHICS - IEEE TCVG Symposium on Visualization*, 2004.

[9] Thierry Delmarcelle and Lambertus Hesselink. Visualizing second-order tensor fields with hyperstreamlines. *IEEE Comput. Graph. Appl.*, 13(4):25–33, 1993.

[10] David C. Banks. Illumination in diverse codimensions. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 327–334, New York, NY, USA, 1994. ACM Press.

[11] Andreas Wenger, Daniel F. Keefe, and Song Zhang. Interactive volume rendering of thin thread structures within multivalued scientific data sets. *IEEE Transactions on Visualization and Computer Graphics*, 10(6):664–672, 2004. Member-David H. Laidlaw.

[12] Andrew Woo, Pierre Poulin, and Alain Fournier. A survey of shadow algorithms. *IEEE Comput. Graph. Appl.*, 10(6):13–32, 1990.

[13] Lance Williams. Casting curved shadows on curved surfaces. In *SIGGRAPH '78: Proceedings of the 5th annual conference on Computer graphics and interactive techniques*, pages 270–274, New York, NY, USA, 1978. ACM Press.

[14] Tom Lokovic and Eric Veach. Deep shadow maps. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 385–392, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[15] Tae-Yong Kim and Ulrich Neumann. Opacity shadow maps. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, pages 177–182, London, UK, 2001. Springer-Verlag.