The Application of GPU Particle Tracing to Diffusion Tensor Field Visualization

Polina Kondratieva, Jens Krüger, Rüdiger Westermann*

Computer Graphics and Visualization Group
Technische Universität München

ABSTRACT

In this paper we introduce GPU particle tracing for the visualization of 3D diffusion tensor fields. For about half a million particles, reconstruction of diffusion directions from the tensor field, time integration and rendering can be done at interactive rates. Different visualization options like oriented particles of diffusion-dependent shape, stream lines or stream tubes facilitate the use of particle tracing for diffusion tensor visualization. The proposed methods provide efficient and intuitive means to show the dynamics in diffusion tensor fields, and they accommodate the exploration of the diffusion properties of biological tissue.

CR Categories: I.3.8 [Computing Methodologies]: Computer Graphics–Applications—; J.3 [Computing Application]: Life and Medical Sciences—; I.3.7 [Computer Graphics]: Methodology and Techniques—Interaction Techniques;

Keywords: Diffusion Tensors, Dynamic Visualization, GPU Particle Tracing and Streamlines, Medical Visualization

1 INTRODUCTION

The diffusion properties of biological tissue can be measured using diffusion tensor magnetic resonance imaging (DT-MRI) [1]. The imaging process reveals the diffusion of water molecules depending on the shape and orientation of tissue cells, i.e. within fibrous material the diffusion is anisotropic while there is an equal diffusion probability in real matter of other types. The diffusion probability is characterized by a second order tensor, which describes the deflection of a molecular pathway as a function of spatial position.

Approaches to visualize the diffusion in real tissue can be classified into two major categories: glyph-based techniques reveal local variations in diffusion tensor fields by mapping tensor properties like deflection or diffusion probability to the shape or appearance of graphical primitives, i.e. ellipsoids [10], composite shapes [15], or superquadrics [8]. In contrast, directional tracking of massless particles along the most probable diffusion directions in tensor field data [4] allows for the classification of anatomical structures, e.g. the white matter fiber tracks. Different geometric representations like stream lines [14, 12] and stream tubes [17, 3], or stream surfaces [17, 13] have been employed to visualize these structures. To improve the quality and stability of such tracking techniques, regularization and filtering approaches [2, 18, 5] along with heuristics to determine the most probable directions [14, 18] have been proposed. Dedicated color and opacity mapping schemes to visually enhance particular features in diffusion tensor data have been presented in [14, 7, 15, 8].

While tracking based techniques can effectively visualize local features in tensor fields, glyph-based imaging techniques for visualizing 3D fields can successfully illustrate the global behavior of such fields. However, it is difficult when using such methods to effectively control glyph density, shape and appearance in a way that depicts both the direction structure of the diffusion *and* the diffusion magnitude. Neither of these techniques allows for interactive exploration of large tensor fields, and they typically fail to visualize and to monitor the diffusion dynamics in real tissue.

In this paper, we propose an interactive technique based on GPU particle tracing for tensor field visualization. This method can display the dynamics of large particle sets in flow fields, and it can thus be used to monitor the diffusion in biological tissue in real time. A number of visualization options like oriented texture splats, stream lines and stream tubes provide the user with an effective means for the visual analysis of 3D tensor fields given on a Cartesian grid.

2 DIFFUSION TENSORS

The second order diffusion tensor can be expressed mathematically as a 3×3 symmetric semi-positive matrix:

$$\mathbf{D} = \begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{pmatrix} \tag{1}$$

From a physical point of view, the tensor $\mathbb{D}$ describes the probability density for the deflection of molecular pathways in diffusive tissue. Following the common classification, diffusion tensors can be represented as ellipsoids with main, medium, and minor axes corresponding to the eigenvectors e_1, e_2, e_3 – with respective eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$ – of the tensor. The relative differences between the eigenvalues are related to the *anisotropy* of the diffusion. Three basic types of anisotropy are usually considered in the literature (see Table 1). A thorough discussion of diffusion tensors and other derived quantities can be found in [8, 16].

Table 1: Classification of diffusion tensor anisotropy.

Name	Coefficient
linear	$c_l = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3}$
planar	$c_p = \frac{2(\lambda_2 - \lambda_3)}{\lambda_1 + \lambda_2 + \lambda_3}$
spherical	$c_s = \frac{3\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}$

Besides the mapping of local anisotropy to the shape of geometrical icons, a number of different color encoding schemes have been proposed:

$$RGB \quad = \quad (\mu_1,\, \mu_2,\, A_3) \tag{2}$$

*kondrati@in.tum.de,kruegeje@in.tum.de,westermann@in.tum.de

$$RGB = (J_4,\ FA,\ A_3) \quad (3)$$
$$RGB = (c_l,\ c_p,\ c_s) \quad (4)$$
$$RGB = FA(|e_{1_x}|,\ |e_{1_y}|,\ |e_{1_z}|) \quad (5)$$
$$RGB = RA(|e_{1_x}|,\ |e_{1_y}|,\ |e_{1_z}|) \quad (6)$$

Here, the coefficients μ_1, μ_2, A_3, J_4, FA, RA are computed as follows:

$$\mu_1 = \frac{1}{3}\sum\lambda_i, \quad 1^{st} \text{ central moment of eigenvalues,}$$
$$\mu_2 = \frac{1}{3}\sum(\lambda_i - \mu_1)^2, \quad 2^{nd} \text{ central moment of eigenvalues,}$$
$$A_3 = \frac{\sum(\lambda_i - \mu_1)^3}{3\cdot\mu_2\sqrt{\mu_2}}, \quad \text{skewness of eigenvalues,}$$
$$J_4 = \sum\lambda_i^2, \quad \text{invariant of tensor } \mathbb{D},$$
$$FA = \frac{3}{\sqrt{2}}\sqrt{\frac{\mu_2}{J_4}}, \quad \text{fractional anisotropy,}$$
$$RA = \frac{\sqrt{\mu_2}}{\sqrt{2}\mu_1}, \quad \text{relative anisotropy}$$

3 GPU Particle Engine

To interactively explore the dynamics in 3D diffusion tensor fields, we employ a particle system for visualizing steady 3D flow fields on Cartesian grids [9]. We exploit features of recent graphics accelerators to advect particles on the GPU, saving particle positions in graphics memory, and then sending these positions through the GPU again to obtain images in the frame buffer. By using this functionality, particle tracing in Cartesian grids can be entirely performed on the GPU without any read back to application memory. This approach allows for interactive streaming and rendering of millions of particles, and it enables virtual exploration of high resolution fields. In the current scenario, the ability to display the dynamics of large particle sets using visualization options like oriented texture splats, stream lines, and stream tubes provides an intuitive means for the visual analysis of 3D tensor fields that is far beyond existing solutions.

Particle advection is performed in the fragment units of programmable GPUs. During particle tracing, particle positions are subsequently read from the current particle container and the result of the advection step is written to an additional container. Containers are internally stored as 2D texture maps. The vector field data is stored in the RGB color components of a 3D texture. To display particle primitives, a vertex array containing as many entries as there are particles is rendered. This array resides in local GPU memory. In the GPU's vertex units, respective particle positions are fetched from the current particle container, and vertex positions are displaced accordingly. This is accomplished using functionality in Shader Model 3.0 [11].

The particle engine provides different visualization options. Oriented 3D icons can be simulated by rendering textured point sprites in combination with a texture atlas. For every icon this atlas contains a set of pre-computed views, of which the one most similar to the actual view is rendered. In addition, the traces of particles can be stored and used to render stream lines, stream balls or stream tubes.

4 Tensor Visualization using Particle Tracing

To use the GPU particle engine (see Figure 1) for tensor field visualization, a number of data-specific extensions have been integrated. In particular, the direction vector along which a particle is traced first has to be reconstructed from the tensor field. The basic differences between particle tracing in flow fields and in tensor fields are illustrated in Figure 2.

In the following, the specific extensions we have integrated to accommodate the use of particle tracing for tensor field visualization are described:

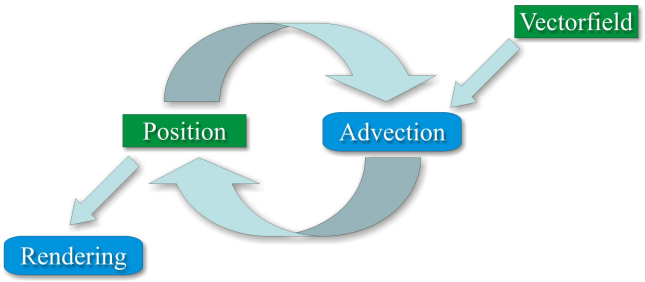

Figure 1: The work cycle of a general particle engine. The advection subsystem takes an initial position, and a vector field to generate a new position which is used for rendering. In the next time-step the initial position is replaced by the advected position and the whole process is repeated from the beginning.

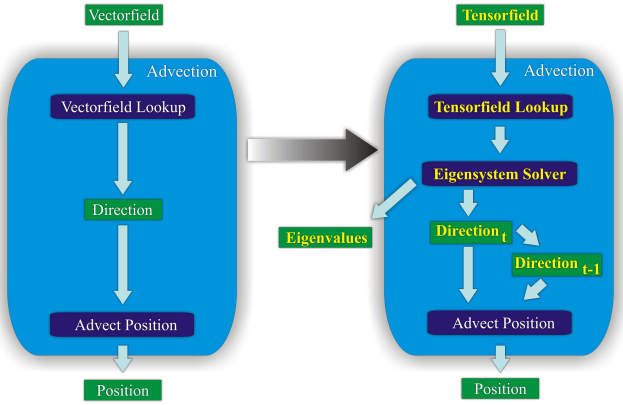

Figure 2: Differences between particle tracing in a vector field and in a tensor field. Note that the advection of a particle in a tensor field as proposed in [14] requires both the current and the last particle direction.

- The six distinct entries of the diffusion tensor are stored in two 3D RGB texture maps. Because hardware accelerated trilinear interpolation of 32 bit floating point textures is not supported on recent GPUs, we use 16 bit floating point textures in the current implementation. If this precision is not sufficient, hand–coded interpolation of 32 bit floating point values can be performed in a fragment shader. As shown in [9], this only results in a slight decrease in performance.

- At every particle position, the tensor is tri-linearly interpolated from the two 3D texture maps.

- To derive a vector field for particle tracing, at every particle position the eigen-decomposition of the resampled tensor is computed on the fly [1]. Therefore, on the GPU we have implemented a non-iterative analytical algorithm proposed by Hasan et al. [6]. If the tensor is not *positive-definite*, or if its eigenvalues are degenerate (equal to each other), the propagation process is terminated.

- For the rendering of textured particle sprites, two different texture atlases for tensor visualization have been designed. The first atlas extends the one proposed in [9] about an additional scaling factor used to emphasize the diffusion

[1] The interpolation of precomputed eigenvectors and eigenvalues does not allow for a consistent computation of diffusion direction [8, 18].

anisotropy $c_a = c_l + c_p$. The second one contains precomputed views of short stream tubes at different orientation and size. By texturing particle sprites with the image of the respective view – scaled according to the tensor attributes – the appearance of closed stream tubes can be simulated.

- Because particle tracking in 3D tensor fields along the largest eigenvector of the tensor can lead to ambiguous results (opposite eigenvector directions are both valid), the outgoing particle direction is computed as a linear combination of the deflected incoming direction and the principal eigenvector [14].

- A number of different criteria to stop particle propagation have been integrated [8]. In particular, if the fractional anisotropy is less than a given threshold, a particle trace is stopped. In addition, if the fractional anisotropy is above a threshold, it is used to modulate the particle transparency. In this way, particles can be faded out continuously in nearly isotropic regions.

- The user can interactively select and change tensor-specific parameters, such as the color mapping scheme, the threshold of the fractional anisotropy used to fade out particles, and the propagation algorithm (along the largest eigenvector or along the deflected direction).

5 Results

All of our experiments have been done on a NVIDIA Quadro FX 4400 graphics card equipped with 512MB video memory. Rendering was performed into a viewport of 1280×1024 pixels. In this section we show screenshots of the dynamic 3D tensor field visualization including various visualization options. Although the images already show the functionality of the particle engine, we should note here that the real benefit of the presented approach is a lot more apparent in the accompanying video.

Table 2 gives timings for the advection and rendering of particles. Timings in the second column include all operations that are carried out until updated particle positions are available in the current particle container, i.e. tensor interpolation, eigendecomposition and particle advection using Euler integration. The third and the fourth columns include timings for the rendering of diffusion-aligned point sprites. To demonstrate the dependency of performance from the number of generated fragments, differently sized sprites are used. The last column shows the time that is required to reconstruct different amounts of stream lines of length 100. In all these tests, a 256^3 tensor data set (canine heart) was visualized.

As can be seen, even for large particle sets the GPU implementation still allows for the interactive exploration of tensor fields. It is interesting to note, that with increasing fragment size the rendering stage quickly becomes fragment bound. Moreover, the number of generated fragments strongly depends on viewing parameters, such as field of view, the distance of particles to the camera. As it is rather difficult to provide precise and meaningful timing statistics for different amounts of particles in combination with differently sized point sprites, we give specific timings for all the generated images shown in the following. Also, since for fibers the number of advection steps is proportional to their length, the approximate time for tracking and rendering a fiber of an arbitrary length can be computed as the time needed for one segment multiplied by the number of segments.

In contrast to previous approaches, where only the final rendering of precomputed entities can be done interactively, the proposed method provides an even more intuitive means to explore high resolution diffusion tensor fields. In particular, the user can interactively select the seed density of particles as well as parameters specific to the propagation process. By using these options, pathways can be visualized at almost arbitrary resolution. The advection of oriented particles allows for the simultaneous exploration of both local *and* global diffusion tensor properties. Fiber structures can be observed without that particle positions have to be connected. Even in still images the fibrous structures in the investigated tissue can clearly be seen.

particles	advection only	oriented splats		*lines*	length 100
		1×1 pixels	7×7 pixels		
64^2	1434	700	470	64	130
128^2	343	185	123	128	100
256^2	83.2	43.7	28.8	256	78
512^2	19.4	8.4	5.9	512	55
1024^2	4.2	2.0	1.6	1024	38

Table 2: Application performance in frames per second, for different amounts of particles and for stream tubes each consisting of 100 segments.

Figures 3 to 9 show a number of examples that were generated using the presented particle engine. Two data sets were used to demonstrate the effectiveness of particle tracking for tensor field visualization: a human brain data set of resolution $148 \times 190 \times 160$ and a canine heart data set of resolution $256 \times 256 \times 256$. Both data sets consist of six diffusion tensor components and an additional confidence value per voxel. Different visualization options and color mappings were selected to emphasize particular diffusion properties and anatomical structures in these data sets. In all examples, visual exploration was performed in real-time, thus allowing for an effective and intuitive analysis of biological tissue.

In Figure 3 colored point sprites of size 5×5 pixels were rendered to visualize the human brain data set. The *FA*-based color mapping scheme (equation 5) was used. By fading out particles in regions showing low anisotropy, highly anisotropic brain structures, such as corpus callosum, corona radiata and pyramid, are emphasized and can clearly be distinguished in the generated image.

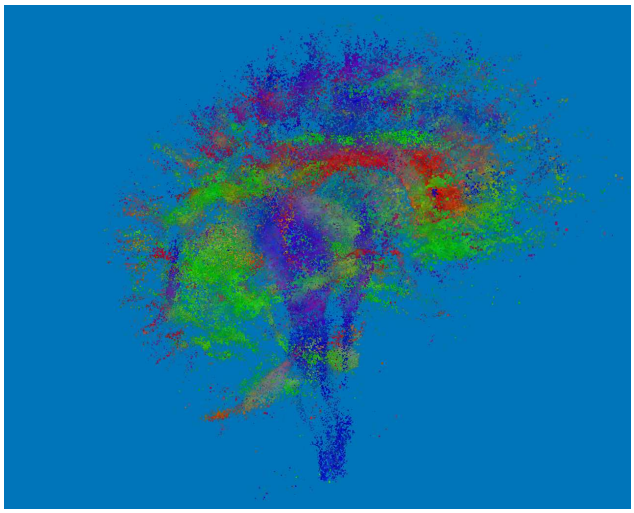

Figure 3: Visualization of the diffusion tensor field measured in a human brain. Below a given anisotropy threshold the transparency of rendered particle primitives is inversely proportional to the measured anisotropy. In this way, particles in regions of low anisotropy are continuously faded out. 64K particles of size 5×5 pixels were rendered at roughly 40 frames per second.

In Figure 4, diffusion-dependent oriented sprites were used to

render a close up view of the corona radiata. According to the *FA*-based color mapping scheme, an optic tract can be clearly distinguished by green color. To generate the image, 64K point sprites of size 45×45 pixels were rendered at 6.1 frames per second.

The left image in Figure 5 shows a visualization of the corona radiata using stream tubes. The seed box from which particle traces were initially released is shown as wire frame in red. Overall, 512 traces of a maximum length of 600 were traced. Textured point sprites of size 15×15 pixels were rendered using the texture atlas described above. Reconstruction and rendering was performed at 2.4 frames per second. As can be seen, this visualization option helps to track the bunches of fibers with biologically similar properties, and it allows the expert to follow the paths of single fibers. On the right of Figure 5 the superior longitudinal fasciculus is rendered by means of colored stream lines. The axial slice along the fasciculus was chosen as seed region. 512 lines of a maximum length of 600 were computed and displayed at 10 frames per second. Both images show nicely the property of stream tubes and stream lines to effectively reveal spatial relationships between different tracks.

The sagittal, coronal, and axial slices through the human brain are visualized in Figure 6 using diffusion-dependent sprites. In combination, shape and color of the rendered ellipsoids give a good impression of the degree of anisotropy and the main diffusion direction within the corresponding brain structures. The brain structures themselves, such as pyramidal tracts, U-shaped fibers, superior longitudinal fasciculus and others, are clearly visible in all slices. The axial and sagittal slices were rendered using 64K particles of size 20×20 pixels at 10.8 frame per second. The coronal slice was produced using 256K particles of size 10×10 pixels at 4.5 frames per second.

Visualizations of the canine heart data set are shown in Figures 7, 8, 9. Figure 7 shows the longitudinal and latitudinal heart slices, rendered with textured point sprites. The orientation of the sprites corresponds to the helical construction of the heart muscle, and the spatial positions depict the heart structure consisting of four chambers and several valves in between. The animation of 64K particles of size 25×25 pixels runs at roughly 8 frames per second.

Figure 8 shows images of the heart data set rendered from different view points using aligned diffusion-dependent point sprites. In the interactive animation, the clockwise and counter-clockwise motion of particles along the muscle structure can be clearly tracked. Both images were rendered using 64K particles at 25×25 and 45×45 pixels per point sprite, respectively. Accordingly, the application performance dropped from 9.4 frames per second to 7.4 frames per second.

Stream tubes were used in Figure 9 to visualize the heart data set. On the right, tubes were colored according to equation 3. Both images were generated using 1K stream tubes of maximum length of 1K in combination with point sprites of size 5×5 pixels to render the texture-based tube segments. Reconstruction and rendering was done at 1.5 frames per second.

6 Conclusion

We have presented an efficient and effective visualization system for 3D diffusion tensor fields. This system allows for the visual exploration of such fields at interactive rates and at arbitrary level of detail. The user can interactively select regions of interest by positioning a particle probe of adjustable position and size. A number of different visualization options can be selected to visualize local as well as global features in the diffusion process.

By using this system, local variations in diffusion tensor fields as well as anatomical structures can be visualized at interactive rates. Due to the possibility to simulate the dynamic behavior of massless particles in the derived diffusion field, a very intuitive approach to understanding diffusion tensor fields has been presented. In contrast to previous methods, real-time advection and rendering of large particle sets produces animations that closely and intuitively mimic the underlying dynamic diffusion process.

Acknowledgments

We would like to thank Prof. Edward W. Hsu, Department of Bioengineering, Duke University, Leonid Zhukov and Gordon Kindlmann for providing the tensor field data sets.

References

[1] P.J. Basser, J. Mattiello, and D. LeBihan. MR diffusion tensor spectroscopy and imaging. *Biophysical Journal*, pages 259–267, 1994.

[2] O. Coulon, D. Alexander, and S.R. Arridge. A regularization scheme for diffusion tensor magnetic resonance images. *Lecture Notes in Computer Science*, 2082:92–105, 2001.

[3] Marco DaSilva, Song Zhang, Cagatay Demiralp, and David H. Laidlaw. Visualizing diffusion tensor volume differences. In *Visualization '01 Work in Progress Proceedings*, pages 16–17, October 2001.

[4] Thierry Delmarcelle and Lambertus Hesselink. Visualization of second order tensor fields and matrix data. In *Proceedings IEEE Visualization*, pages 316–323, 1992.

[5] C. Gossl, L. Fahrmeir, B. Putz, L.M. Auer, and D.P. Auer. Fiber tracking from DTI using linear state space models: Detectability of the pyramidal tract. In *NeuroImage*, volume 16, pages 378–388, 2002.

[6] K. M. Hasan. Analytical computation of the eigenvalues and eigenvectors in DT–MRI. *IEEE Computer Graphics Applications*, pages 36–46, 1991.

[7] G. Kindlmann and D. Weinstein. Hue–balls and lit–tensors for direct volume rendering of diffusion tensor field. In *Proceedings IEEE Visualization*, pages 183–189, 1999.

[8] Gordon Kindlmann. *DTI Visualization and Analysis of Diffusion Tensor Fields*. PhD thesis, University of Utah, 2003.

[9] J. Krueger, P. Kipfer, P. Kondratieva, and R. Wetsermann. A particle system for interactive visualization of 3d flows. *IEEE Transactions on Visualization and Computer Graphics*, to appear.

[10] David H. Laidlaw, Eric T. Ahrens, David Kremers, Matthew J. Avalos, Carol Readhead, and Russell E. Jacobs. Visualizing diffusion tensor images of the mouse spinal cord. In *Proceedings IEEE Visualization*, pages 127–134, 1998.

[11] Microsoft. Shader model 3.0 specification. http://msdn.microsoft.com, 2004.

[12] S. Mori, B.J. Crain, V.P. Chacko, and P.C. van Zijl. Three dimensional tracking of axonal projections in the brain by magnetic resonance imaging. *Ann Neurol*, 45(2):265–269, 1999.

[13] A. Vilanova, G.Berenschot, and C. van Pul. DTI visualization with streamsurfaces and evenly–spaced volume seeding. In *VisSym '04 Joint EG –IEEE TCVG Symposium on Visualization*, pages 173–182, 2004.

[14] David M. Weinstein, Gordon L. Kindlmann, and Eric C. Lundberg. Tensorlines: Advection-diffusion based propagation through diffusion tensor fields. In *IEEE Visualization '99*, pages 249–254, 1999.

[15] C.-F. Westin, S.E. Maier, H. Mamata, A. Nabavi, F.A. Jolesz, and R. Kikinis. Processing and visualization for diffusion tensor MRI. *Medical Image Analysis*, 6:93–108, 2002.

[16] Masutani Y., Aoki S., Abe O., Hayashi N., and Otomo K. MR diffusion tensor imaging: recent advance and new techniques for diffusion tensor visualization. *European Journal of Radiology*, 1(46):53–66, 2003.

[17] Song Zhang, Charles T. Curry, Daniel S. Morris, and David H. Laidlaw. Streamtubes and streamsurfaces for visualizing diffusion tensor MRI volume images. In *Visualization '00 Work in Progress*, October 2000.

[18] L. Zhukov and A. Barr. Oriented tensor reconstruction: tracing neural pathways from diffusion tensor MRI. In *Proceedings of the IEEE Visualization '02*, pages 387–394, 2002.

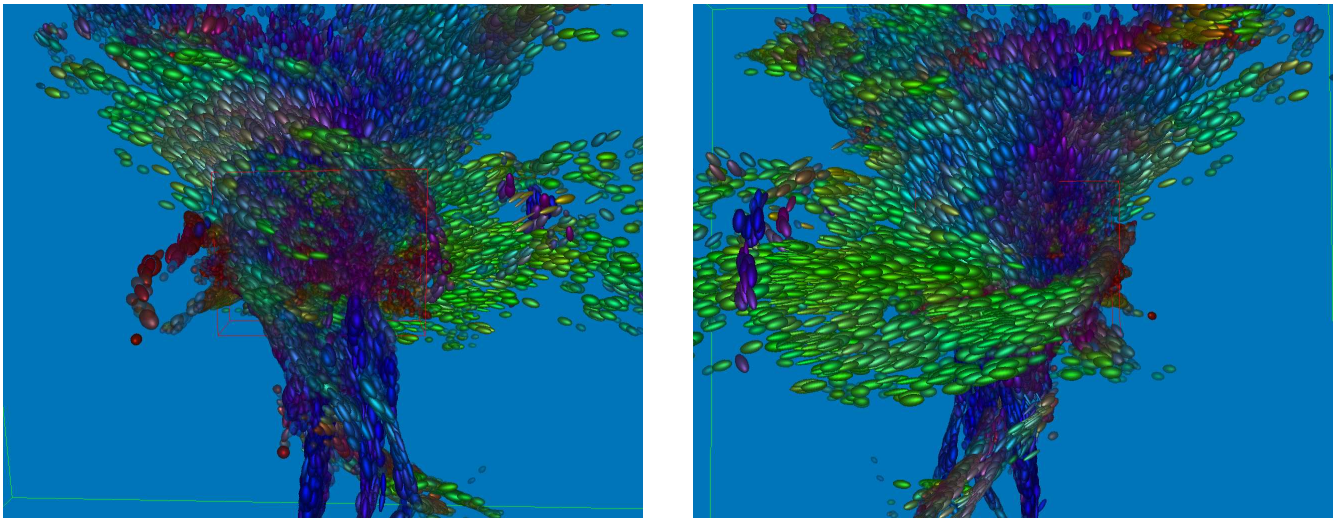

Figure 4: Brain structures: By using the color mapping scheme described in equation 5, the optic tract (green) and corona radiata (blue) fibers can be clearly distinguished. At 6.1 frames per second, 64K particles can be advected, and rendered using textured sprites of size 45×45 pixels.

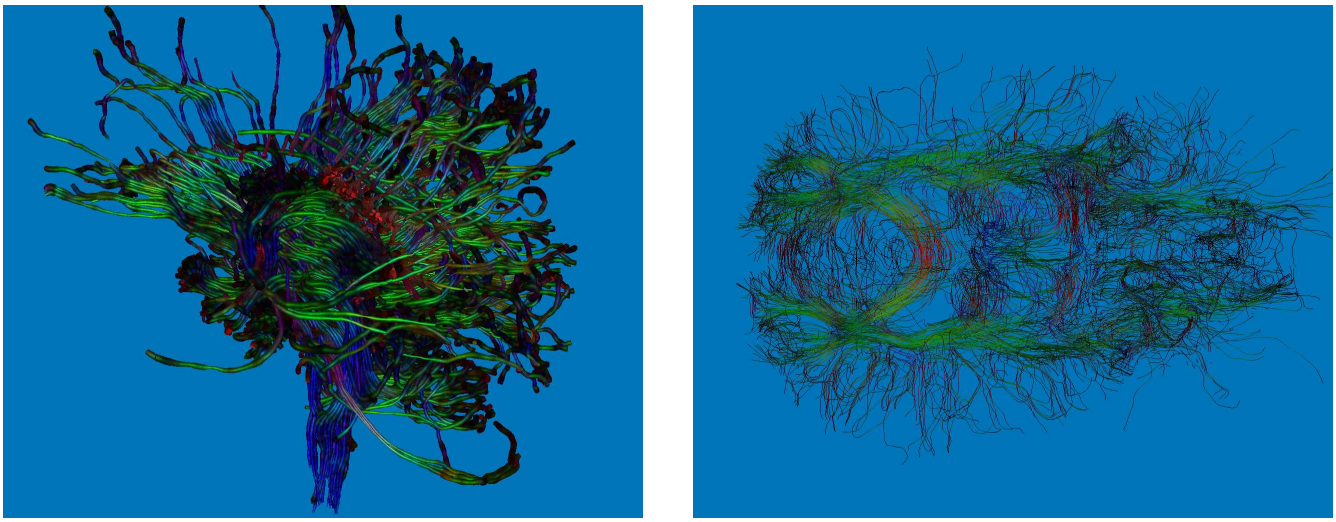

Figure 5: Brain structures: On the left, corona radiata is visualized using stream tubes. On the right, the axial slice along the longitudinal fasciculus is rendered by means of simple stream lines. The FA-based color mapping scheme (equation 5) is applied. 512 fibers with a maximum length of 600 can be generated and rendered at 2.4 and 10 frames per second, respectively.

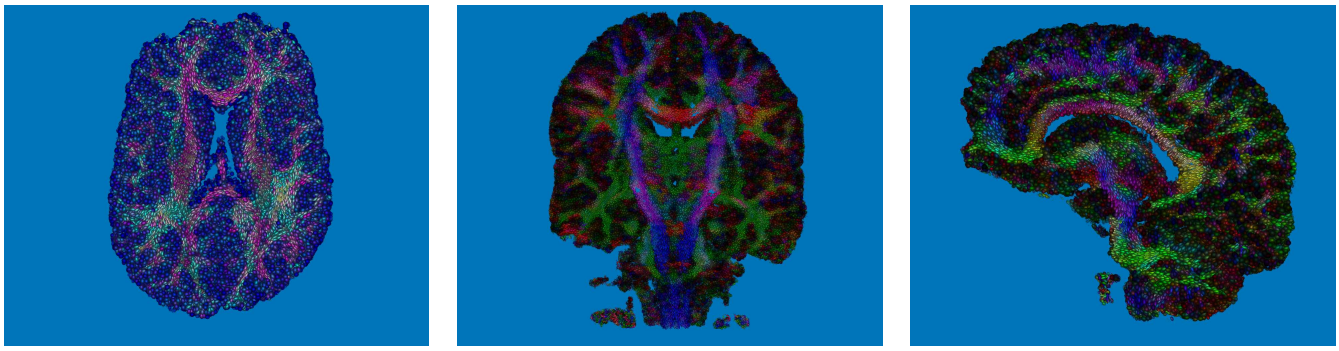

Figure 6: Brain structures: Axial (left), coronal (middle) and sagittal (right) slices through the brain are shown. The coronal and sagittal slices are color coded using equation 5, and equation 4 is used to color the axial slice. The axial and sagittal slices were rendered using 64K particles of size 20×20 pixels at 10.8 frames per second. The image of the coronal slice was generated at 4.5 frames per second using 256K particles of size 10×10 pixels.

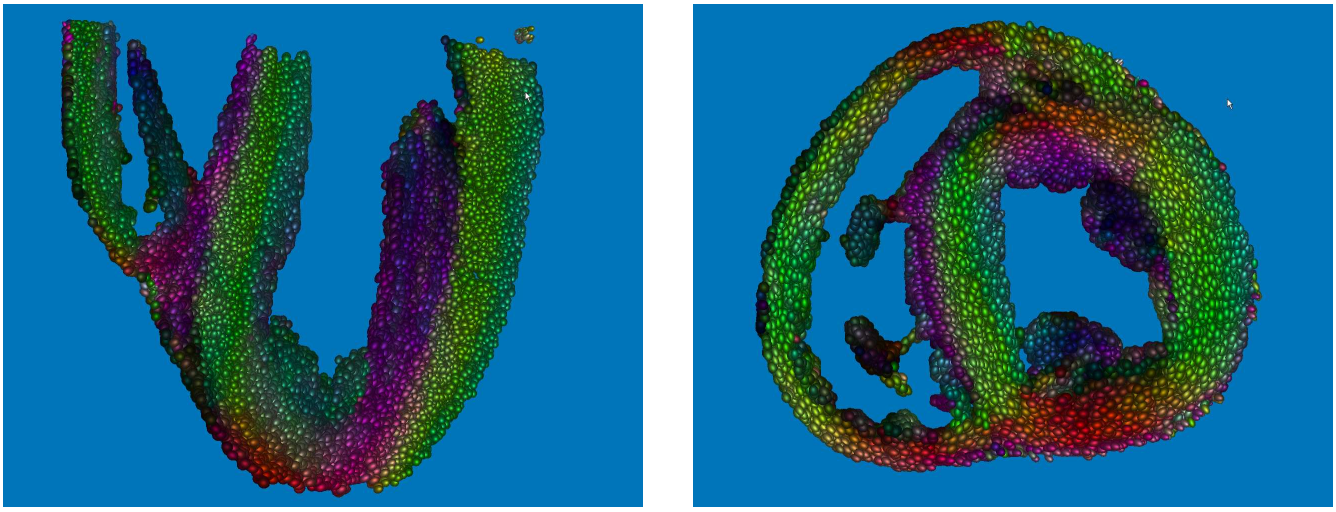

Figure 7: Longitudinal (top) and latitudinal (bottom) slices of the heart muscle are reconstructed and rendered at 8 frames per second. The helical orientation of the heart muscle fiber becomes apparent from the latitudinal slice. Both slices also depict the four-chambered structure of the heart. The FA-based color mapping scheme was chosen.

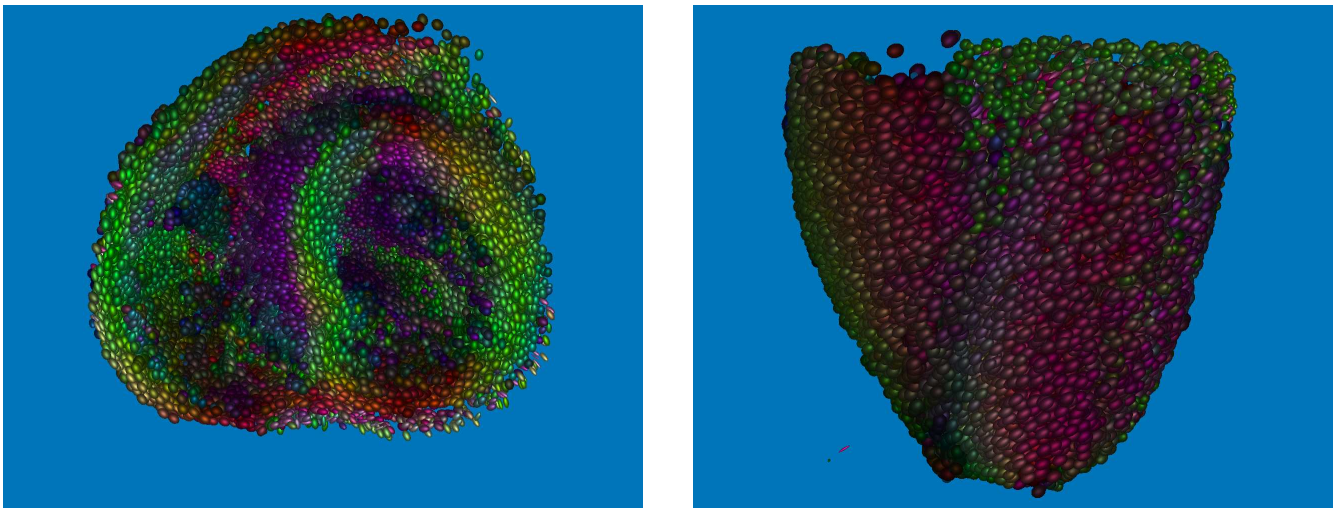

Figure 8: Heart muscle: Multiple views of the data set are shown. The helical structure of the heart muscle can be easily tracked on both images. The FA-based color mapping scheme (equation 5) was applied. 64K particles of size 25×25 pixels (left) and 45×45 pixels (right) were animated at 9.4 and 7.4 frames per second, respectively.

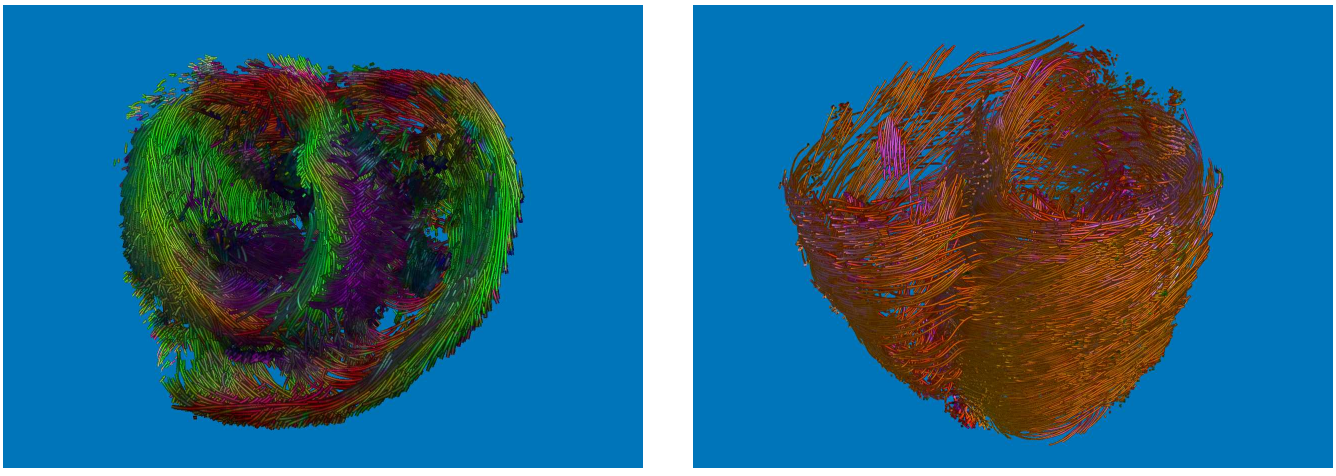

Figure 9: Heart muscle: Multiple views of the data set are shown. The color mapping schemes described in equation 5 (left) and equation 3 (right) were used. 1K stream tubes of maximum length of 1K were reconstructed and rendered with point sprites of size 5×5 pixels. Both images were generated at 1.5 frames per second.