High-Order Diffusion Tensor Connectivity Mapping on the GPU

Tim McGraw and Donald Herring

Purdue University

Abstract. We present an efficient approach to computing white matter fiber connectivity on the graphics processing unit (GPU). We utilize a high-order tensor model of fiber orientation computed from high angular resolution diffusion imaging (HARDI) and a stochastic model of white matter fibers to compute and display global white matter connectivity in real time. The high-order tensor model overcomes limitations of the 2nd-order tensor model in regions of crossing or fanning fibers. By utilizing modern GPU features exposed in recent versions of the OpenGL API we can perform processing and visualization without costly GPU-CPU data transfers.

1 Introduction

Diffusion tensor magnetic resonance imaging (DT-MRI) makes it possible to compute, *in vivo*, neuronal connectivity in the brain, a valuable capability when assessing brain injury and diseases of the central nervous system. To make DT-MRI practical for clinical use many processing and visualization challenges must be overcome. Computing neuronal connectivity provides crucial information for diagnosis, but it is a time-consuming process. This drawback must be overcome in order to make usage of DT-MRI in time-critical situations feasible. Effective interactive visualization techniques must be developed so that clinicians and researchers can quickly pinpoint areas of interest in the brain.

Random molecular motion causes transport of water at a microscopic scale within biological systems. The properties of the surrounding tissue can affect the diffusion process, making it anisotropic (directionally dependent). Within an oriented structure, such as a bundle of axonal fibers within white matter of the brain, water tends to diffuse parallel to the fiber direction. MRI protocols have been developed which permit the tensor which characterizes directional dependence of diffusion to be estimated. The resulting images can be used to infer tissue structure. Numerous approaches to diffusion tensor estimation and processing have been proposed, many imposing constraints such as symmetry and positive-definiteness so that the tensor represents a physically plausible diffusion process [1].

Visualization of DT-MRI is complicated by the fact that the data are tensor-valued. It is difficult to perceive the underlying connectivity information without explicitly tracing the white matter fibers. Even when they are traced and

displayed as lines or tubes the resulting image can be visually cluttered. By leveraging the power of the graphics processing unit (GPU) we have developed methods that make it possible to visualize scalar connectivity values computed from DT-MRI at interactive rates.

The diffusion tensor described up to this point is a 2nd-order tensor which can be represented by a matrix. However, diffusion is better described by a higher-order tensor in bifurcating or intersecting white-matter fiber bundles in the brain. The price of this improved model is increased memory requirements and computational complexity. A 2nd-order tensor,D_{ij} , has $3^2 = 9$ elements in 3 dimensions, while a 4th-order tensor, D_{ijkl}, has $3^4 = 81$ elements. Those numbers fall to 6 and 15 respectively when the tensor is constrained to be symmetric.

Efficient processing of high-order tensor images and effective visualization techniques will aid medical clinicians in the diagnosis and treatment of disease by enhancing the ability to assess white matter structure in the brain. This will benefit studies on aging, Alzheimer's disease, multiple sclerosis, schizophrenia, traumatic brain injury, and many other conditions.

2 Previous Work

2.1 Diffusion Tensor MRI

The goal of most DT-MRI analysis is to determine the integrity and trajectory of white-matter pathways. The problem of determining the presence or absence of white matter is commonly solved using scalar measures of diffusion magnitude and anisotropy. Fractional anisotropy (FA) is a scalar quantity that can be efficiently computed from the 2nd-order diffusion tensor at each voxel and is robust to noise in the underlying data. FA has been used to assess tissue damage after stroke, but as a strictly local measure it does not indicate the regions of the brain where connectivity has been changed.

Tractography is the process of computing white-matter fiber pathways. Early approaches to tractography were similar to streamline tracing techniques in vector field visualization. By repeatedly stepping in the direction of highest diffusion, axonal fiber tracts can be estimated. However, tractography becomes unreliable in regions where anisotropy becomes low. In 2nd-order DT-MRI anisotropy may be low due to the absence of white matter or due to crossing or branching fibers. Regularizing assumptions about the underlying fiber smoothness may help tracking through small areas of low anisotropy. Stochastic tractography can generate a large population of feasible fibers while taking uncertainty due to noise into account. The connectivity mapping approach that we present relies on stochastic tractography as a means of estimating the probability of connectivity between two regions of the brain.

2.2 High-Order Tensor Models

The Gaussian diffusion process assumed by Basser et al. [2] in the development of DT-MRI is characterized by a symmetric, positive-definite matrix. But

diffusion within biological structures can be non-Gaussian, for example, when bifurcating or intersecting white-matter fiber bundles occur in the brain. So high angular resolution diffusion imaging (HARDI) techniques have been developed to overcome this weakness of the 2nd-order tensor model. Approaches include model-free techniques which compute diffusivities along a large number of directions [3], mixtures of unimodal distributions [4] and higher-order tensors (order > 2) [5]. Figures (1) and (2) demonstrate the ability of the 4th-order tensor model to resolve multiple fibers in a single voxel.

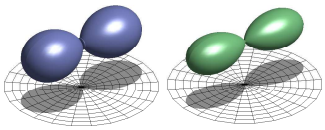

Fig. 1. Order 2 (*left*) and 4 (*right*) tensor diffusivity profiles in a voxel containing a single fiber orientation. Diffusivity is plotted as a displaced sphere where large radius denotes high diffusivity

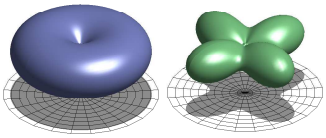

Fig. 2. Order 2 (*left*) and 4 (*right*) tensor diffusivity profiles in a crossing fiber voxel

The ability of the 4th-order model to discriminate multiple fibers can also be seen in the polynomial models for diffusivity d as a function of direction v using each tensor. The second-order tensor represents

$$d(v) = \sum_{i=1}^{3}\sum_{j=1}^{3} D_{ij} v_i v_j \tag{1}$$

which is a second-degree homogeneous polynomial in the components of v. The fourth-order tensor represents

$$d(v) = \sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{k=1}^{3}\sum_{l=1}^{3} D_{ijkl} v_i v_j v_k v_l \,, \tag{2}$$

a fourth-degree polynomial which can have more local maxima and minima than the second-degree model.

The 4th-order tensor model has recently been studied by Barmpoutis et al. [1] and others for representing not only diffusivity, but also the fiber orientation distribution function (ODF). In 2nd-order DT-MRI the probable fiber orientation coincides with the direction of peak diffusivity, but in HARDI this is no longer the case. For high-order tensors anisotropy cannot be characterized using FA, but other measures, such as generalized anisotropy (GA) [6] can be used, as shown in Figure (3).

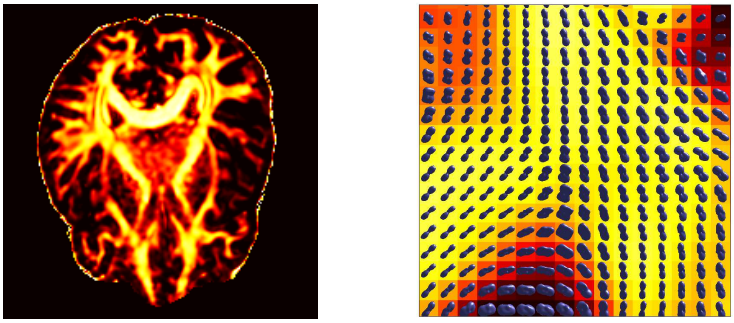

Fig. 3. Generalized anisotropy in an axial slice of the human brain (*left*) and detail of 4th-order fiber ODFs (*right*)

2.3 GPU Processing

The connectivity approach described in [7] was implemented in pure OpenGL since general-purpose GPU (GPGPU) APIs, like CUDA [8], were new at the time of publication. So features like render-to-texture, additive alpha blending were used to achieve the computation of the connectivity map. In subsequent years APIs such as CUDA and OpenCL became more mature and greatly simplified the process of writing code for the GPU. In turn, the solutions to many medical image processing problems have been accelerated by parallelization on the GPU. For an overview, see Eklund et al. [9].

3 Methods

In this work we take the approach of using new OpenGL 4 features [10] to implement the connectivity mapping. Compute shaders are a portable, lightweight way of quickly adding general-purpose calculation to a graphical application. By contrast, CUDA is available only on Nvidia GPUs. OpenCL [11] is portable and makes it possible to share some OpenGL buffers with the GPGPU code, but the API is heavyweight since it supports many devices besides GPUs. OpenGL compute shaders can make direct use of OpenGL buffer objects, they can be

written in familiar GL shading language (glsl), and don't require the overhead of initializing another API context. We also make extensive use of shader image load/store functionality which permits read/write access to texture data, and atomic counters which make it possible to accumulate and integrate values across multiple shader invocations. Whereas the previous approach [7] required 3 rendering passes to evaluate one iteration of connectivity computation, our method permits it to be computed in one compute shader pass. Unlike previous approaches, ours permits mapping from high-order tensor fields which will improve the accuracy in regions of the brain with complex fiber geometry. We utilize modern OpenGL compute shaders to enable our method to be tightly integrated with a custom raycast visualization method.

3.1 Data Acquisition

Our technique requires no changes to the image acquisition process. The data for our experiments were acquired on a 3.0 Tesla General Electric Medical Systems Horizon LX imaging system with a diffusion weighted spin echo pulse sequence. Imaging parameters were : effective TR = 9000 ms, TE = 78 ms, NEX = 1. Diffusion-weighted images were acquired with 25 different gradient directions with b = 1000 s/mm^2 and a single image was acquired with b $\approx$ 0. The image field of view was 24×24 cm and the acquisition matrix was $256 \times 256 \times 30$ voxels.

From the 26 images we computed 4th-order fiber orientation tensors using the method described by Weldeselassie et al. [12].

3.2 Connectivity Model

As in [13] and [7] we use a Bayesian model of fiber probability. In the Bayesian framework the prior distribution models the smoothness constraint on the fibers, and the likelihood function models the dependence of fiber displacements on the ODF data, C. The axonal fibers we track are modeled as a sequence of vertices, x_i, and displacements, v_i, such that $x_{i+1} = x_i + v_i$ as shown in Figure (4). As we generate the sequence of displacements, we take into account $C(x_i)$ computed from the diffusion tensor at the vertex x_i and the previous displacement v_{i-1}.

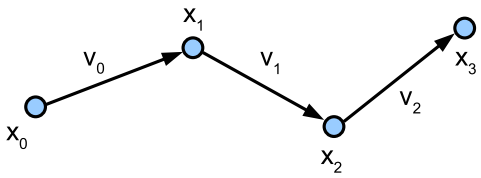

Fig. 4. Vertices, x_i, and displacements, v_i, along a simulated fiber path

The prior probability is written as $p(v_i|v_{i-1})$ which denotes the probability of observing displacement v_i given that the previous displacement was v_{i-1}. We

wish to impose a smoothness constraint on the fibers, so the prior should have high probability that $v_i \approx v_{i-1}$, and low probability that $v_i \cdot v_{i-1} <= 0$. This behavior can be modeled using a von-Mises Fisher (vMF) distribution, which is similar to a Gaussian wrapped around a sphere. It is given by

$$p(v_i|v_{i-1}) = \frac{1}{Z_{vmf}} \exp(\kappa \frac{v_i}{||v_i||} \cdot \frac{v_{i-1}}{||v_{i-1}||}) \tag{3}$$

where $\kappa \geq 0$ is the concentration parameter which controls the variance of the distribution about the mean direction $v_{i-1}/||v_{i-1}||$, and Z_{vmf} is a normalizing constant. Plots of sample distribution for several values of κ are shown in Figure (5).

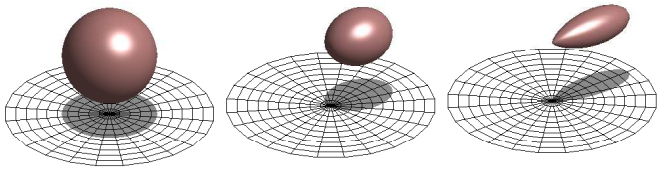

Fig. 5. von Mises-Fisher prior distributions for $\kappa = 0$ (*left*), 2 (*middle*), 10 (*right*)

Our likelihood model comes from a sharpened version of the computed fiber orientation tensor. The sharpening process [14], implemented here by raising the computed fiber probability to a positive exponent $p_{sharp}(v_i|C) = (1/Z_{sharp})p(v_i|C)^m$, reflects the distinction between water molecule displacement probability and fiber orientation probability. Due to intracellular diffusion, water molecules may diffuse perpendicular to the fiber direction, leading to tensors that are more isotropic than they should be. Additionally, there is a limit to the sharpness of the fiber ODF that can be modeled by a 4th-order tensor, since that tensor is actually representing a 4th-degree homogeneous polynomial. In our experiments we use $m = 2$ to suppress the low probability directions of the likelihood and emphasize the peaks of the distribution, as shown in Figure (6).

The likelihood function $p(v_i|C)$ describes our observations of displacements given the fiber orientation tensor, C. Applying Bayes' rule, the posterior probability of observing v_i given both C and v_{i-1} is given by

$$p(v_i|C(x_i), v_{i-1}) = \frac{p(v_i|v_{i-1})p(v_i|C(x_i))}{p(v_i)} \tag{4}$$

where $p(v_i)$ is a constant of proportionality.

Fibers are stochastically generated by initializing fiber points x_0 within a user-specified region of interest (ROI) and repeatedly drawing from the posterior Equation (4) using rejection sampling using the prior as a proposal distribution.

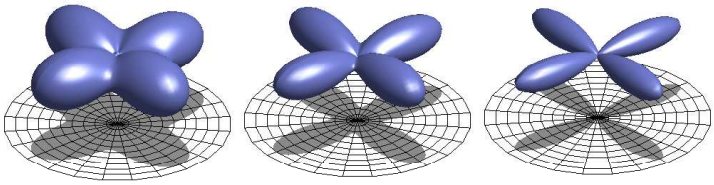

Fig. 6. Sharpened fiber ODFs with $m = 1$ (*left*), 2 (*middle*), 4 (*right*)

If N fibers are initialized in the ROI, the measure of connectivity between the ROI and a voxel x is given by $n(x)/N$ where $n(x)$ is the number of fibers passing through x.

3.3 Connectivity Implementation

The connectivity model described in the previous section was implemented in an OpenGL compute shader. Due to space restriction we cannot print the full shader source here but instead we present a high-level unoptimized summary below.

```
ivec2 ix = ivec2(gl_GlobalInvocationID.xy);
vec4 x0 = imageLoad(fiberXTex, ix);
vec4 v0 = imageLoad(fiberVTex, ix);

vec4 v1 = draw_from_posterior(x0, v0);
vec4 x1 = x0 + v1;

// reinitialize if outside image or below anisotropy threshold
if(reinit_needed(x1))
{
   x1.xyz = seed_pos;
   v1.xyz = vec3(0.0);
   atomicCounterIncrement(N);
}

imageStore(particlePosTex, ix, x1);
imageStore(particleVelTex, ix, v1);

// accumulate connectivity if we stepped into a new voxel
ivec3 x0_vox = ivec3(round(worldToConnectVox*x0.xyz));
ivec3 x1_vox = ivec3(round(worldToConnectVox*x1.xyz));
if(x0_vox != x1_vox)
{
   imageAtomicAdd(connectivityTex, x1_vox, 1);
```

```
}
```

The fiber positions and displacements are stored in 32 bit floating point buffers and bound to image units so that the shader can read and write to the buffers. The connectivity map is a 3D unsigned integer buffer also bound to an image unit. Since each shader invocation reads then writes a unique memory location we do not need expensive memory barrier or synchronization mechanisms for the fiber updates. The connectivity updates and total fiber count, N, computations use atomic operations, so they are also free from memory conflicts. Data which are only read are bound as textures: the tensor components, generalized anisotropy, and a buffer of random values which are used to emulate a random number generator on the GPU.

3.4 Visualization

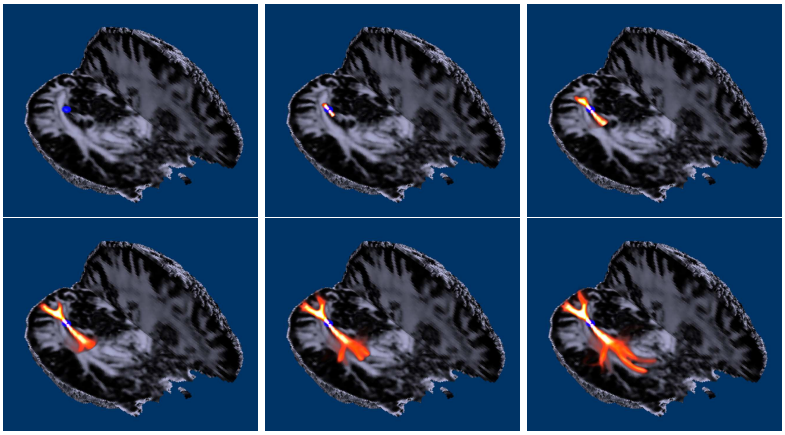

Fig. 7. Connectivity mapping evolution during computation from iteration 1 (*top-left*) to 100 (*bottom-right*)

Visualizing connectivity is not useful without some anatomical reference. Tractography and connectivity mapping often use the b $\approx$ 0 image or anisotropy as context for visualization. In our results we display raytraced anisotropy (GA) with a user controllable clip box for reference. Since we use GA during computation to compute the prior probability of fiber direction we don't need to upload any additional data for visualization. The GA image is displayed with no texture filtering to give the user a true impression of the resolution of the underlying data. The connectivity data is overlaid with GA using a hot color palette. Voxels below a specified threshold are transparent, and increasing connectivity progresses through the red, orange, yellow, white color sequence. An iteration of connectivity mapping is interleaved between frames of volume rendering so

that the user may watch the connectivity map evolve and change the viewpoint interactively.

4 Results

In this section we present images of the resulting connectivity maps, a table of computation times and comparisons between results computed from order 2 and order 4 data. Our connectivity mapping application was implemented in C++ and executed on 2 systems. System 1 was a Dell Optiplex workstation with a 3.4 GHz Intel Core i7-3770 CPU, 8 GB RAM, Nvidia GeForce GTX 750 Ti with 640 shader cores and 2 GB GDDR5 dedicated video RAM. System 2 was a Surface Pro 2 Tablet with a 1.6 GHz Intel Core i5-4200U CPU, 8 GB RAM, Intel HD Graphics 4400 with 20 shader cores and 1793 MB shared video RAM.

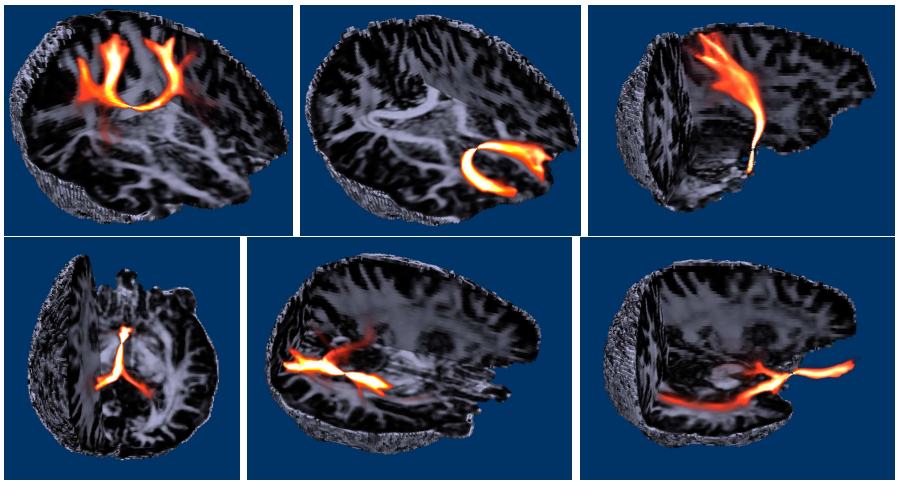

Fig. 8. Connectivity mapping in corpus callosum (CC) midbody (*top-left*), CC genu (*top-middle*) and corticospinal tracts (*top-right*), fornix (*bottom-left*), inferior longitudinal fasciculus (*bottom-middle*), occipitofrontal fasciculus (*bottom-right*)

Figure (7) shows 6 iterations of connectivity map computation. The connectivity maps in Figure (8) were seeded in a single fiber and run for 350 iterations. The corpus callosum (CC) connects the left and right hemispheres of the brain as clearly reflected in the connectivity map. Our maps also capture the fanning behavior of the corticospinal tracts as they approach the cortex and correlate well with known neuroanatomy.

All connectivity maps were computed by simulating 262000 fiber tracts per iteration. System 1, the desktop workstation, required 11.4 ms per iteration of the compute shader. System 2, while slower at 63.1 ms, was capable of running our algorithm at usable rates. This demonstrates that our technique is portable

across GPU vendors and also that high-end hardware is not required to take advantage of GPU-accelerated algorithms.

5 Conclusions

We have presented a fast high-order diffusion tensor MRI connectivity mapping algorithm and developed an implementation of this algorithm in modern OpenGL. Our implementation using compute shaders is portable across Nvidia and Intel GPUs. We have demonstrated that this model can generate plausible connectivity maps, even in regions of bifurcating and crossing fiber tracts. Future work will involve validation of connectivity results and incorporation of anatomical prior information.

References

1. Barmpoutis, A., Hwang, M.S., Howland, D., Forder, J.R., Vemuri, B.C.: Regularized positive-definite fourth order tensor field estimation from DW-MRI. NeuroImage **45** (2009) S153–S162
2. Basser, P.J.: Inferring microstructural features and the physiological state of tissues from diffusion-weighted images. NMR in Biomedicine **8** (1995) 333–344
3. Tuch, D.S., Reese, T.G., Wiegell, M.R., Wedeen, V.J.: Diffusion MRI of complex neural architecture. Neuron **40** (2003) 885–895
4. McGraw, T., Vemuri, B.C., Yezierski, B., Mareci, T.: Von Mises-Fisher mixture model of the diffusion ODF. In: 3rd IEEE International Symposium on Biomedical Imaging: Nano to Macro, 2006, IEEE (2006) 65–68
5. Özarslan, E., Mareci, T.H.: Generalized diffusion tensor imaging and analytical relationships between diffusion tensor imaging and high angular resolution diffusion imaging. Magnetic Resonance in Medicine **50** (2003) 955–965
6. Özarslan, E., Vemuri, B.C., Mareci, T.H.: Generalized scalar measures for diffusion MRI using trace, variance, and entropy. Magnetic Resonance in Medicine **53** (2005) 866–876
7. McGraw, T., Nadar, M.: Stochastic DT-MRI connectivity mapping on the GPU. IEEE Transactions on Visualization and Computer Graphics **13** (2007) 1504–1511
8. Nvidia: CUDA programming guide, version 6.0. Nvidia Corporation, February (2014)
9. Eklund, A., Dufort, P., Forsberg, D., LaConte, S.M.: Medical image processing on the GPU–past, present and future. Medical Image Analysis **17** (2013) 1073–1094
10. Shreiner, D., Sellers, G., Kessenich, J.M., Licea-Kane, B.M.: OpenGL programming guide: The Official guide to learning OpenGL, version 4.3. Addison-Wesley Professional (2013)
11. Khronos OpenCL Working Group, Edited by Munshi, A.: The OpenCL specification version: 1.0 (2014)
12. Weldeselassie, Y.T., Barmpoutis, A., Stella Atkins, M.: Symmetric positive semi-definite Cartesian tensor fiber orientation distributions (CT-FOD). Medical Image Analysis **16** (2012) 1121–1129
13. Friman, O., Farneback, G., Westin, C.F.: A Bayesian approach for stochastic white matter tractography. IEEE Transactions on Medical Imaging **25** (2006) 965–978
14. Descoteaux, M., Lenglet, C., Deriche, R.: Diffusion tensor sharpening improves white matter tractography. In: Medical Imaging, SPIE (2007)