Interactive Volume-Based Visualization and Exploration for Diffusion Fiber Tracking

Dominik Sibbing, Henrik Zimmer, Robin Tomcin, Leif Kobbelt

Computer Graphics and Multimedia, RWTH Aachen University
sibbing@cs.rwth-aachen.de

Abstract. We present a new method to interactively compute and visualize fiber bundles extracted from a diffusion magnetic resonance image. It uses Dijkstra's shortest path algorithm to find globally optimal pathways from a given seed to all other voxels. Our distance function enables Dijkstra to generalize to larger voxel neighborhoods, resulting in fewer quantization artifacts of the orientations, while the shortest paths are still efficiently computable. Our volumetric fiber representation enables the usage of volume rendering techniques. Therefore no complicated pruning or analysis of the resulting fiber tree is needed in order to visualize important fibers. In fact, this can efficiently be done by changing a transfer function. Our application is highly interactive, allowing the user to focus completely on the exploration of the data.

1 Introduction

Diffusion Magnetic Resonance Imaging (dMRI) is a non invasive imaging method, that measures the diffusion of water within living tissue. Fibers, membranes and other molecular structures are pipelines or obstacles for the water molecules and thus influence their movements into certain directions, i.e. water majorly flows along fibers. Diffusion MRI measures this flow and translates these measurements to oriented distribution function (ODFs), which locally characterize the probability of water flowing into a certain direction. We use Diffusion Tensor Imaging (DTI) to demonstrate the effectiveness of the suggested pipeline, but a generalization to High Angular Resolution Diffusion Imaging (HARDI) such as Q-Ball Imaging [1] is just a matter of adapting a distance function.

To better understand the connectivity within the human brain, the ultimate goal is to reconstruct the "true" fiber pathways from these ODFs. We present an efficient method to compute and visualize such globally connected fiber tracts based on Dijkstra's shortest path algorithm and represent the resulting pathways in a volumetric fashion. Related approaches typically use 26 neighbors defining the edge relation of Dijkstra's graph, leading to discretization artifacts, especially problematic for HARDI data. We explore the usage of larger neighborhoods and thereby reduce such artifacts. Our approach efficiently integrates the ODF along

This work was partially supported by the DFG (IRTG 1328)

a path between neighboring voxels and controls the directional sensitivity of the propagated front by a single parameter.

The idea of our volumetric fiber representation is to count the number of pathways (computed during Dijkstra) that run through a given voxel. We adapt a transfer function, commonly used in volume visualization, to continuously blend in and out fibers instead of tediously trace and prune the obtained tree structure.

2 Related Work

Classical tractography traces possible nerve tracts starting from a seed position. Early deterministic approaches are the streamline tracking technique (STT) and tensor deflection (TEND) [2]. A number of *probabilistic* tractography techniques have been presented based on modeling and sampling an ODF in each voxel [3,4]. Difficulties in probabilistic tractography mainly involve the reported running times [5], which can range to 24h using Monte Carlo-based sampling [3] or need to be implemented on modern GPUs to reduce the computation time [6]. Some recently proposed tracking algorithms are based on a weighted graph-structure [5,7]. Using shortest path algorithms enables globally optimal, deterministic results and allows to bridge tract connections otherwise lost in other approaches. As noted in [5] more than 26 neighboring voxels may be required to correctly capture fiber tract orientations, which leads to a computationally more involved tracing approach, but no practical details on implementational aspects or analysis of different neighborhoods are given. The most related work to our approach is [7]. The main difference is that we use physically motivated weights and allow for arbitrarily sized neighborhoods, which are efficiently integrated in Dijkstra's algorithm. Additionally we do not require any heuristics for pruning the shortest path tree. Simultaneously visualizing all pathways in 3D, often affects the clarity of the illustration. To enhance explorability, some approaches filter the computed tracts [8,7] in a preprocessing step. Additionally visualizing scalar properties such as uncertainties, densities or distances from a seed is often limited to separate 2D opaque images [5,7]. 3D volume rendering has been used to render local scalar quantities, such as uncertainty of probabilistic tractography pathways [9], or local tensor orientations [10]. Our visualization is also based on volume rendering. Since, highlighting important fibers requires only the adaption of a transfer function, which can be done in real time. The visualized information is purely based on the shortest path tree and thereby follows up on the future work posed in [7] to further exploit the Dijkstra information for a better visualization. For an exhaustive overview of dMRI we refer to the survey [11].

3 Fiber Extraction Pipeline

Our approach to transform the information given as local diffusion tensors into globally connected pathways representing the fibers is based on Dijkstra's shortest path algorithm. This algorithm takes an edge weighted graph $G = (V, E)$ as input, starts from a given (set of) seed node(s) and sequentially conquers the

remaining nodes of the graph in the order of increasing distances to the seed (cf. Fig. 1). Since a predecessor map $pre[v]$ is stored for each node v, it is always possible to trace back a unique path to the seed. In our setting the nodes of the graph are a subset of the DT image voxels and the weight between two neighboring nodes shall be small if the ODF indicates a large flow between them. To represent the fibers we use the resulting predecessor map to trace all possible pathways back to the seed and thereby count for every voxel the number of pathways that run through it. We use volume rendering techniques to highlight regions with a high count value, i.e. regions containing many fibers.

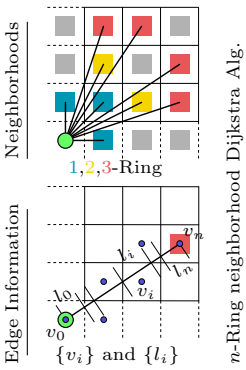
3.1 Tracing the DTI Volume

The simultaneous motion of water molecules in different directions (Brownian motion) can be described by a tensor $T \in \mathbb{R}^{3\times3} = U \cdot D \cdot U^T$, where U is an orthogonal matrix storing the main diffusion axes column-wise and $D = diag(\lambda_1, \lambda_2, \lambda_3)$ is the diagonal matrix of eigenvalues of T. Given a direction $\mathbf{r}$ this tensor can be used to compute the flow of water into this direction $f(\mathbf{r}) = \mathbf{r}^T \cdot T \cdot \mathbf{r}$.

Nodes of G. To restrict our computations to the white matter of the human brain we define the nodes V of G to be those voxels with a fractional anisotropy (FA) > 0.1 [8]. This filters out voxels with no pronounced fiber directions.

Edges of G. When defining the connectivity of G and thereby the set E of edges, one typically uses the 26 neighbors of the 1-ring of a voxel for simplicity. To more accurately capture the tensor directions in strongly anisotropic regions, we present an efficient way to also handle arbitrary n-ring neighborhoods. Since the goal of using larger voxel neighborhoods is to increase the number of *directions*, we can save memory and compute time by discarding redundant edges whose orientations are already covered by shorter edges (cf. top left image of Fig. 1). When using n-ring neighborhoods with $n \geq 2$ some care has to be taken during Dijkstra when conquering a voxel v. To avoid back and forth motion of the front we need to conquer all voxels between v and $pre[v]$. In a preprocessing step we intersect each possible edge with a smaller grid defining the neighborhood stencil and store a list of visited voxels $\{v_i\}$ and a corresponding list of line segments of length $\{l_i\}$ (cf. bottom left image of Fig. 1). During Dijkstra all voxels between v and $pre[v]$ can then simply be tagged by traversing the respective list $\{v_i\}$.

Edge weights. Imagine the edge e is embedded in a continuous tensor field $T_{\mathbf{x}}$, with $\mathbf{x} \in \mathbb{R}^3$. Intuitively spoken the contribution to the edge distance $d[e]$ of one infinitesimal line segment dt at position $\mathbf{x}$ should be low if the local tensor indicates a strong flow into the edge direction $\mathbf{r}$ ($\|\mathbf{r}\| = 1$). We base the edge distance on the inverse metric tensor and define $d[e] = \int_e \left(\mathbf{b}^T \cdot diag\left(\frac{1}{\lambda_1^\alpha}, \frac{1}{\lambda_2^\alpha}, \frac{1}{\lambda_3^\alpha}\right) \cdot \mathbf{b}\right) dt$, where $\mathbf{b} = U^T \cdot \mathbf{r}$. The introduced parameter α controls the sensitivity to the anisotropy of this flow metric. When increasing α, the front rapidly moves along significant fibers first (cf. Col. 2 of Fig. 2). This integral over an edge is discretized by a Riemann sum with nearest neighbor sampled tensors $\{T_i\}$ of the voxels $\{v_i\}$ intersected along the edge and weighted by the length of the respective segments $\{l_i\}$ (cf. Fig. 1) such that $d[e] \approx \sum_i \left(\mathbf{r}^T \cdot T_i^{-\alpha} \cdot \mathbf{r}\right) l_i$.



n-Ring neighborhood Dijkstra Alg.

Input: A graph $G = (V, E)$ with edge distances $d[e]$ and seed nodes $\{s\}$
Output: A predecessor map $pre[v]$ and a distance map $dist[v]$ $\forall v \in V$
for $v \in V$ **do**
  $dist[v] = \infty$; $pre[v] = -1$;
$dist[s] = 0$; $F = \{s\}$; $C = \emptyset$;
while $F \neq \emptyset$ **do**
  $v_c = \arg\min_{v \in F} \{dist[v]\}$
  $L = \{$ all $v \in G$ intersected by the line between $pre[v_c]$ and $v_c\}$
  $C = C \cup L$; $F = F \setminus L$;
  $N = \{v \in V | (v_c, v) \in E\}$
  for $v_n \in N \cap (V \setminus C)$ **do**
    $dist' = dist[v_c] + d[(v_c, v_n)]$;
    if $v_n \in F$ **then**
      if $dist' < dist[v_n]$ **then**
        $dist[v_n] = dist'$; $pre[v_n] = v_c$;
    else
      $F = F \cup \{v_n\}$; $dist[v_n] = dist'$; $pre[v_n] = v_c$;

Fig. 1. On the top left a quadrant of a 2D visualization of different neighborhood sizes is shown. Omitted, linear dependent directions are color-coded in gray. The information (intersected voxels and lengths) stored for each $e \in E$ is shown on the bottom left.

3.2 Representing Fibers

The unique predecessor relation $pre[v]$ (cf. Fig. 1) defines a tree of globally optimal shortest paths along nerve fibers starting from a seed node. Instead of extracting an explicit tree topology, we represent the structure of this tree in a volumetric fashion by tracing back all *end points* to the seed (*end points* are defined as tree nodes having a predecessor but no successors and are found by building a successor map from the predecessor information). Whenever we traverse a voxel during this process we increase a value stored at that voxel by the respective partial edge length l_i. The result is a volumetric image where voxels have high intensity values if intersected by many pathways.

4 Results

We used a DT image with 128x120x75 voxels, from which we extracted 171041 nodes with FA > 0.1. We simultaneously visualize an anatomical image and the volumetric fiber representation (cf. Sec. 3.2) using direct volume rendering. By designing a transfer function that maps intensities I smaller than a user defined value I_a to blue and opacity 0 and intensities larger I_b to red and opacity 1, we can continuously prune fibers by interactively adapting I_a and I_b. Within the interval we linearly interpolate opacity and hue values. Defining a narrow interval at the right spectrum of intensities only visualizes the main fibers, while a wide interval also reveals fine structures (cf. Col. 4 of Fig. 2). Exploring the volume and simultaneous seed placement using a 3D mouse is possible at high frame rates of about 50 fps on an Intel(R) Core(TM) i7 CPU with 2.67 GHz. Since all computations (cf. Sec. 3) run in a separate thread and updates are displayed immediately after computation, the application is highly interactive.

Since voxels with a high flow rate are conquered first, the last conquered nodes of G usually do not contain many paths. This allows us to speed up the

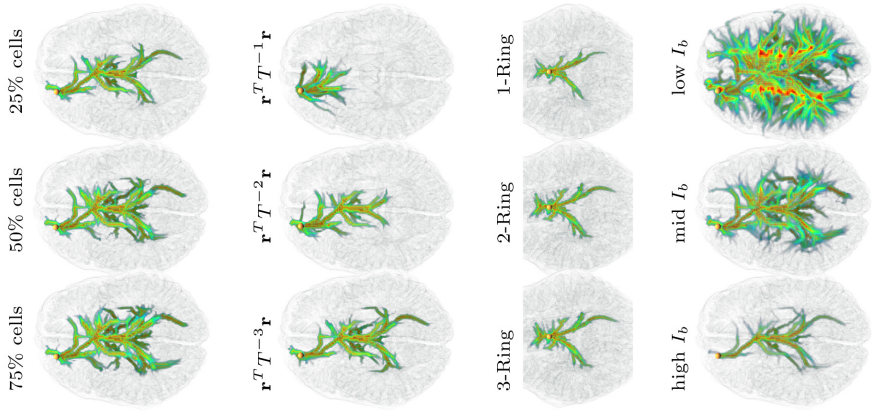


Fig. 2. Connection between the eyes and the visual cortex. Col. 1: Effect of conquering different portions of the graph. Col. 2: Influence of α. Col. 3: Different neighborhood sizes. Col. 4: Visual pruning of the tree by changing the transfer function.

computation by only conquering a certain portion of the graph. Our experiments indicate that all important pathways have been extracted, when aborting the computation after conquering 50% of the nodes (cf. Col. 1 of Fig 2). While

Cells (%)	0-Ring	1-Ring	2-Ring	3-Ring
100	160/160	240/160	420/160	660/155
75	135/110	190/120	350/140	600/155
50	90/80	140/75	255/105	450/118
25	35/28	100/50	105/40	300/70

Table 1. Timings (in ms) for different portions (25% - 100%) and neighborhood sizes (the 0-ring has 6 neighbors).

timings for Dijkstra are mainly affected by the neighborhood size (first numbers in Table 1), the timings for tracing back endpoints (second numbers) are mainly influenced by the amount of endpoints, which depends on the portion of nodes to be conquered. Col. 2 of Fig. 2 shows how the parameter α can be used to influence the anisotropic movement of the front. Keeping α small results in a more spherical expansion pattern while high values for α force the front to quickly follow important paths. In this example we place the seed near the left eye and major pathways end near the visual cortex. Using the 1-ring results in pathways similar to using larger neighborhoods (cf. Col. 3 of Fig. 2). This is good, since such neighborhoods were often used in previous works. Nevertheless, including voxels from the 2-ring reveals more, finer structures. When using larger neighborhoods we only gain a bit more information, so we use the 2-ring by default.

5 Discussion

We presented an efficient method based on Dijkstra's algorithms to compute and visualize globally connected pathways representing fibers originating from

an interactively placed seed point. Compared to earlier approaches we investigated the usage of larger neighborhood sizes to reduce discretization artifacts, presented an efficient way to precompute the edge weights, and implemented the fiber extraction in a separated thread, which enables the interactive exploration of the diffusion data. Since our visualization approach is based on volume rendering, highlighting important fibers is just a matter of adapting a transfer function. We visually validated that the enhancement of the anisotropic movement of the front (parameter α in Section 3.1) and the usage of a larger voxel neighborhood, yields plausible and smooth pathways according to our collaborating partners from the field of medicine.

To visualize specific nerve bundles, we plan to interactively define a target region besides the seed, which requires the user to have some anatomical knowledge. After computing Dijkstra one can automatically select those endpoints, whose paths to the seed intersect the user defined target region. We expect that tracing back (cf. Sec. 3.2) those paths would restrict the visualization to specific connections. In our current system we represent the ODF by a single elliptically shaped tensor. In future work we would like to use arbitrary ODFs to compute the edge distances. Then, this approach also works for HARDI data like Q-Ball Images, which would lead to even more reliable representations of the fibers.

References

1. Tuch DS. Q-ball imaging. Magn Reson Med. 2004 Dec;52(6):1358–1372.
2. Crettenand S, Meredith SD, Hoptman MJ, Reilly RB. Quantitative Analysis and Comparison of Diffusion Tensor Imaging Tractography Algorithms. In: Irish Signals and Systems Conference; 2006. p. 105 –110.
3. Behrens TE, Woolrich MW, Jenkinson M, Johansen-Berg H, Nunes RG, Clare S, et al. Characterization and propagation of uncertainty in diffusion-weighted MR imaging. Magn Reson Med. 2003;50(5):1077–88.
4. Friman O, Farneback G, Westin CF. A Bayesian Approach for Stochastic White Matter Tractography. TMI. 2006;25(8):965–978.
5. Zalesky A. DT-MRI Fiber Tracking: A Shortest Paths Approach. Trans Medical Imaging. 2008;27(10):1458–1471.
6. McGraw T, Nadar MS. Stochastic DT-MRI Connectivity Mapping on the GPU. IEEE Trans Vis Comput Graph. 2007;13(6):1504–1511.
7. Everts MH, Bekker H, Roerdink JBTM. Visualizing white matter structure of the brain using Dijkstra's algorithm. In: Proc. ISPA; 2009. p. 569 –574.
8. Sherbondy A, Akers D, Mackenzie R, Dougherty R, Wandell B. Exploring Connectivity of the Brain's White Matter with Dynamic Queries. IEEE TVCG. 2005;11(4):419–430.
9. Rick T, von Kapri A, Caspers S, Amunts K, Zilles K, Kuhlen T. Visualization of Probabilistic Fiber Tracts in Virtual Reality. In: Studies in Health Technology and Informatics. vol. 163; 2011. p. 486–492.
10. Kindlmann G, , Westin CF. Diffusion Tensor Visualization with Glyph Packing. IEEE TVCG. 2006 sept-oct;12(5):1329 –1336.
11. Tournier JD, Mori S, Leemans A. Diffusion tensor imaging and beyond. Magnetic Resonance in Medicine. 2011;65(6):1532–1556.