Visualization of Neuronal Fiber Connections from DT-MRI with Global Optimization

Nathaniel Fout
Department of Computer Science, The University of California, Davis, CA
2063 Kemper Hall, One Shields Avenue, Davis, CA 95616
530-400-8671 (tel)
530-752-4767 (fax)
nrfout@ucdavis.edu

Jian Huang
Department of Computer Science, The University of Tennessee, Knoxville, TN
203 Claxton Complex, 1122 Volunteer Blvd, Knoxville, TN 37996
865-974-4398 (tel)
865-974-4404 (fax)
huangj@cs.utk.edu

Zhaohua Ding
Institute of Imaging Science, Vanderbilt University, Nashville, TN
1161 21st Avenue South, CCC 1121 MCN, Nashville, TN 37232-2675
615-322-7889 (tel)
615-322-0734 (fax)
zhaohua.ding@vanderbilt.edu

Visualization of Neuronal Fiber Connections from DT-MRI with Global Optimization

Abstract

Diffusion Tensor MRI (DT-MRI) provides valuable 3D data describing diffusion characteristics of water molecules in the human brain. From DT-MRI, it is hoped that neuronal fiber connections among cortical regions can be reliably extracted and adequately interpreted. To achieve this goal, several significant challenges persist. In this paper, by means of dynamic programming we have developed a global fiber reconstruction algorithm enabling efficient visualization of neuronal connections queried by both the start *and* end points on the fly. Besides an inherent ability to handle noisy datasets, our algorithm also naturally addresses situations where neuronal fibers branch or cross each other. We demonstrate the efficacy of our approach with visualization of neuronal connections among activated brain cortical regions detected by functional MRI (fMRI).

Keywords: DT-MRI, fMRI, reconstruction, fiber tracking, optimal pathway, dynamic programming, conditional probability and Bayes rule.

1. Introduction

In recent years advances in medical imaging have produced powerful non-invasive techniques for exploring the human brain. In particular, Diffusion Tensor MRI (DT-MRI) and functional MRI (fMRI) have emerged as potentially revolutionary tools for exploration of brain structure and function, respectively. While the opportunity for new discoveries is great, effective interpretation of these modalities is still a maturing discipline.

Inside the brain, water diffuses according to local structure; in areas such as white matter (WM) water will diffuse linearly in the direction of the fiber tracts. In other areas, such as gray matter (GM), water diffuses isotropically. DT-MRI indirectly measures the directional-dependent motion of water molecules in the brain, producing a set of coefficients which are then used to calculate a symmetric rank-2 tensor. A common geometric representation of this tensor is an oriented ellipsoid, where the surface represents the probability of diffusion in every direction. In order to recover the underlying neuronal fibers, researchers have developed a variety of methods. These methods can be divided into two categories, as observed by S. Mori and P. Zijl in their comprehensive review of fiber tracking [1]. First are line propagation methods, which propagate fibers based on local tensor information. In this category are streamlines and hyper-streamlines [2], as well as tensorlines [3] and several others [4,5,6]. The usual approach is to start at a seed point and step through the volume, with the direction of propagation taken from the local tensor. In general the eigenvector corresponding to the largest eigenvalue of the tensor defines the local fiber orientation, and so most methods essentially reduce the tensor field to a vector field in which well-established vector field integration techniques are applied. The second category includes methods which attempt to find the most energetically favorable path between two points, and are therefore referred to as global methods. Included in this category are the fast marching [7] and simulated annealing [8] methods, both of which find optimal connections between points in the vector field of major eigenvectors.

2. Global Fiber Tracking

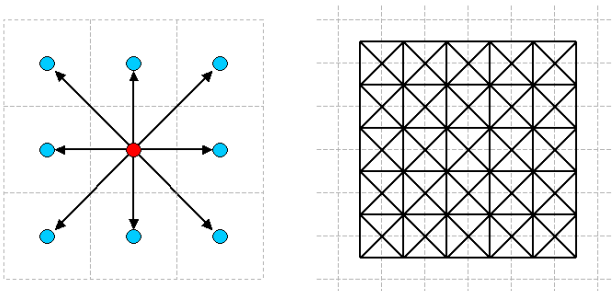
2.1 Overview

Constructing neuronal fibers in DT-MRI is a challenging endeavor in general. Signal noise and partial volume effects (PVE) are the two primary obstacles. Signal noise could cause significant deviation of the major eigenvector of the tensor matrix from the underlying fiber direction, thus misguiding the tracking process. This phenomenon is especially problematic for line propagation techniques, since errors accumulate as propagation proceeds. PVE is a consequence of limited imaging resolution. A single voxel may contain hundreds of individual fibers; the tensor matrix acquired for a voxel is representative of the average fiber orientations within the voxel. This fact can lead to poorly defined major eigen-directions in voxels wherein fibers cross or branch. Similar to noise effects, PVE can also mislead tracking but in a more unpredictable manner.

In order to overcome the limitations inherent in fiber tracking based on local information, we propose a method that integrates fiber segments within each voxel while constructing complete fibers using global optimization. By using global information we create a context within which questions regarding connectivity can be answered. The piecewise construction of fiber segments eliminates the accumulation of error along fiber paths. In global optimization, instead of reducing the tensor field to a vector field, our method operates directly on the tensor field, leveraging Bayes rule to rigorously evaluate the probability of a connection's existence considering all possible directions in which the fiber may enter and leave each voxel.

The algorithm begins by constructing fiber segments between voxels. For each voxel hypothetical segments are created which connect the voxel to its 26 neighbors, as illustrated in Figure 1a. We then evaluate the probability that these fiber segments exist based on the local tensors. Each segment is marked with its corresponding probability of existence, and the process is repeated for each voxel. At

this point we end up with a connected graph in 3D space, with the nodes as voxels and the edges as plausible fiber segments (Figure 1b). We can now perform operations on this graph, such as finding the path of highest probability connecting two points.



(a) (b)

Figure 1. 2D views of (a) discrete fiber segments connecting voxels, and (b) the graph resulting from segments joining all voxels in the volume.

In summary, the algorithm proceeds in four steps:

1. For a given voxel, construct segments connecting it to the 26 nearest neighbors.
2. Evaluate the probability of each segment's existence.
3. Repeat for all voxels to construct a 3D graph.
4. Take two points as input and query graph to find the most likely fiber(s) connecting the points.

While constraining the fiber paths to lie along these discrete segments may produce fibers, which locally vary from the underlying physical fibers, the global topology of the fibers is unaffected (see Figure 2). We construct final smooth fibers between the two points by using the approximated path from the graph as a guide for conventional tracking using line propagation.

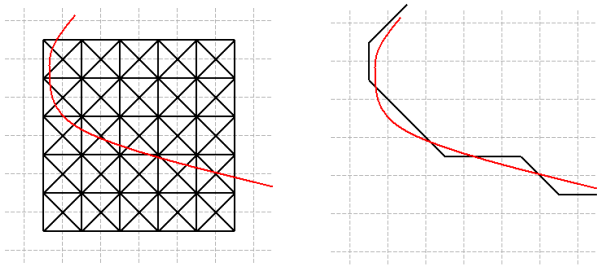


Figure 2. Approximation of a physical neuronal fiber (red) by graph edges (black) in 2D.

2.2 Fiber Segment Probabilities

In order for our method to construct fibers which reflect the underlying physical fibers, it is necessary that we obtain an accurate estimate of the probability of a given fiber segment connecting two voxels. This probability is certainly a function of the local tensor, but how should it be computed? Looking to line propagation methods will not help, because these methods generally do not provide a probability for each direction, but rather the direction with the maximum probability. We could directly use the local tensor, which provides a probability distribution function (pdf) capable of answering the question, but PVE will still be a problem in regions where fibers cross or branch.

In order to address this problem we developed a framework based on conditional probabilities, similar to that used in [9]. The idea behind conditional probabilities is that of updating an estimate of the current probability based on past information. More specifically, conditional probability allows us to better calculate the probability of an event D_i occurring given the fact that another event D_k has occurred. If we further consider the events D_k and D_i as members of an event space U of size n containing many events D, then the conditional probability $P(D_i|D_k)$ for any D_k and D_i in U can be computed using Bayes Rule:

$$P(D_i|D_k) = \frac{P(D_i)P(D_k|D_i)}{\sum_{j=1}^{n} P(D_j)P(D_k|D_j)} \qquad (1)$$

The analogy to computing fiber segment probabilities can be found by letting the event space U be all possible directions of fiber propagation. The event D_i is the fiber following direction D_i, and thus the term $P(D_i|D_k)$ is the probability of a fiber taking the outgoing direction D_i given that it came from direction D_k.

The term $P(D_i)$ is the unconditional probability of a fiber following direction D_i and can be taken directly from the pdf given by the local tensor. A convenient visualization of this probability profile can be obtained by plotting probability as a function of the two spherical angles θ and ϕ. This creates a surface representing the probability as a function of direction, as shown in Figure 3.

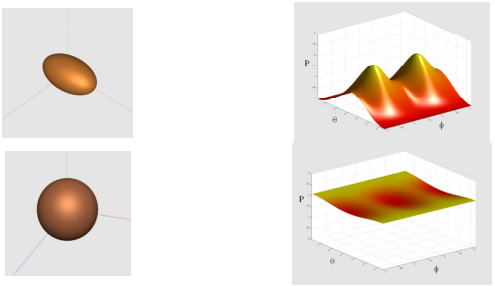


Figure 3. Two different tensors represented as ellipsoids (left) and pdfs (right).

The term $P(D_k|D_i)$ is the probability of a fiber entering from direction D_k, given that it leaves in direction D_i. Intuitively this term is related to the concept of bending energy; that is, the energy needed to bend the fiber from D_k to D_i. The selection of $P(D_k|D_i)$ can therefore be made based on fiber modeling. If we assume fibers to be somewhat stiff, then $P(D_k|D_i)$ will have a maximum probability along the incoming direction (i.e. no bending)

and decrease as the angle widens. The profile of this fall-off will determine the stiffness of the fibers. In practice we choose a smooth profile like a Gaussian or elevated cosine, which allows us to use a single parameter to vary the profile. The cosine-shaped profile is favored due to the simplicity of its computation (a single dot product).

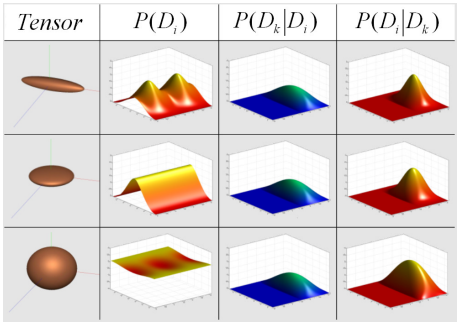


Figure 4. Application of Bayes Rule to different types of tensors using a cosine profile for $P(D_k|D_i)$. The input direction is the z-axis (red line in leftmost column). The top tensor is prolate (spindle-shaped), the middle tensor is oblate (disc-shaped), and the bottom tensor is spherical.

Once the selection of the profile of $P(D_k|D_i)$ has been made then the conditional probabilities $P(D_i|D_k)$ can be computed for each outgoing direction D_i given an incoming direction D_k. Figure 4 shows the resulting pdfs for the three types of tensors (prolate, oblate, and spherical) using a cosine profile for $P(D_k|D_i)$. Notice that the ambiguity in the case of oblate and spherical tensors is resolved.

		j				
		1	2	…	25	26
i	1	$P_{1,1}$	$P_{1,2}$	…	$P_{1,25}$	$P_{1,26}$
	2	$P_{2,1}$	$P_{2,2}$	…	$P_{2,25}$	$P_{2,26}$
	⋮	⋮	⋮		⋮	⋮
	25	$P_{25,1}$	$P_{25,2}$	…	$P_{25,25}$	$P_{25,26}$
	26	$P_{26,1}$	$P_{26,2}$	…	$P_{26,25}$	$P_{26,26}$

Figure 5. A PCM is kept for each voxel, indicating the probability P_{ij} of a fiber entering from direction I and leaving in direction j.

The use of Bayes Rule in calculating probabilities for fiber segments requires considering both incoming and outgoing directions. On each voxel, for each outgoing direction (pointing to one of its 26 neighbors), we compute and store 26 probabilities, each correspond to a possible incoming direction (again, from one of its 26 neighbors). Obviously, if a fiber cannot double back there would be only 25 possible incoming directions for each outgoing direction. However, for regularity in storage, we still store 26 probabilities for each outgoing direction. This forms a probabilistic connection map (PCM) to be stored for each voxel (see Figure 5). A PCM is a 26x26 matrix. Entry p_{ij} in the PCM is the probability of the fiber leaving in direction j, given that it arrived from direction i. In practice many of these probabilities will be close to zero, allowing compressing PCM with classic schemes developed for sparse matrices.

In summary, we can leverage Bayes rule to estimate the probabilities for each fiber segment. In our implementation we use the cosine or Gaussian function for the $P(D_k|D_i)$ term. Finally, we note that filtering techniques could be used with our algorithm to further reduce noise by modifying the local tensor pdf directly.

2.3 Domain Restriction

As our goal is to derive fiber pathways through the WM of the brain, it would be beneficial to limit the extent of the graph to WM regions only. In this way the size of the graph is greatly reduced, thereby aiding the computationally expensive step of searching for probable fiber paths as well as the time to construct the graph. In our implementation we use a simple region-growing scheme based on anisotropy to segment the WM areas from other regions (see Figure 6).

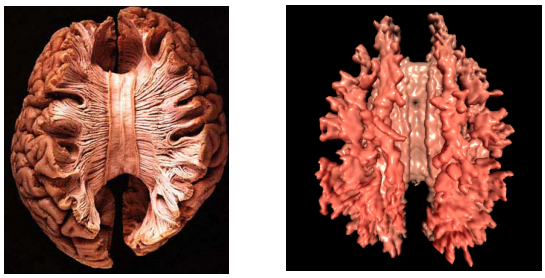


Figure 6. Picture taken of actual WM in the brain (left). Reconstructed WM resulting from region-growing using anisotropy for thresholding, followed by volume rendering (right).

2.4 Discovering Fiber Paths using Dynamic Programming

Once the segments connecting each voxel are assigned probabilities, the resulting graph may then be used to query for specific connections. A statement of the problem is:

Given two end points, S and E, are there plausible fiber(s) connecting S to E, and if so, what are those?

This is an especially difficult problem to answer, considering that the probability of an edge leaving a node in the graph depends on which edge was taken to get to the node in question. A brute-force approach is to consider all combinations of paths between S and E, taking the one(s) whose segments' probabilities give the largest product. As is, this approach is computational intractable; however, we can leverage the paradigm of dynamic programming to compute the optimal solution in polynomial time. To

further speed up the computations, we make a few reasonable simplifications in the problem.

First, although the graph itself is cyclic, we do not allow cycles in fibers. This means paths cannot double back, nor can they pass through the same voxel twice. Second, we restrict the length of fibers to be not too much longer than the Euclidean distance between S and E. How much longer is a controllable parameter. Third, paths leaving the WM or the volume boundaries are terminated. Finally, we assume that the most probable path from a voxel to a directly adjacent voxel is the segment connecting them directly.

Despite these simplifications the problem is still large enough to require extensive computational resources. We observe that a path of highest probability will consist of subpaths of highest probability, although the overall optimal solution would only consist of a subset of the entire set of optimal subpaths. These subpaths are not independent (subpaths may share other subpaths), thus leading to a plausible solution by means of dynamic programming. Dynamic programming allows us to reuse previously computed subpaths by storing them in a table. In this way we achieve significant computational savings, albeit at the expense of increased storage.

In order to use dynamic programming we recast the goal of finding probable fibers as an optimization problem having many potential solutions, each with an associated "cost". The process of finding an optimal solution involves a series of decisions. In our case the decision is which segment to follow next in the construction of a neuronal path. The cost of our decision is simply the probability of that segment. In this way we construct a potential solution as a choice of connected fiber segments which taken together yield a probability for the entire path. In order to compute the potential solutions we subdivide the problem recursively using the following equation:

$$P(S,E) = \max_{\forall S', n(S)} \{P(S,S') + P(S',E)\} \qquad (4)$$

where n(S) denotes the set of neighbors of S such that ||n(S)|| = 26. There are two base cases for terminating the recursion: P(S,S'), which is taken directly from the corresponding PCM table, and P(E,E), which is by definition zero. In dynamic programming, we compute each subpath P(S',E) once, obtaining significant savings.

While the metric of optimality in this equation is the probability of fiber segments, it is trivial to incorporate other information in the definition of an optimal fiber, such as diffusion rate or curvature. Ideally, our definition of optimality should be based on knowledge of real fibers. Since this information is not available, we here use as few assumptions as possible and therefore our implementation simply uses a segment's probability of existence as the metric of optimality.

```
// find optimum path: S→E
targetpath = Path(S,E);
stack.push(targetpath);

while(stack.notempy()) {
  path = stack.top();

  // check if already computed
  if (table.find(path))
    stack.pop();
    continue;

  // need all path: s→n(s) and path: n(s)→E
  haveAllSubPaths = true;
  for (n : neighbor(s)) {
    if (!table.find(Path(s,n)))
      table.insert(graph.getedge(s,n));
    if (!table.find(Path(n,E)))
      stack.push(Path(n,E));
      haveAllSubPaths = false;
  }
  if (!haveAllSubPaths) continue;

  // have all subpaths necessary to calc opt path: s→E
  for (n : neighbor(s)) {
    p0 = table.find(Path(s,n));
    p1 = table.find(Path(n,E));
    p = p0 × p1; //to compute Equ. 4 (see below)
    if (p.prob() >maxprob)
      maxprob = p.prob();
      optpath = p;
  }

  // insert the optimum path from s to E into the table
  table.insert(optpath);
  stack.pop();
}
```

Figure 7: Pseudo code to find optimal (highest probability) path from S to E. *graph* is the graph of fiber segments and *table* refers to the dynamic programming table.

While there are many ways to implement dynamic programming, our implementation uses stack recursion to propagate possible fibers. Pseudo code of the algorithm is given in Figure 7. For a single query, we proceed in three steps. In the first step we check the dynamic table to see if the given query has already been computed. If not then in the second step we check to see if all subpaths required for the computation are available. The terms P(S,S') are taken directly from their respective edges in the graph. The terms P(S',E), if not found in the dynamic table and not a terminal case, must be pushed onto the stack for subsequent calculation. At this point if any terms P(S',E) are not available then we halt the calculation of the current subpath, but keep it on the stack. When the stack once again unwinds to this calculation we will then have all subpaths necessary to continue on to the third step. In the third step the probability of subpaths are combined. Although the classic way to combine costs in dynamic programming is an

addition as in Equation 4, we multiply the two costs together since they are really probabilities. At last, the fiber segment with the highest probability is selected and stored in the dynamic table.

2.5 Constructing Final Nerve Connections

Once an optimal path has been found, connecting the voxel centers on the fiber path produces a piece-wise approximation of the real fiber connection, which is only C_0 continuous. To construct smooth neuronal connections that faithfully reflect the path we have discovered, we again employ Bayes Rule in a more conventional fiber tracking procedure. Our approach is similar to tensorline propagation. In particular, we use a linear combination of two vectors according to:

$$v_{out} = \alpha_{track} v_{Guide} + (1 - \alpha_{track}) v_{Bayes} \quad (5)$$

where v_{Bayes} is the direction of maximum probability resulting from the evaluation of Bayes Rule and v_{Guide} is the direction of the guide path, i.e. the direction pointing towards the next voxel in the path. The parameter α_{track} controls how tightly the fiber follows the guide path, with smaller values allowing the fiber to "wander" more. Fiber tracking starts by choosing a seed near the starting point of the path, and from there the fiber is propagated using v_{out}. The tracking stops when the fiber arrives at a point within the ending voxel of the path.

3. Results

In this section we present the results of the global path algorithm along with images depicting the visualization of possible neural fibers within the WM. All testing was performed using a single machine equipped with a Pentium IV 3.4 GHz with 2.0 GB RAM. Test subjects included two separate tensor data suites, one 'small' (64x64x18, corresponding to the resolution for current clinical use) and one 'large' (256x256x30) research dataset. Each set consists of the tensor volume and an associated fMRI volume marking the ROI. fMRI identifies ROI in the brain based on the correlation between brain activity and a given task performed by the patient. By using the ROI from fMRI as the endpoints of the global path search, we can explore the structural connectivity between regions known to be functionally connected.

As described in Section 3.1, the first step is to create a graph from the tensor dataset by building edges between voxels and computing the PCM tables, while excluding any voxels not within the WM. This construction occurs once as a preprocessing step and the graph is stored for later use. Table 1 gives the size of the resulting graphs and the time needed to construct the graph for each dataset. Note that for the large dataset we create the graph at a lower resolution to facilitate reasonable search times.

	Small Suite	Large Suite
Volume Dimensions	64x64x18	256x256x30
Voxel Dimensions	3x3x5 mm	1x1x3 mm
Graph Dimensions	64x64x18	128x128x30
Graph Size	16 MB	120 MB
Graph Construction Time	5 min.	35 min.

Table 1. Test data and associated 3D probability graphs.

In our implementation the user selects both the start and end ROI (derived from fMRI) and then searches for high probability paths. For a given session the dynamic table is maintained between consecutive searches to further increase the search speed. At some point, however, main memory fills up and the table must be purged. The drawback to this enhancement is that search times, already quite irregular, become even more unpredictable. Once path(s) have been found, fibers are integrated according to Equation 5. Based on our experiments, α_{track} should be set between 0.4 and 0.8. Values below 0.4 almost always allow the fiber to wander off, and values above 0.8 result in fibers that look almost exactly like the guide path. We set α_{track} to 0.6 for all our results.

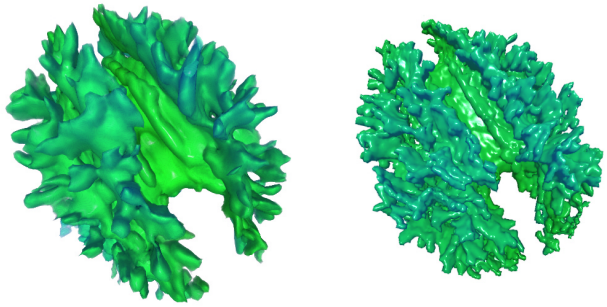


Figure 8. WM surface based on anisotropy for two DT-MRI datasets. The dataset on the left is 64x64x18, whereas the dataset on the right is 256x256x30.

Upon completion of the path search and fiber tracking session, the user is offered a visualization of the results. An important part of this visualization is the WM, which provides important contextual clues. We render the WM as a scalar volume using a hardware volume renderer in order to provide flexible real-time visualization. The scalar volume is built from the tensor volume in two steps. First, region growing is used with an appropriate anisotropy threshold to create a binary volume or mask with '1' indicating WM and '0' otherwise. Then in the second step a scalar volume of anisotropy is calculated only within the WM using the generated mask. Figure 8 shows the WM rendered as an opaque surface for both datasets.

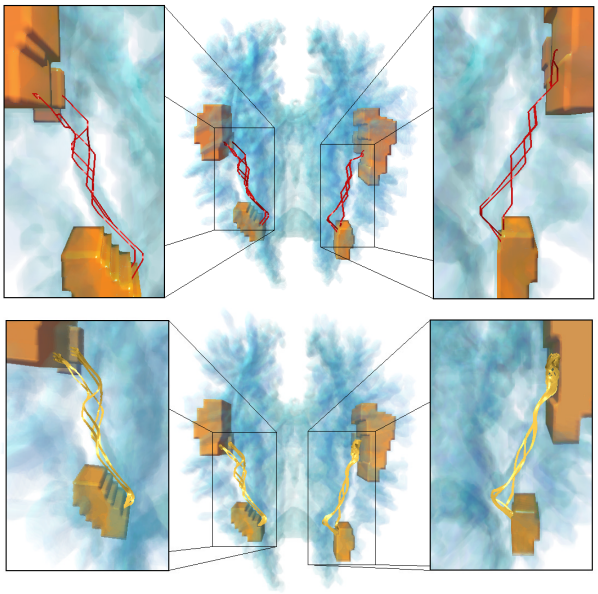


Figure 9. Paths connecting the same side of the brain.

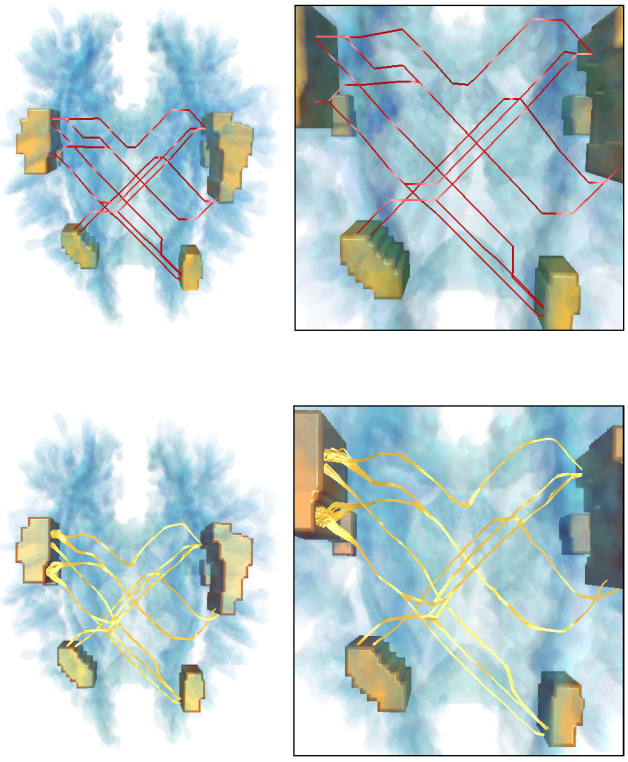


Figure 10. Paths connecting opposite sides of the brain.

Figures 9 and 10. Integrated visualization of neuronal connections in the brain. WM is rendered transparently in blue, and ROI from fMRI are rendered opaquely in orange. The top images (red lines) show the computed optimal paths, and fibers integrated along those paths are shown on the bottom (gold lines).

In Figures 9 and 10, paths resulting from the optimal path algorithm are displayed for the high-resolution dataset, together with fibers integrated along the paths. Due to the confidential nature of this data set, we are not at liberty to identify the ROI. As the ROI consist of more than one voxel, our method uses the centroid of the ROI to define a single start/end point. For large ROIs we manually subdivide the region and use the centroid of each partition.

In some cases there may be two different paths with almost the same probability. In these cases we keep and display all such paths, as each path may be valid. In addition, as the designation of the start and end ROI is arbitrary, we run each search twice, switching the start and end points for the second run. In Figure 9 several paths connecting ROIs on the same side of the brain are shown. These paths are relatively short, consisting of 20-30 edges and requiring anywhere from 4 minutes to 10 minutes to compute. In Figure 10 paths connecting regions on opposite sides are computed. These paths consist of 35-45 edges and require 30-45 minutes to compute. For searches on the same side, typical memory usage is about 500-700 MB for the first search. If the table is maintained between searches and the same area is searched, as many as ten searches can be conducted before flushing is required. In the case of searches across the brain, the table must be flushed after each search. We chose not to store temporary results to disk, although this could further reduce search times. Times for the small dataset (not shown here) are approximately 30 seconds to 2 minutes for searches on the same side and 3 minutes to 12 minutes for searches on the opposite side.

4. Conclusions and Future Work

We have proposed a framework based on global optimization, by means of dynamic programming and cost assignment using Bayes rule. Three issues are tackled here. First, we provide the functionality to explicitly query the optimal connectivity between two end points. This capability is useful to understand the structural connectivity among functionally related cortical regions. Second, our method inherently addresses PVE, which introduces difficulty to reconstruct neural connectivity in areas where nerves cross and branch. In our framework, if the reconstructed optimal connections between two pairs of end points share a common segment of nerve fiber, then a case of nerve branching is discovered, as shown in Figure 9. Nerve crossing can be handled in a similar fashion. Third, the segment-by-segment reconstruction procedure, as well as the dynamic programming method, help to eliminate error accumulation and misguidance by noisy local tensors. However, we recognize that extremely large amounts of signal noise could cause difficulty to even reconstruct a probable nerve segment in a voxel, and thus filtering would still be necessary as a pre-processing.

A key issue regarding any fiber reconstruction algorithm is verification. Although tracers and phantoms have been used previously, this problem is still largely

unsolved. In particular, our method is aimed at discovering new fibers which cannot be discovered any other way, and so verification is largely unattainable at present.

References

1. Mori, S. and van Zijl, P. Fiber Tracking: Principles and Strategies - A Technical Review. NMR Biomed, 2002. **15**(468-480).
2. Delmarcelle, T. and L. Hesselink, Visualizing second-order tensor fields with hyper streamlines. *IEEE Computer Graphics and Applications*, 1993: p. 25-33.
3. Weinstein, D., G. Kindlmann, and E. Lundberg. Tensorlines: Advection-Diffusion based Propagation through Diffusion Tensor Fields. In *Proc. of IEEE Visualization Conference*. 1999. San Francisco, CA.
4. Basser, P.J., Pajevic, S., Pierpaoli, C., Duda, J., Aldroubi, A. In vivo fiber tractography using DT-MRI Data. *Magn Reson Med*, 2000. **44**(625-632).
5. Poupon, C., Clark, C.A., Frouin, V., Regis, J., Bloch, L., Le Bihan, D., Mangin, J.F. Regularization of diffusion-based direction maps for the tracking of brain white matter fascicles. *NeuroImage*, 2000. **12**: p. 184-195.
6. Zhukov, L. and A. Barr. Oriented Tensor Reconstruction: Tracing Neural Pathways from Diffusion Tensor MRI. In *Proceedings IEEE Visualization*. 2002. Boston, MA.
7. Parker, G.J. Tracing fiber tracts using fast marching. In *Proceedings, International Society of Magnetic Resonance*. 2000. Denver, CO.
8. Tuch, D.S., Wiegell, M.R., Reese, T.G., Belliveau, J.W., Wedeen, V. Measuring cortico-corital connectivity matrices with diffusion spectrum imaging. In *Proceedings of International Society of Magnetic Resonance*. 2001. Glasgow, UK.
9. Bjornemo, M., Brun, A., Kikinis, R., Westin, C.F. Regularized Stochastic White Matter Tractography Using Diffusion Tensor MRI. 2002.