Structure Tensor Analysis Update

Scott Trinkle

April 9, 2018

NOTE: The results in this document are inaccurate. The xray data is given as unsigned 8-bit integers. When an 8-bit unsigned integer is assigned a value greater than 255, NumPy wraps the overflow around 0; i.e., if "a" is an unsigned 8-bit int, and you assign a = 260, a will *actually* be assigned the value 4. When calculating the structure tensor, the partial derivatives are squared, causing this overflow in many, but not all voxels. Updated figures are available in the folder 'notes/2018-04-24-redo-report-figs-with-dtype-fix'.

The concept of structure tensor analysis seems to have been initially developed in a 1987 computer vision paper by Bigun and Granlund [1]. It has since been used for a variety of computer vision and materials science applications, including finding the orientation of textile fibers with synchrotron microCT imaging [2]. It was notably applied to validating diffusion MRI with 2D histology by Budde [3, 4] and others. Schilling [5, 6] and Khan [7] have extended the analysis to 3D for the same purpose. In this report, I am following the derivation and notation given in Appendix A of the most recent Schilling paper [6].

1 Derivation

Consider a three-dimensional intensity image, $f(x, y, z)$, with gradient:

$$\nabla f_{\sigma_d} = (f_x, f_y, f_z)^T \tag{1}$$

which is calculated using Gaussian derivative filters:

$$f_x = g_{x,\sigma_d} * f \qquad f_y = g_{y,\sigma_d} * f \qquad f_z = g_{z,\sigma_d} * f \tag{2}$$

where $*$ denotes the convolution operator and g_{i,σ_d} denotes the spatial derivative in the i-direction of a 3D Gaussian, g_{σ_d}, with standard deviation σ_d:

$$g_{\sigma_d}(x, y, z) = \frac{1}{\left(2\pi\sigma_d^2\right)^{3/2}} \exp\left\{-\frac{x^2 + y^2 + z^2}{2\sigma_d^2}\right\} \tag{3}$$

The image gradient is used to construct a "gradient square tensor" by taking the dyadic product of the gradient vector with itself:

$$\mathrm{GST}(x, y, z)_{\sigma_d} = \nabla f_{\sigma_d} \nabla f_{\sigma_d}^T = \begin{pmatrix} f_x^2 & f_x f_y & f_x f_z \\ f_x f_y & f_y^2 & f_y f_z \\ f_x f_z & f_y f_z & f_z^2 \end{pmatrix} \tag{4}$$

Each element of the GST is then averaged over a local neighborhood to create the voxel-wise structure tensor. Most groups have performed averaging using a 3D Gaussian filter with standard deviation σ_N:

$$\mathrm{ST}_{\sigma_N}\left(\nabla f_{\sigma_d}\right) = g_{\sigma_N} * \left(\nabla f_{\sigma_d} \nabla f_{\sigma_d}^T\right) = g_{\sigma_N} * \begin{pmatrix} f_x^2 & f_x f_y & f_x f_z \\ f_x f_y & f_y^2 & f_y f_z \\ f_x f_z & f_y f_z & f_z^2 \end{pmatrix} \tag{5}$$

This results in a 3×3 symmetric, positive semi-definite, rank-two tensor at every voxel. The direction of minimal intensity variation is given by the eigenvector of this tensor corresponding to the smallest eigenvalue. We make the assumption that this direction also describes the corresponding fiber orientation of that voxel.

The eigenvalues can be used to estimate the certainty in the estimated fiber orientation, i.e., the voxel-wise "anisotropy index," which I will refer to as AI. Several specific metrics have been proposed:

The textile group [2] uses the ratio of minimum and maximum eigenvalues:

$$\mathrm{AI} = \begin{cases} 1 - \frac{\lambda_3}{\lambda_1} & \lambda_1 > 0 \\ 0 & \lambda_1 = 0 \end{cases} \tag{6}$$

where λ_1, λ_2 and λ_3 are the primary, secondary and tertiary eigenvalues of the structure tensor.

Schilling [5, 6] uses the "Westin-measure":

$$\mathrm{AI} = \frac{\lambda_2 - \lambda_3}{\lambda_1} \tag{7}$$

Khan [7] uses "fractional anisotropy," which is also used in DTI:

$$\mathrm{AI} = \sqrt{\frac{1}{2} \frac{(\lambda_1 - \lambda_2)^2 + (\lambda_2 - \lambda_3)^2 + (\lambda_1 - \lambda_3)^2}{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}} \tag{8}$$

The AI is then commonly used to threshold the image f, so voxels with low anisotropy are not included in the orientation distribution.

2 Implementation

2.1 Gaussian filtering

Currently, the Gaussian filtering is being implemented with the ndimage.gaussian_filter function in the SciPy Python package. This function is implemented in C and convolves the nD input in the spatial domain with a series of 1D Gaussian filters with specified standard deviation and order along each axis (an order of 0 corresponds to a Gaussian, an order of 1 corresponds to the first partial derivative of a Gaussian).

Schilling [6] uses values of $\sigma_d = 1$ µm and $\sigma_N = 2.5$ µm for histological squirrel monkey brain data. Khan [7] uses $\sigma_d = 2$ µm and performs non-Gaussian neighborhood averaging within a 3D ROI for histological rhesus macaque brain data.

Both papers provide relatively thorough descriptions of their sensitivity analysis. Schilling based his choice of σ_d and σ_N on comparisons with manually segmented fibers, which I think would be good for us to implement as well. In general, both groups found that the results were reliable within a relatively narrow range of σ_d

(partial derivative) corresponding to the diameters of the myelinated fibers. The results were less sensitive to choice of averaging window, σ_N. Khan recommends using this value to match the resolution of the MRI data.

2.2 Eigenvalues/vectors

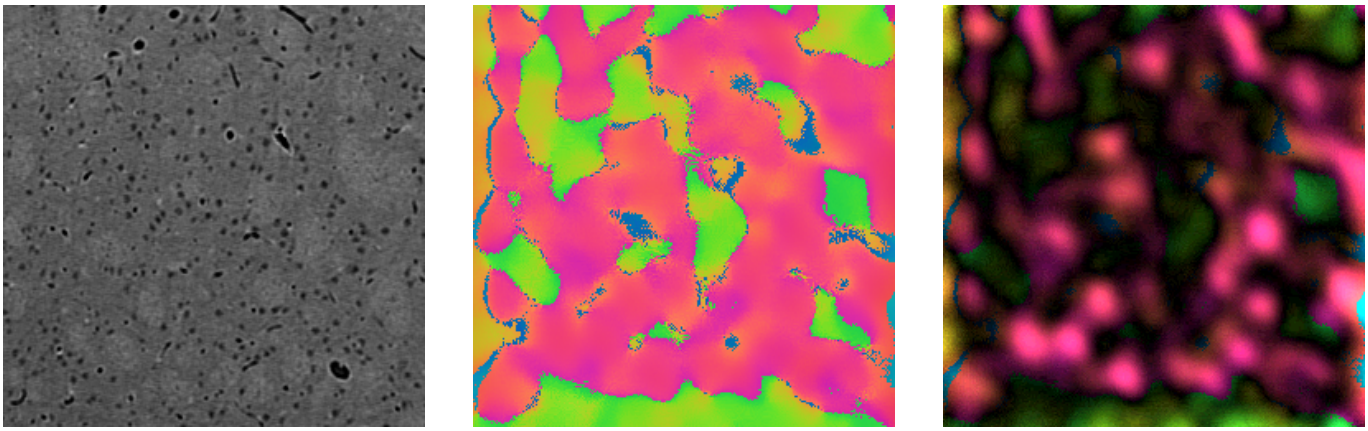
The eigenvalues and eigenvectors are currently calculated with the linalg.eigh function in the NumPy Python package. This is also implemented in C and is optimized for Hermitian/symmetric matrices.

3 Visualization

3.1 Raw results

To visualize the results of the structure analysis, I am currently mapping the three components of the eigenvector at each voxel from [-1, 1] to [0, 255], scaling the vector by the AI (I am currently using the Westin metric, following [6]), converting to an unsigned integer and saving as an RGB tiff stack. The conventional mapping in the tractography literature is $(x, y, z) \rightarrow (R, G, B)$, which I have kept here, defining
$(column, row, slice) \rightarrow (x, y, z)$ in NumPy/ImageJ image axes coordinates.

An example is shown in Figure 1 for $\sigma_d = 2$ µm and $\sigma_n = 7$ µm (for now, these values were chosen to provide the best visual alignment with the myelinated axons in the raw data - there will be more rigorous sensitivity studies soon)



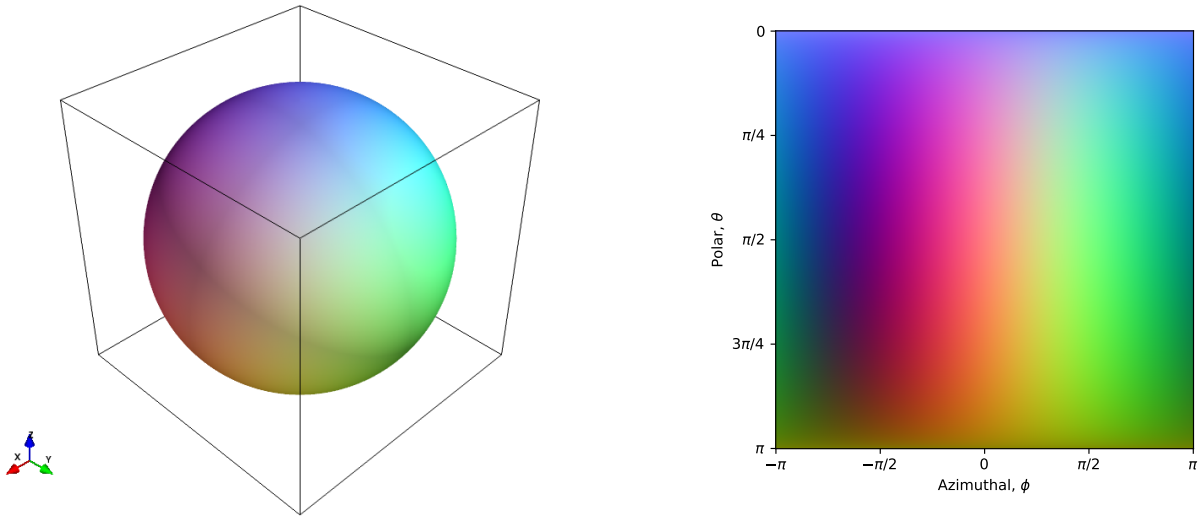
(a) Raw data (b) Orientation estimate (c) Intensity scaled by AI

Figure 1: (a) Slice of a 256 × 256 × 256 ROI in the raw xray data. The nominal voxel size is 1.2 × 1.2 × 1.2 µm^3. (b) Orientation results. RGB color corresponds to column-row-slice (x, y, z) eigenvector components, respectively. (c) Orientation results scaled by AI.

NOTE: These results are different than those shown in group meeting last week. I found an error in how I had implemented the partial derivative filters, as well as how I was doing the AI scaling. The promising result from these figures is that the structure tensor analysis seems to be successfully identifying the axon bundles. Regions of high anisotropy (high intensity in Figure 1c) correspond to the axon bundles in Figure 1a, and the orientations (color) within these regions are relatively homogeneous.

A more rigorous sensitivity study is needed to validate the orientation directions themselves. The fibers in Figure 1c are predominately red, indicating strong orientation in the x direction. To verify, I made an orientation colormap in spherical coordinates, shown in Figure 2. This colormap was created by mapping a 2D

grid of polar and azimuthal angles to 3D Cartesian coordinates, then scaling the (x, y, z) results to 0–255 RGB values. I manually chose a few points within axon regions of Figure 1b and found their spherical position using the actual RGB values in Figure 2b. This seemed to indicate that the current results are reporting the axons



(a) Spherical (b) 2D

Figure 2: Orientation colormap.

traveling in the xy–plane ($\theta \approx \pi/2$), when (to my eye) they are actually traveling more along the slice direction in the raw data, meaning that there might be inconsistencies somewhere in how I have defined my coordinate axes.

I have also experimented with visualizing the results as a 3D vector field. The pipeline is in progress, but I'm still working on the best way to threshold the results; visualizing every vector in the volume looks extremely cluttered.

3.2 Orientation Distribution Function

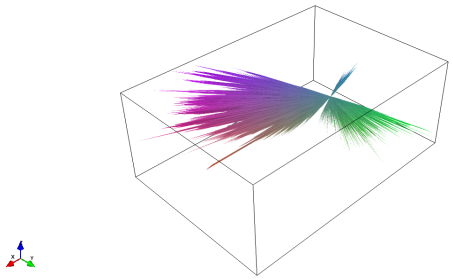
Ultimately, we are interested in the distribution of orientations within an ROI. As with the colormap above, we accomplish this by first reparameterizing the orientations into spherical coordinates, with polar angle $\theta \in [0, \pi]$ and azimuthal angle $\phi \in [-\pi, \pi]$:

$$r = \sqrt{x^2 + y^2 + z^2} \tag{9}$$

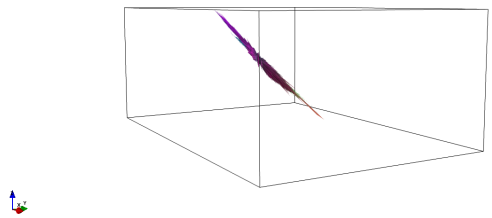
$$\theta = \cos^{-1}\left(\frac{z}{r}\right) \tag{10}$$

$$\phi = \tan^{-1}\left(\frac{y}{x}\right) \tag{11}$$

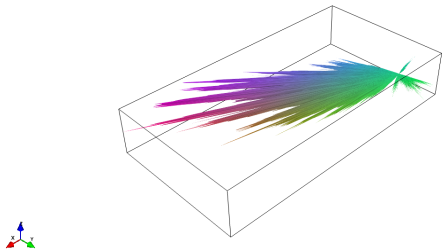
A 2D histogram of (θ, ϕ) orientations is generated with the histogram2D function in the NumPy package, with bin locations determined by spherical Fibonacci sampling, which produces approximately evenly spaced points on the sphere. The histogram results are then visualized on a sphere with the mayavi Python package, with the radius of the sphere depending on the number of counts at that angular location. Initial results are shown in Figure 3. Figures 3a and 3b depict the raw ODF for the entire sample volume, while Figures 3c and 3d depict the ODF after thresholding by AI > mean(AI). As expected, the AI thresholding generates a more narrow,



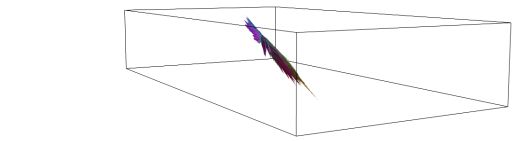
(a) Raw ODF, view 1



(b) Raw ODF, view 2



(c) AI-thresholded ODF, view 1



(d) AI-thresholded ODF, view 2

Figure 3: (a-b) Two views of the raw ODF, sampled with n=800 points on the sphere. (c-d) Two views of the same ODF, thresholded to only include voxels with AI > mean(AI). Note that the orientations fall narrowly into a single plane.

directed ODF. Also, as with the results in Figure 1, most orientations appear to fall more in line with the positive x-axis than the z-axis, indicating an incorrect coordinate frame definition at some point in the pipeline. It is also worth noting that most orientations seem to be falling within a single plane, identified in Figures 3b and 3d. Again, a rigorous sensitivity study will allow us to validate the absolute orientations, as well as to optimize the choice of AI metric and threshold value.

The next step is to expand this orientation distribution function on the spherical harmonics for use in various comparison metrics with the MRI distribution functions. We will also need to address how sensitive those results are to n, the number of sampling points for the histogram.

4 Current Issues / TODO list

- Sensitivity study
  - Segment myelinated axons from ROI and/or create fiber phantom
  - Use segmented data / phantoms to validate coordinate frame
  - Compare orientation/AI results from raw data to segmented fibers / phantom to optimize the choices of σ_d, σ_n, AI metric and AI threshold value
- Implement spherical harmonic expansion

  - How sensitive is the expansion to the number of sampling bins?

- Gordon Conference
  - Finish abstract
  - Identify a few sample ROI in x-ray and MRI data for comparison.
  - Potentially look at ROI registration for quantitative comparison

- Scalability
  - With the current test parameters, it takes about 30 seconds on my Macbook to calculate the structure tensor results for this $256 \times 256 \times 100$, 6.6 Mb ROI
  - I will do more detailed testing to find the bottlenecks. As we have discussed, I suspect I will need to look at implementing the convolutions in the Fourier domain.

References

[1] J. Bigun and G. H. Granlund, "Optimal orientation detection of linear symmetry," pp. 433–438, IEEE, 1987.

[2] I. Straumit, S. V. Lomov, and M. Wevers, "Quantification of the internal structure and automatic generation of voxel models of textile composites from X-ray computed tomography data," *Composites Part A: Applied Science and Manufacturing*, vol. 69, pp. 150–158, 2015.

[3] M. D. Budde and J. A. Frank, "Examining brain microstructure using structure tensor analysis of histological sections," *NeuroImage*, vol. 63, no. 1, pp. 1–10, 2012.

[4] M. D. Budde and J. Annese, "Quantification of anisotropy and fiber orientation in human brain histological sections," *Frontiers in Integrative Neuroscience*, vol. 7, no. February, pp. 1–8, 2013.

[5] K. Schilling, V. Janve, Y. Gao, I. Stepniewska, B. A. Landman, and A. W. Anderson, "Comparison of 3D orientation distribution functions measured with confocal microscopy and diffusion MRI," *NeuroImage*, vol. 129, pp. 185–197, 2016.

[6] K. G. Schilling, V. Janve, Y. Gao, I. Stepniewska, B. A. Landman, and A. W. Anderson, "Histological validation of diffusion MRI fiber orientation distributions and dispersion," *NeuroImage*, vol. 165, pp. 200–221, jan 2018.

[7] A. R. Khan, A. Cornea, L. A. Leigland, S. G. Kohama, S. N. Jespersen, and C. D. Kroenke, "3D structure tensor analysis of light microscopy data for validating diffusion MRI," *NeuroImage*, vol. 111, pp. 192–203, 2015.