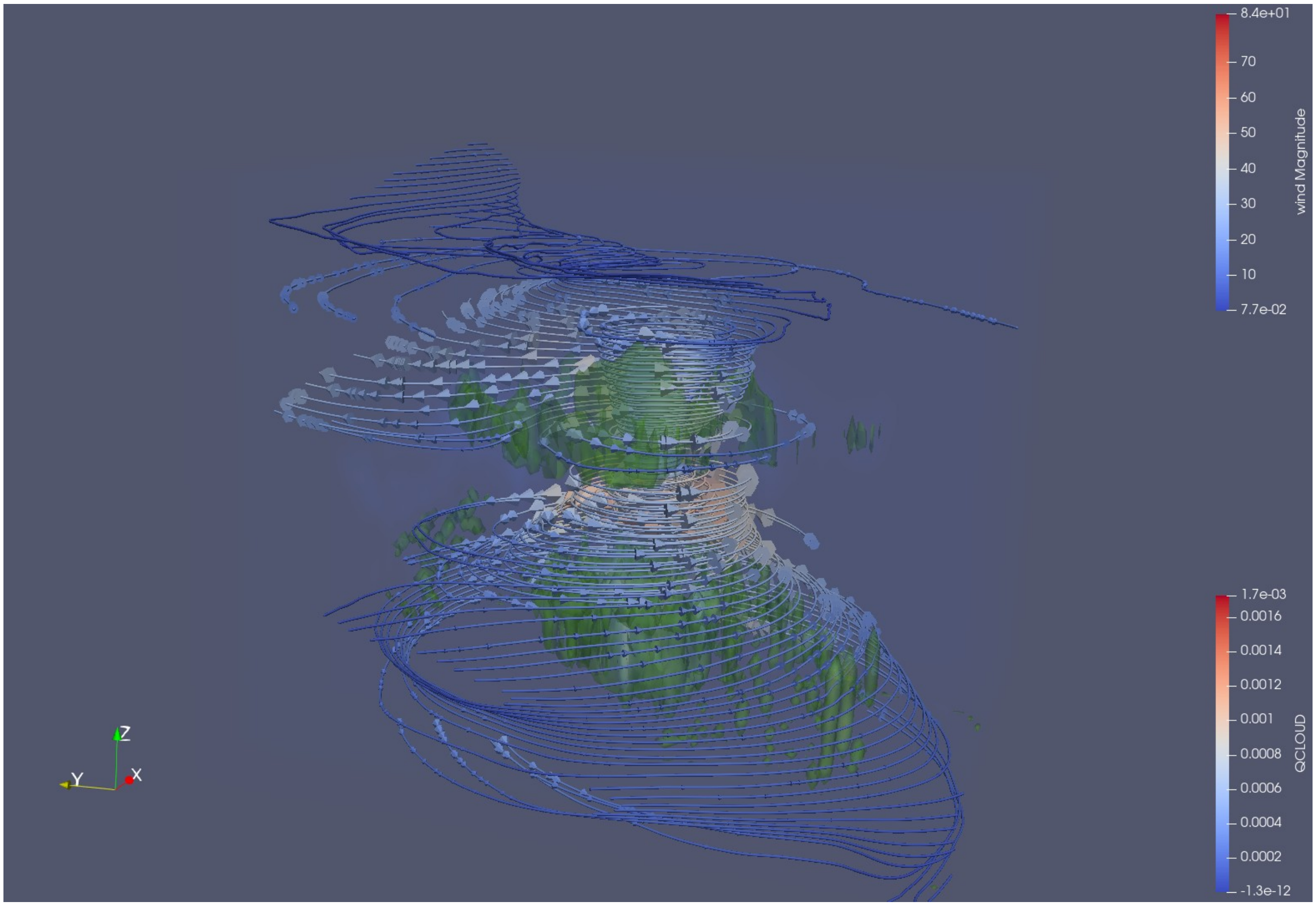
Jack Wilburn
Assignment 5

Part 1)
QCLOUD magnitude and wind vector visualization of hurricane Katrina



Here I created a visualization with all of the requested elements. I used volume rendering for the QCLOUD dataset after importing the data. I then applied the contour filter to create the isosurface and turned on ray tracing to allow for the opacity change that was requested. I reduced the opacity down to 0.4 so that it was semi-transparent. I then added the streamlines using a line through the center of the data and a resolution of 100 lines. I added glyphs to the lines to show the wind direction and magnitude. After all this, I modified the color mapping so that the wind magnitude was mapped onto the streamlines and the glyphs, the QCLOUD was mapped to the volume rendering, and I chose a green for the isosurface color (just so it wouldn't interfere too much with the other layers).

From this visualization we can see that the hurricane rotated counter clockwise (using the streamline and glyph data), that the wind speed was largest at the eye wall and where the wind was being pulled away from the eye wall, and that the QCLOUD variable seems to have the largest magnitude slightly above and below the center of the eye.

Part 2)

1) What is diffusion tensor MRI imaging? State three types of diffusivities and describe each briefly. How a diffusion tensor can be represented mathematically?

Diffusion tensor MRI imaging is a visualization technique used to determine the anisotropic diffusion properties of a substance (usually water) within a certain region of a tissue. Inside of neuron cells, for example, the water flows most easily along the axons and thus we can tell the orientation of the cell by looking at which direction the water flows most easily.

The three types of diffusivities are: linear anisotropic, planar anisotropic, and isotropic. Isotropic diffusivity means that the substance diffuses equally in all directions. In linear anisotropy, the diffusion is fastest in a direction along a line and slower in other directions. In planar anisotropy, the diffusion is fastest across a plane and is slower in other directions.

Diffusion tensors are represented as a matrix, which allows for many linear algebra style manipulations, such as deriving eigenvalues and eigenvectors, doing decompositions, etc.

2) Briefly describe box, ellipsoid, and superquadric glyphs for visualization of tensors field. Compare and contrast the benefits and disadvantages for these glyphs.

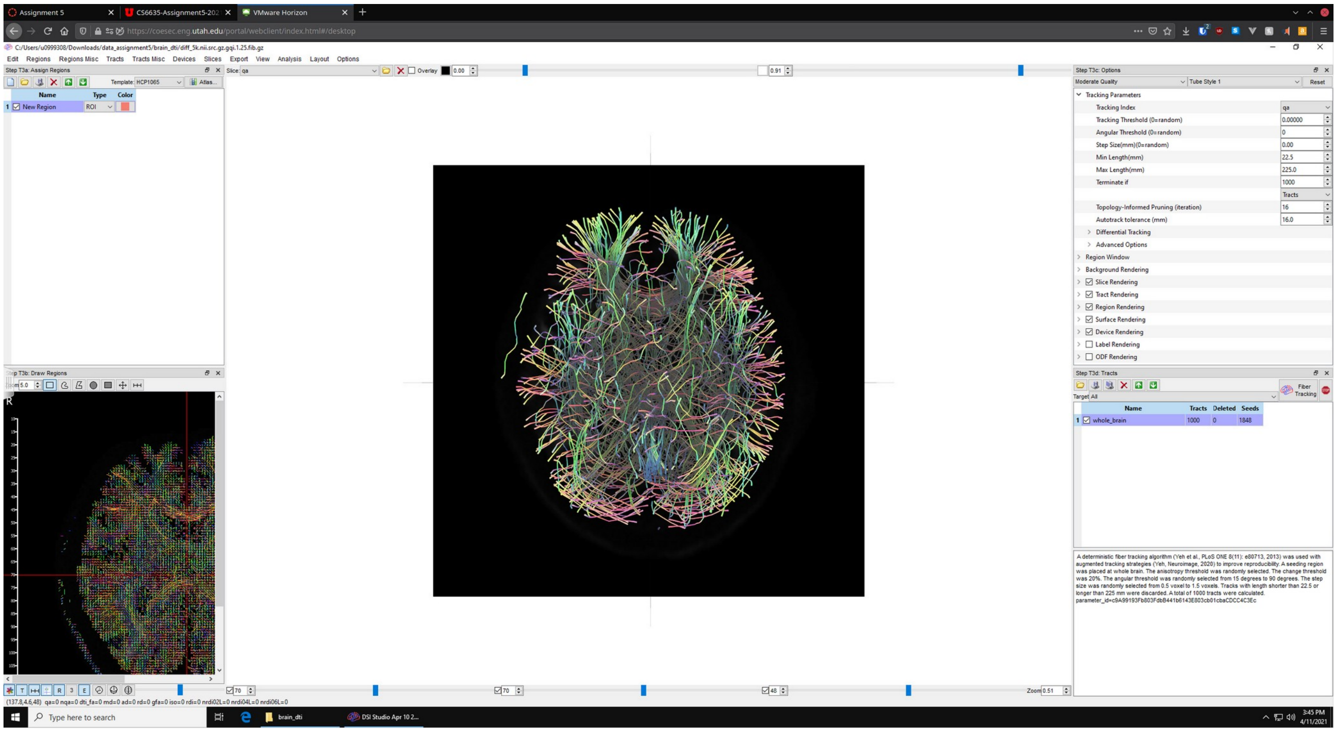
Box glyphs represent the anisotropy of a tensor field using cuboid shapes. The lengths of the side of the boxes are mapped to the magnitude of the diffusion in 3 directions. They work well to show linear anisotropy, but can be misleading for planar anisotropy and isotropy.

Ellipsoid glyphs show the mean squared distance that water will diffuse in all directions over some time step. This means that the shape of the ellipsoid shows us directly, which directions have the fastest diffusion. In isotropy, you'd have spheres and in anisotropy, you'd have an elongated spherical object. These are both ellipsoids, with the spheres just being a special case.

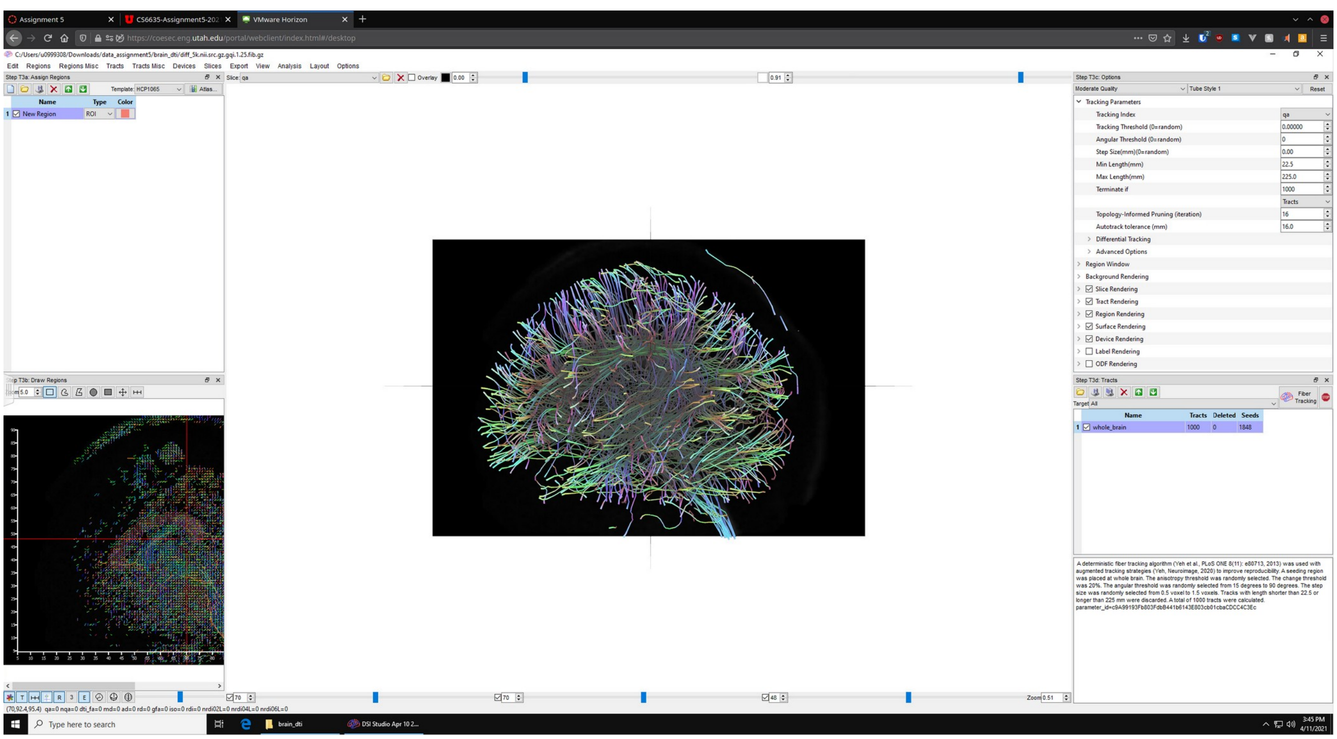
Superquadric glyphs use a combination of cylinders, rounded boxes, and ellipsoids to represent the diffusion tensors. Linear anisotropy and planar anisotropy take the form of cylinders (with larger radius/smaller heights for planar isotropy), and isotropy takes the form of a sphere. Because the glyphs are rotationally symmetric, this glyph set avoids misleading representations.

Part 3)

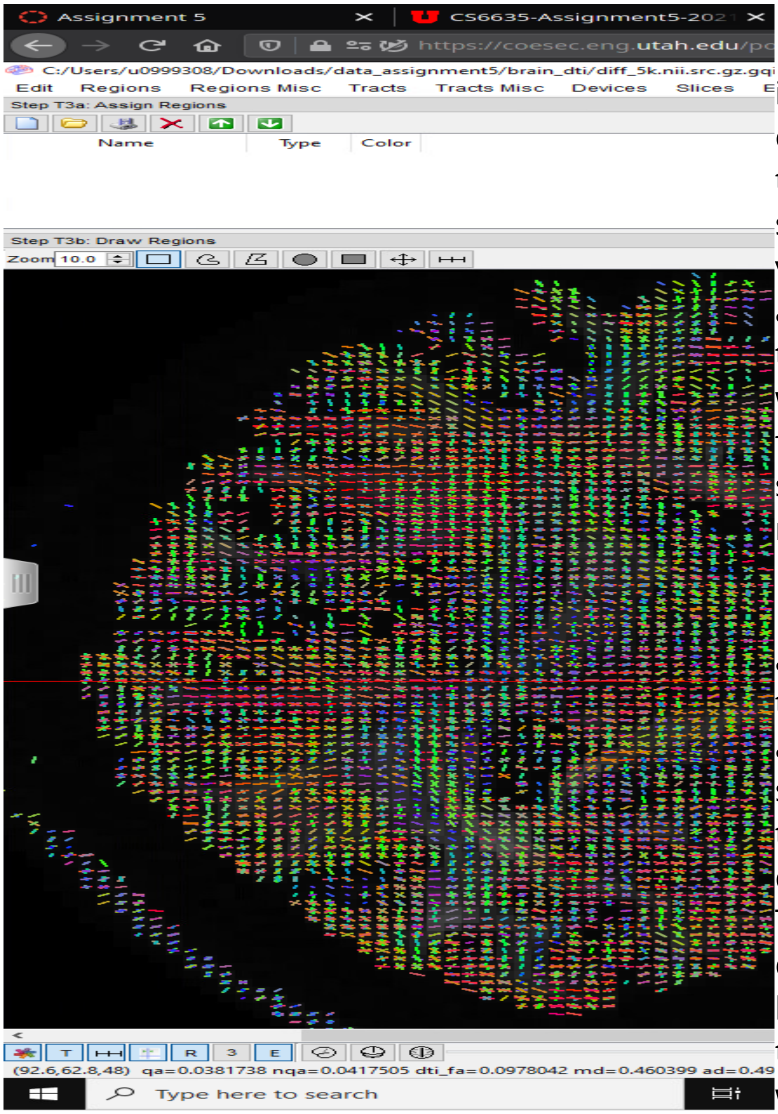
1000 tracts across the human brain using DSI studio. Transverse view:



1000 tracts across the human brain using DSI studio. Sagittal view:



Transverse view of the diffusion tensor field of the human brain using DSI studio:

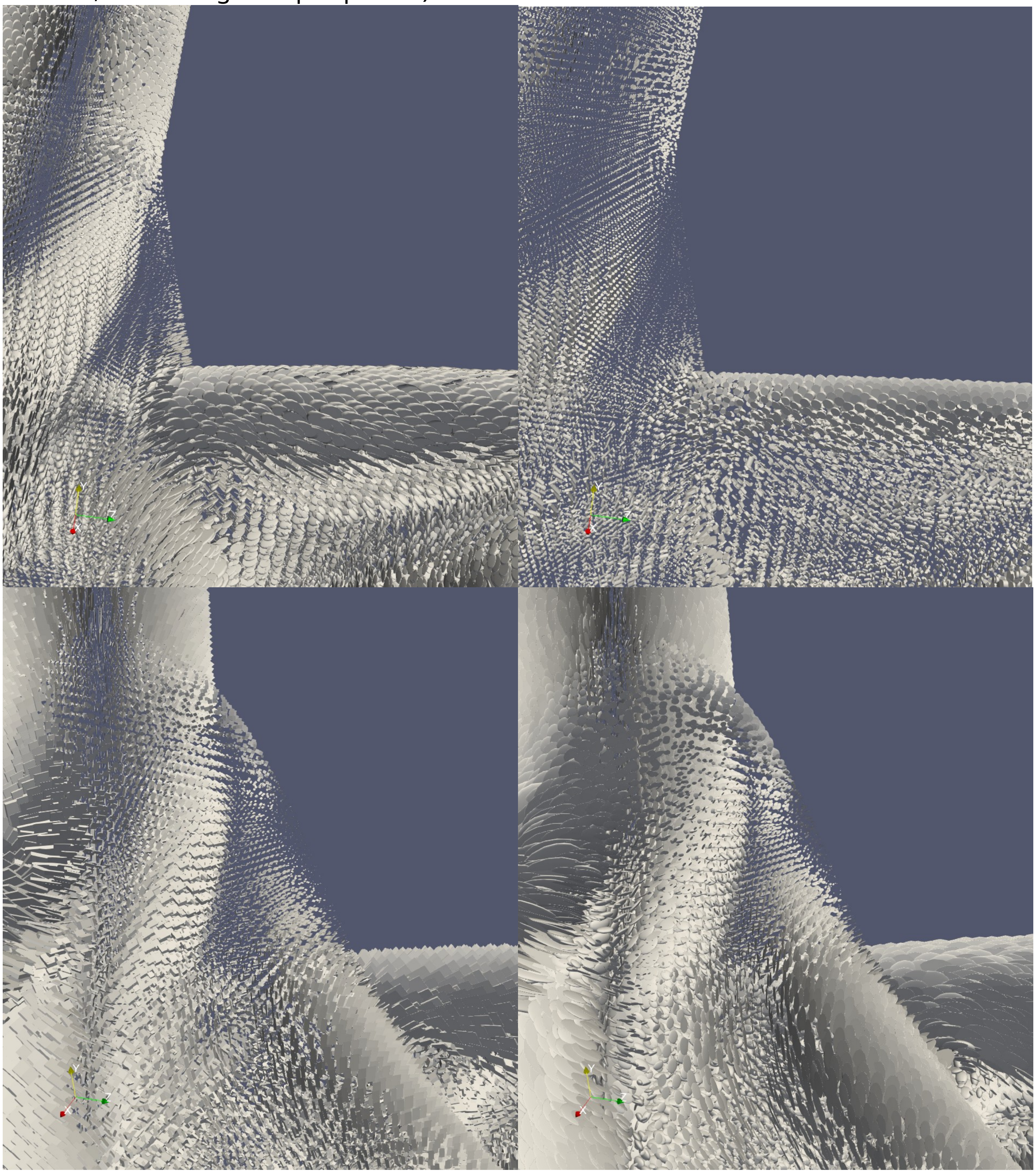


For the first visualizations, I imported the data on the windows cade machine and viewed the full track of the brain. I limited my sample to just 1000 track, so that the visualization was not overwhelming and that you could see the individual tracks. I had attempted to show the white matter tracks through the front of the brain using various tract selections, but I couldn't find a good representation.

For the last vis, I imported the data and increased the size of the panel to the left to see the tensor field. It appears as though the green glyphs show areas of larger diffusivity and the glyph are oriented in the direction of most diffusivity. Therefore, we can see lots of diffusivity through the center of the brain, through the white matter tracks, with regions on the outside with relatively less diffusivity.

Part 4)

4 views of the tensor field using paraview (top left: sphere, top right: cylinder, bottom left: box, bottom right: superquadric)



Parameters:

Top Left: Sphere, radius = 0.3, theta resolution = 8, phi resolution = 8, scale factor = 0.001

Top Right: Cylinder, radius = 0.1, height = 1, scale factor = 0.001

Bottom Left: Box, x,y,z = 0.5, center = (0,0,0)
Bottom Right: Superquadric, center = (0,0,0), theta roundness = 1, phi roundness = 1

Which glyphs did you find the most informative for gaining insight into the tensor field data? Why?

I found the ellipsoid glyphs most easy to interpret and understand. I found that it was the most intuitive to understand, probably due to it having the best scaling. The direction of the anisotropy is clear and we can see all the planar anisotropy on the arm area near where it connects to the body. I think the shape is most intuitive to me out of the options.

The second best is superquadric, but they seem to have much more intense memory usage so I wouldn't choose them as a first choice.

Part 5)
I read “Gaussian Transfer Functions for Multi-Field Volume Visualization” by Kniss et al.

- What is the new innovation described in the paper?

  The new innovation in this paper is called “Gaussian Transfer Functions” that relate to computing volume rendering attributes such as color and opacity. The claim is that when using traditional transfer functions, multivariate data makes the lookup tables so large that they can’t be well stored in memory. This leads to slow visualization performance as data is swapped and generally takes a long time to move around. Using a Gaussian Transfer Function is preferable because the number of parameters that must be stored is much smaller and the computation to map the input to it’s output is not computationally expensive.

- What did you learn by reading the paper?

  This paper taught me to think more deeply about how the transfer functions that we apply to volume rendering take up memory space and how a lookup function is sometimes not the best choice. I tend to have the understanding that using a hash table and doing look ups is about the fastest thing that you can do (O(1) time), but I’d never really considered that the data you might want to store in it could be bigger than the memory of the system. I just hadn’t worked with examples that big. This insight can likely be applied to other parts of my computing knowledge.

- What are the weaknesses in the paper? E.g. What claims are not convincing, and why? Are there claims made without adequate evidence? Are there weaknesses in how the authors evaluate the performance/effectiveness of their proposed technique?

  The authors claim that decomposing higher dimensional transfer functions (e.g. a 4d transfer function) into smaller transfer functions that can be multiplied together (e.g. 4 * 1d transfer functions) “dramatically limit[s] the kinds of features that can be visualized”, but I’m not sure exactly why that would be. The paper would certainly be a bit more friendly to visualization novice if they explained why.

It's also not immediately clear to me that using a computation over a lookup table would be faster to render. If the table fits into RAM, then a lookup is surely faster. And if it doesn't fit into RAM, then is the computation actually faster than swapping out data. It seems to be implicitly assumed that's the case. How has this changed over the last 18 years

- Is what the paper presents useful, or is it just a curiosity?

  This information in this paper is truly useful. Assuming that this computation is faster than swapping out data, this computation will allow the visualization of large dataset with more features. If the visualization system supports that well, that would lead to more insight and understanding of the interconnected variables.

- What did you want to know more details about?

  After reading this paper, I'm considering some of the other potential ways that we might speed up this multivariate volume rendering, how the increased density of RAM might obviate the need for this kind of computation, and with faster disk speeds than we had in 2003 if this papers assumptions about RAM and swap hold up.

- Does the paper provide enough detail to allow you to implement the proposed technique yourself? If not, which part is not detailed enough?

  This paper definitely gives enough detail for a user to implement it. There are equations for each step and a discussion of edge cases that the computations have to support.

Discussion

Please find my analysis and comparisons inline above.

I found that the instructions for this project were pretty straight forward, but I still ran into a couple of issues. That one bug with paraview rendering when applying many filters I managed to solve just by adding ray tracing. I also got into a bit of a loop with the DSI studio importer where it wasn't clear to me that I needed to add the values and vectors to the data we imported. Thus I kept importing and seeing nothing happen and I couldn't visualize the data. After a while, I saw that it was necessary, but I spent about 20 minutes doing the same thing and getting no results.

Conclusion

I learned a lot in this assignment. To list them: how to render multiple different attributes with opacity and volume rendering in paraview, how to use DSI studio, how to interpret tensor fields and the glyphs in the visualizations of the fields, and how Gaussian Transfer Functions work.

References

Kniss et al. Gaussian Transfer Functions for Multi-Field Volume Visualization. IEEE Visualization. October 19-24, 2003

ZHANG, S, LAIDLAW, D, KINDLMANN, G. Diffusion Tensor MRI Visualization. The Visualization Handbook. 2005.

ZHUKOV, L , BARR, A. Oriented Tensor Reconstruction. The Visualization Handbook. 2005.