Visualizing brain fibres with VTK.js and D3.js

Loraine Franke
loraine.franke001@umb.edu
University of Massachusetts Boston

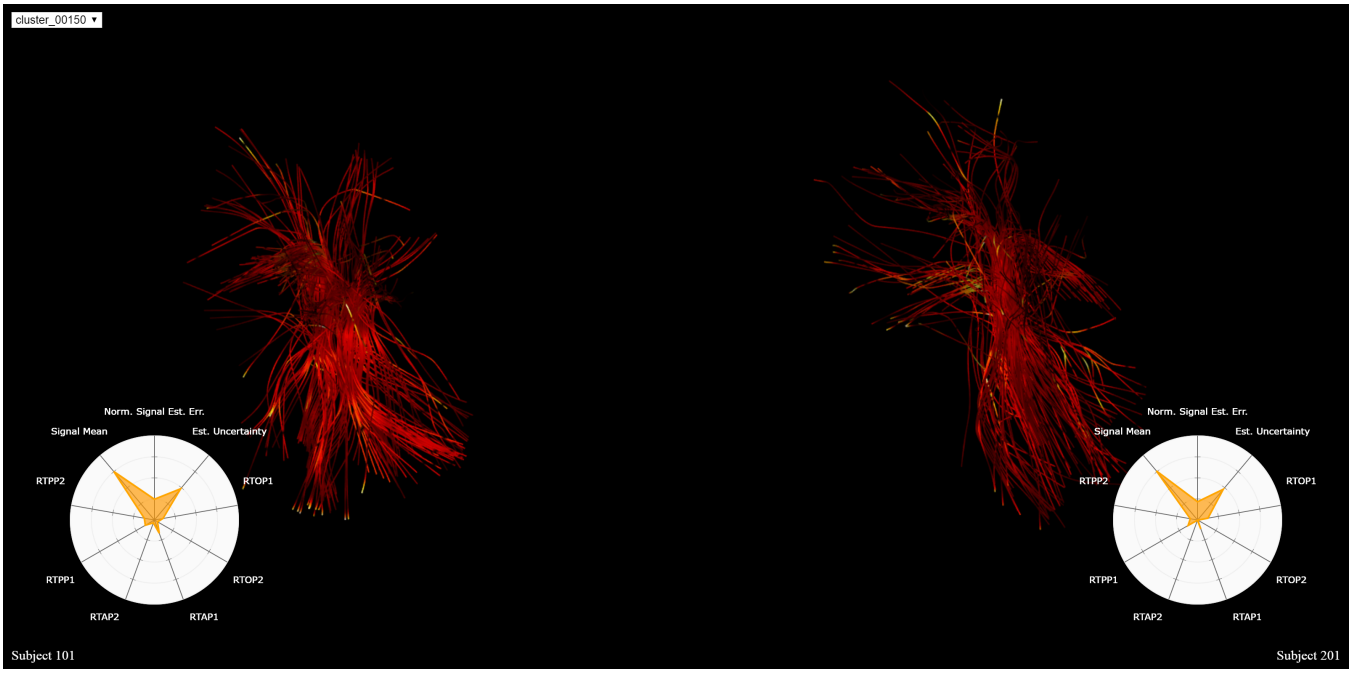

Figure 1: Overview of the visualization tool with VTK.js and D3.js showing 3D fibre clusters of two brain subjects with two-dimensional plots for each cluster showing the mean of the data from every property: https://lorifranke.github.io/ABCD/vtkjs/vtkjs.html

ABSTRACT

In this paper, we present *Fibrestars*, a novel visualization tool for detailed analysis of brain fibre clusters from DTI data. The original 3D fibre structure of a cluster is abstracted into a 2D radial plot by calculating the means of the scalar data for all fibres and plotting them onto the axes. The tool works fully in the browser and is written in JavaScript using the libraries Vtk.js and D3.js. This visualization technique facilitates the understanding, exploration and analysis of data from fibre clusters.

KEYWORDS

VTK.js, D3.js, Visualization, brain fibres, cluster, neuroimaging, fibre tracking, DTI, connectomics

CS460, Fall 2019, Boston, MA

ACM ISBN 1337.
https://CS460.org

ACM Reference Format:
Loraine Franke. 2019. Visualizing brain fibres with VTK.js and D3.js. In *CS460: Computer Graphics at UMass Boston, Fall 2019*. Boston, MA, USA, 3 pages. https://CS460.org

1 INTRODUCTION

In recent years, studying the brain and its neural connectivity has become an emerging field in neuroscience. Especially research concerning connectomics has gained in popularity. These are used to construct maps and diagrams of the brain's connectivity with high resolution images to get better insights into functioning of the brain [4]. Moreover, they can help to track and detect disease patterns [3]. For example, when comparing the brain connectivity between a healthy subject and a subject suffering from disease, it is helpful to get deeper insights into the data to derive potential causes. The main goal of this paper is to visualize the available fibre cluster data in a two-dimensional diagram combined with previous existing three-dimensional approaches. This paper should contribute to help neuroscientists, neurosurgeons and other professionals in this field to analyze and explore brain structures. The data can be shown in different levels of abstraction. For this task, we chose two

dimensional *radar charts* or *radial plots* to complement the three dimensional visualization of fibre clusters.

2 RELATED WORK

In recent years, the field of neuronal connectivity visualization is emerging. Related research such as Al-Awami et al. have shown how to visualize 3D brain tissue data onto a 2D subway map to allow scientists in this field analyzing neurons and their connectivity at nanoscale level. Further work from Mohammed et al. presents a tool to explore connectomics datasets on different abstraction levels. Other abstract visualization approaches have been used in illustrating the cerebral artery network [6]. This paper will present a novel abstract visualization technique and will be based on the following two JavaScript libraries: At first, VTK.js which is derived from the VTK software and is used for 3D scientific visualizations in the browser. It provides 3D rendering using WebGL for both geometry and volume rendering [7]. For 2D visualizations the JavaScript library D3.js offers a variety of methods to manipulate Document Object Models (DOM) based on data with powerful visualization components [2].

3 METHOD

The fibre data was collected as diffusion tensor images (DTI) from two subjects located in their left hemisphere of the brain. The data was available in a trk- or vtp-file format for further processing. For the purpose of this visualization we use the JavaScipt libraries D3.js and VTK.js. These can be implemented with code snippets in the header of the html-file.

3.1 Implementation

At first, the library VTK.js is used to access the data from the vtp-file. Each cluster consists of *polydata*. PolyData can be described as a surface mesh structure that holds data arrays in points, cells or in the dataset itself. The point data contains a number of data arrays with floats and scalars or properties to each point of the fibre. With VTK.js we can read the cluster's polydata with

```
polydata = reader.getOutputData()
```

and then extract the values from the lines with

```
polydata.getLines().getData()
```

and values from the points with

```
polydata.getPoints().getData()
```

The means are calculated per fibre and then per full cluster. If a fibre contains *NaN* values, we skip these and continue with adding up the values to finally divide them by their quantity to obtain the mean.

Next, the data needs to be implemented in the radial plots. This is done by creating the plots within an SVG using D3.js. For example, creating a single circle for the outside of the radial plots, we can use the following code:

```
var circle = svg.append('circle');
circle.attr('cx', CCX)
        .attr('cy', CCY)
        .attr('r', CRAD)
        .attr('stroke', '#efefef')
        .attr('fill', '#fafafa');
```

In the code above, *r* describes the radius of the circle, while *cx* and *cy* are the coordinated for the center of the circle. With D3.js we can easily manipulate the visual attributes of the circle like color, fill, thickness of the stroke etc.

When the frame for the radial plot is built with the axes and circles, we can draw the lines by creating a path with D3.js from point to point using moveTo(), lineTo() and closePath() methods. The *endpoint* function describes the end of each axis. Moreover, the data needs to be scaled on the axes. For example, the *Normalized Signal Estimation Error* has values in between 0 and around 0.05, while the *Estimated uncertainty* can have larger numbers up to 22 thousand. We extract the range of the point data from the minimum to the maximum value witth the methods of VTK.js. The variable *d* describes the normalization between 0 and 1.

```
var path = d3.path();
for(k = 0; k < num_cols; k++)
{
  mean[k] /= count[k];
  min = polydata.getPointData().getArrays()[k]
            .getRange()[0];
  max = polydata.getPointData().getArrays()[k]
            .getRange()[1];
  a = mean[k];
  d = (a - min) / (max - min);
  angle = -Math.PI / 2 + k * (Math.PI * 2 / num_cols);
  uv = endpoint(d * CRAD, angle, CCX, CCY);
  if(k == 0)
  {
    path.moveTo(uv[0], uv[1]);
  } else
  {
    path.lineTo(uv[0], uv[1]);
  }
}
path.closePath();
var p = svg.append('path');
p.attr('d', path).attr('stroke', 'orange')
 .attr('fill', '#FF9900aa')
 .attr('stroke-width', '2');
```

In a last step, merging the two parts of this visualization was done by rendering the radial plots into each div of each of the two subjects. Then, we adjusted its CSS code such as changing the function names. When loading one cluster, the window.onload function needs to render the 3D cluster and the 2D plot simultaneously.

3.2 Milestones

The following section describes the development process and milestones of this project:

3.2.1 Milestone 1. Explore the data: To extract features and access the details contained in the data, the general structure of the data needed to be examined.

3.2.2 Milestone 2. Implement the methods from VTK.js to access the data and get information from the documentation.

3.2.3 Milestone 3. Use D3.js to generate simple SVGs and manipulate them. We start by drawing simple lines and incremented the number of added elements until the radial plot has circles and the different axes for each scalar.

3.2.4 Milestone 4. Drawing the data from the vtp-file into the radial plot.

3.2.5 Milestone 5. Merge the 3D visualization in vtk.js with the 2D visualization in D3.js.

3.3 Challenges

The challenges of this project can be described as the following:

- Challenge 1: Get desired information from the data with vtk.js.
- Challenge 2: Understand the manipulations that are possible with D3.js.
- Challenge 3: Merge the two 2D and 3D visualization types into one whole program.

4 RESULTS

The final visualization result of the *Fibrestars* can be seen in the image 1. The three dimensional fibre clusters in the background shown in red can still be turned and viewed from all sides by holding the mouse clicked. The two subjects are shown side by side next to each other allowing comparisons between the 3D illustration and the 2D plots.

Figure 2 shows one radial plot with nine axes named by the scalars Normalized Signal Estimation Error, Estimated uncertainty, RTOP1 etc.

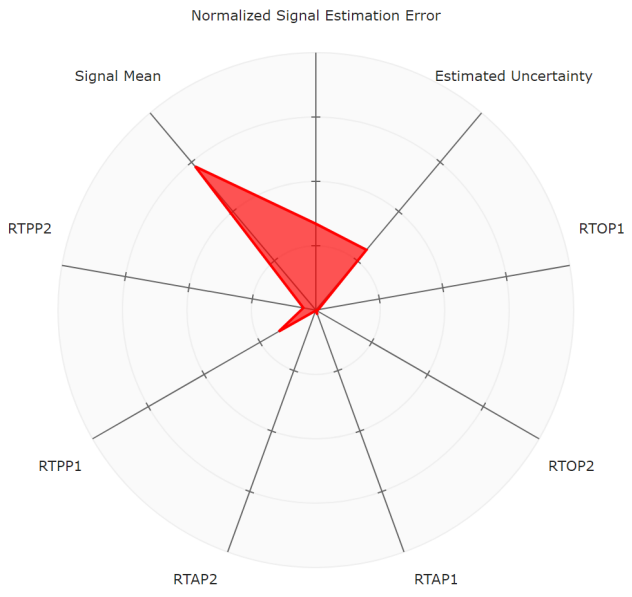

Figure 2: Radial plot with 9 axes showing the mean of the scalar data of the fibres

Figure 3 illustrates how different values in the radial plots can look for the different fibre bundles.

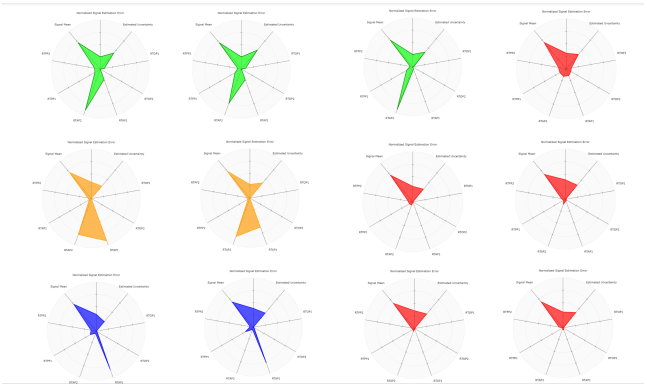

Figure 3: Different 2D plots for each fibre cluster drawn with D3.js as SVG

5 CONCLUSION

Fibrestars can help to support a better visualization and understanding of 3D brain fibre cluster data in 2D. For this task, we used VTK.js and D3.js, which are powerful JavaScript libraries for rendering the brain fibre data in the browser. The existing three dimensional visualization of fibre clusters is expanded with a two dimensional radial plot showing the details of the fibres in a more abstract way. The radial plots can facilitate the comparison between different subjects.

Future work can implement more features in the visualization such as calculating the standard deviation, minimum and maximum values of each axis. Furthermore, clicking on values in the radial plot can yield to changes in the 3D cluster, for example, changing the color of the fibres when clicking on one axis.

REFERENCES

[1] Ali K Al-Awami, Johanna Beyer, Hendrik Strobelt, Narayanan Kasthuri, Jeff W Lichtman, Hanspeter Pfister, and Markus Hadwiger. 2014. NeuroLines: a subway map metaphor for visualizing nanoscale neuronal connectivity. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (2014), 2369–2378.

[2] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D^3 data-driven documents. *IEEE transactions on visualization and computer graphics* 17, 12 (2011), 2301–2309.

[3] Alex Fornito, Andrew Zalesky, and Michael Breakspear. 2015. The connectomics of brain disorders. *Nature Reviews Neuroscience* 16, 3 (2015), 159.

[4] Jeff W Lichtman and Winfried Denk. 2011. The big and the small: challenges of imaging the brain's circuits. *Science* 334, 6056 (2011), 618–623.

[5] Haneen Mohammed, Ali K Al-Awami, Johanna Beyer, Corrado Cali, Pierre Magistretti, Hanspeter Pfister, and Markus Hadwiger. 2017. Abstractocyte: a visual tool for exploring nanoscale astroglial cells. *IEEE transactions on visualization and computer graphics* 24, 1 (2017), 853–861.

[6] Aditeya Pandey, Harsh Shukla, Geoffrey S Young, Lei Qin, Amir A Zamani, Liangge Hsu, Raymond Huang, Cody Dunne, and Michelle A Borkin. 2019. CerebroVis: Designing an Abstract yet Spatially Contextualized Cerebral Artery Network Visualization. *IEEE transactions on visualization and computer graphics* 26, 1 (2019), 938–948.

[7] Will J Schroeder, Bill Lorensen, and Ken Martin. 2004. *The visualization toolkit: an object-oriented approach to 3D graphics.* Kitware.