Interactive Data Visualization

Designing and Implementing an Interactive Visualization

Julius Laitala

May 10, 2020

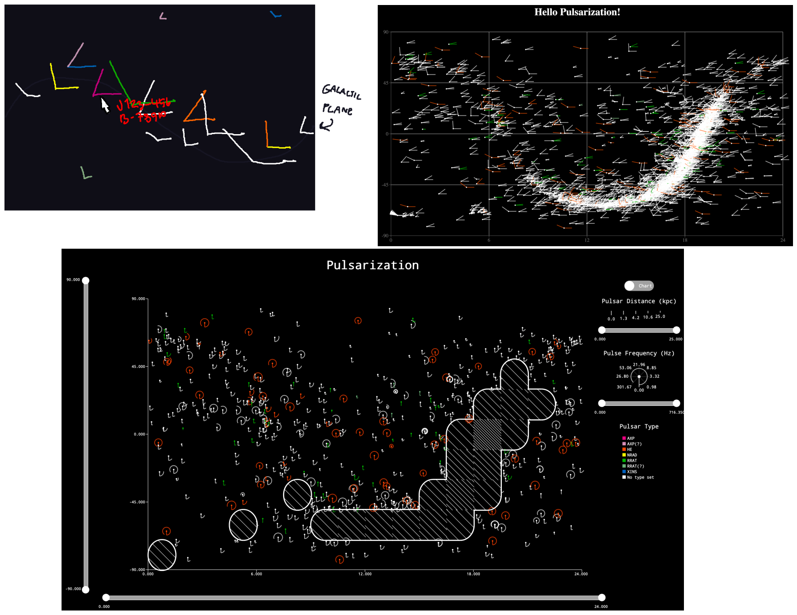

Figure 1: The first mockup (top-left), the first prototype as of commit `1e9a46d` (top-right), and the final visualization as of commit `c211dbd` (bottom).

As my interactive visualization project I visualized pulsar data. The journey from the first hand drawn mockup to the visualization at the time of writing this can be seen in figure 1. The visualization is available at `https://pulsarization.azurewebsites.net/`, and the source code can be acquired under GNU GPL v3 at `https://github.com/laitalaj/pulsarization`.

1 The aim

The three aims for my visualization that I set when beginning to work on this project were showing the distribution of different types of pulsars in the sky at a glance, enabling a visual search for pulsars with parameters in a user-given range and looking aesthetically pleasing.

2 Data

The data I used is found in `https://www.atnf.csiro.au/research/pulsar/psrcat/`. The source site contains some visualization options, but they are not very approachable or beautiful.

The source dataset contains a lot of fields, many of which I was open to including in my visualization. Due to time constraints, though, the visualization only includes the pulsar's J name; the right ascension and declination (i.e. the ecliptic coordinates) of the pulsar; the pulsar's pulse frequency; the pulsar's distance; and various pulsar types that the pulsar belongs to.

2.1 Data abstraction

The included fields can be abstracted as follows:

Field	**Data type**
J name	Nominal
Right ascension	Field
Declination	Field
Pulse frequency	Field
Distance	Field
Pulsar types	Nominal

3 Task abstraction

Some of the tasks that I wanted to make possible via the interactive visualization include seeing the distribution of pulsars (before or after filtering) in the sky at a glance; finding pulsars in a given patch of sky; selecting pulsars with parameters, such as distance and pulse frequency, in a specific range; and seeing accurate data on single pulsars.

The first example task is, pretty much word-for-word, one of my stated aims, while the last three tasks are more related to enabling a visual search, another aim set in section 1.

I've further discussed elements of these tasks when explaining my design rationales in the next section.

4 The design: choices, rationales, iterations

4.1 Visualization base: Scatter chart

I built the design based on a scatter chart. This was due to two factors: First of all, the scatter is a very natural way to show these stars, as if "on the sky". Secondly, position is a great way to visualize fields, and a scatter allowed very accurate visualization of the ecliptic coordinates.

Scatter chart was also a great choice for showing a lot of data at once, thus enabling the at-glance evaluation of pulsar distribution, one of my aims and one of the example tasks.

4.2 Frequency and distance

After some sketching, I decided to show pulse frequency and pulsar distance via length and angle. Initially, this was done via displaying each pulsar as an angle, with the size of the angle encoding the pulse frequency and the length of the angle's arms encoding the pulsar's distance. This encoding is used in the top-right image of figure 1.

I quickly noticed that this encoding wasn't too pretty, and that it conveyed the distance differences poorly, hampering the visual search task. After some iterative design, these evolved so that pulse frequency was encoded simultaneously via angle and length, and the pulsar distance was encoded via length and area. This gives a great contrast between both low- and high frequency pulsars, and near- and far pulsars, helping in the visual search task as can be seen at the bottom image of figure 1.

4.3 Dense areas

When I had done the first prototype, I quickly noticed that the densest areas on the galactic plane contained so many pulsars that the markers merged into a white mass, as seen in the top-right image of figure 1. This resulted in the said areas being unusable for the planned tasks, and additionally caused performance problems.

I mitigated this by creating a grid, and counting the number of pulsars in each grid square. All the pulsars in grid squares for which the pulsar counts were above a threshold value were discarded, instead showing on the grid square a blurry blob that showed the number of pulsars in it's area when hovered over. This helped with the performance problems while not causing any major loss of visibility, as the areas it affected were already pretty much unusable.

I further improved this design in later iterations by modifying the markers of adjacent blobs so that they formed a continuous area, and texturing the markers so that areas with higher pulsar density have denser textures. A simplified, simpler representation of these areas reduce the load on the user's senses, while still highlighting the dense areas, helping to get a general image of the areas. The final textured high-density areas are shown in the bottom image of figure 1.

4.4 Pulsar types and colors

I encoded the pulsar types via different colors for different types, and varying saturation for when the type is uncertain. I arrived at this decision pretty quickly, as both hue and saturation are known to represent categories well, and saw no need to change it during further development. The colors were chosen to be as far away as possible from each other on the color wheel for extra contrast, and the colors for the types that can have uncertainty were chosen so that the desaturated versions look as different as possible compared to the saturated versions.

I also chose a dark background, due to both it reminding the user of the night sky and making the actual data pop out.

4.5 J name

Even though the J name is an important, individual identification for a pulsar, I decided to show it only in tooltips that appear when the user hovers over a pulsar, and in the table view that the scatter view can be switched to. This was done due to text taking a lot of space, and due to the fact that the pulsar name isn't meaningful in the pulsar seeking tasks, to which category 3 out of 4 task abstraction examples fall to.

4.6 Interactivity

The first added element of interactivity are the tooltips – hovering over a pulsar in the scatter chart gives detailed information on it, including the otherwise hidden J name. Other interactivity in the visualization consists mostly of filtering tasks, fitting for the searching tasks listed in the task abstraction.

The user can filter the viewed area – that is, the ecliptic coordinates – via either large sliders next to the scatter axes or by brushing on the scatter chart. Filtering by pulse frequency and by distance is implemented via similar sliders, as they felt nice and intuitive. The user can also toggle display of specific pulsar types on and off, by clicking on them in the type legend.

In addition to these, to aid in the detail gathering task, the scatter can be switched into a table via a switch in the top-left corner of the visualization.

5 Things left undone

5.1 The table view

During the last few days of development I added a table view, which shows tabularized data on the currently shown pulsars. Intended for easy browsing of pulsar details, this table view was left mostly undone – getting the table to play nicely with the other components when used in lieu of the scatter took a lot more CSS time than I expected. Therefore, it's very very lacking in functionality.

5.2 Quality of life improvements

Many quality of life improvements that I had in mind didn't make it in. For example, I was planning on including a button for resetting the filters, and on having the brushing be a bit smoother and work even if you end your brushing outside the scatter area.

5.3 Including all the data

The source dataset is incomplete – the J name seems to be the only field that's filled for each pulsar entry. Instead of including all the data in the dataset, the source data's example tooling calculates values for the missing data based on the existing data. I did this for pulse frequency, but not for any other field. Therefore, not all the data is included in the visualization, as we don't include pulsars with no ecliptic coordinate and distance information. This results in, for example, every pulsar type not having a representative in the final visualization.

6 Lessons learned

6.1 Prefer ugly and quick over analysis paralysis

At a bunch of points during the project I started tuning fine details of the implementation before having any actual implementation ready and working. I'm aware that I'm prone to this *analysis paralysis*; instead of doing something quick and dirty that I know will work, I end up spending time on making the half-finished code cleaner and prettier.

Whenever I managed to avoid overanalysis, I usually managed to get something pretty as the end result, after a bit of refactoring before committing my changes. I spent so much time on this unnecessary fine-tuning of unfinished work that I'm now more than ever aware that I need to actively combat this paralysing mindset.

6.2 You really need to know CSS

Frontend isn't new to me. I've been dealing with it a bit every now and then in my work. However, my CSS skills consist basically of Googling what does every margin and flex option do, and then applying them blindly until the end result is pleasing enough. This is slow, especially when you try to do something a bit more exotic.

Before starting another project with a Javascript frontend, I'd therefore do some CSS learning.