Brushing Techniques for Exploring Parallel Coordinates

Diana Tofan
Technical University of Denmark
s172393@student.dtu.dk

Katarzyna Zukowska
Technical University of Denmark
s172091@student.dtu.dk

Abstract - Even though parallel coordinates can successfully solve the issue of visualizing multidimensional data, they often lead to overplotting when datasets are too large, making the visualization illegible and the underlying patterns undiscoverable. This is a huge problem since people fail to understand the information hidden in the data when they see nothing but visual clutter. This paper aims to improve the overall user experience when interacting with parallel coordinates and presents five types of brushing that are specifically designed to filter the data in different ways. Simple, multiple, line and angular brushing are implemented and a step-by-step example on how to use each is provided. Furthermore, a technical survey that requires users to use specific brushing techniques for answering questions related to the car dataset will be presented, along with an interpretation of the results.

1 INTRODUCTION

Visualizing multi-dimensional data has always been a challenging task for many and it has been proved that the human brain was only trained to perceive 2 dimensions [1]. One of the most well-known and widely used strategies to tackle this issue is represented by parallel coordinates, which are powerful enough to display an entire dataset in no time.

When representing data, parallel coordinates follow a slightly unique approach that allows users to visualize multiple dimensions all at once. Each dimension corresponds to a vertical axis and each item from the dataset is shown as a series of connected points along the dimensions/axes (see Figure 1).

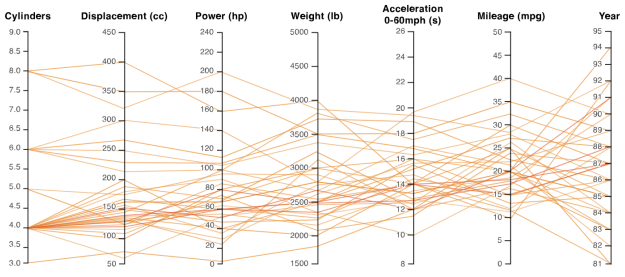

Figure 1: Example of a parallel coordinates plot

Even though parallel coordinates are successful enough to visualize multivariate data, they come with certain limitations that need to be solved or diminished for ensuring a smooth user experience. The central problem addressed in this paper is that large datasets create a lot of visual clutter and all the horizontal lines in the plot (known as "polylines") are subject to overlapping, thus obscuring any meaningful patterns that might be observed. These patterns are crucial in data analysis, since they are the key to revealing interesting stories about the data. Therefore we need to provide users with a safe means by which they can explore their datasets more effectively.

The previously mentioned issues with big data have already been solved by brushing techniques directly applied to the plot [2]. Basically, the main purpose of brushing is to select and highlight a subset of the data in a chart, hence simplifying the visualization and improving its readability. However one constraint that needs to be mentioned is the lack of flexibility, since most brushing techniques that are available nowadays only work for one dimension and consequently do not solve all the user's pains.

In the following sections, this paper will present an extended version of parallel coordinates, allowing users to explore the data much easier and switch between multiple types of brushing according to their needs. Multidimensional brushing will be introduced and user testing will be conducted on each type of brushing to find out which option is the most suitable for the novice user.

2 RELATED WORK

The concept of "smart brushing" was first introduced in 2015 [2] where brushing techniques were divided into four main categories: 0-order brushing (also known as "probing"), 1st-order brushing, 2nd-order brushing and higher order brushing. 0-order brushing refers to direct selection of lines in a plot, 1st-order brushing can be applied to individual axes and is based on interaction with two points, 2nd-order brushing is based on three points and two edges, whereas higher order brushing involves more than 3 points. A novel approach called "sketch-based brushing" was introduced, which dealt with higher order brushing by allowing the user to draw a sequence of interconnected line segments across parallel coordinates to filter data. Additionally, data-driven brushing was implemented which rendered the plot progressively and provided real-time feedback to the user during the interaction. These two techniques proved to be very convenient for the newcomers after preforming face-to-face user testing, the only drawback mentioned being the increased learning curve, therefore special training would be required to teach people how to properly use the newly implemented brushing techniques.

Another significant brushing technique was described in [3], which focuses on angular brushing. This type of brushing is based on two dimensions and its strongest point is that it can display the relation between different coordinate axes, facilitating the discovery of outliers within the data. In the same paper, parallel coordinates were combined with scatter plots, which deeply improved the visualization when composite brushes were selected.

3 METHOD

In the following section, a step-by-step use case scenario will be described. Through the paper, we will use the same dataset as the one from user testing, containing information about cars.

3.1 Uploading a dataset

We created a web application which renders parallel coordinates based on input .csv datasets. When first accessing the website (**link here**), the app asks users to upload a .csv file before being able to interact with the data (Figure 2).

Figure 2: Initial state of the app

In order to upload a dataset, the users have to press on "Choose a file..." button and upload a dataset from their local drive. For simplifying the testing part, a sample dataset named `cars.csv` file was provided and can be downloaded when clicking on the upper right icon from the header.

3.2 Choosing dimensions

The default behavior when uploading a .csv dataset is to render the parallel coordinates based on all dimensions, sorted in alphabetical order (Figure 3). However, one can add or remove dimensions in case they are not needed by toggling the corresponding dimension button and also change the order of the vertical axes by dragging them to left or right (Figure 4).

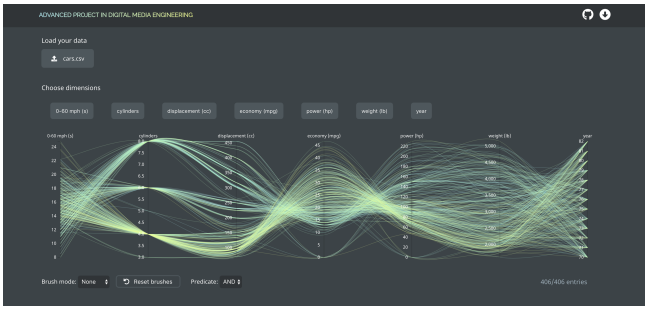

Figure 3: Default parallel coordinates plot. All dimensions are displayed alphabetically

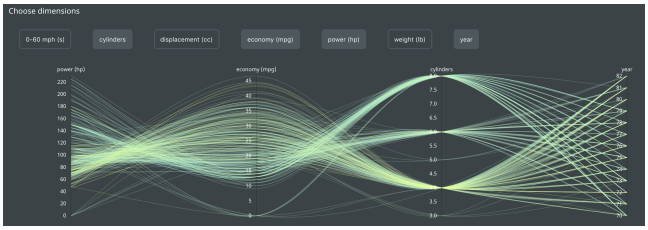

Figure 4: Altered parallel coordinates plot with 3 dimensions removed and axis order rearranged

3.3 Brushing techniques

Our method encompasses five types of brushing, both simple and multidimensional, and allows users to switch between the logical predicates AND/OR for comparing their results.

For selecting the brushing type, users must open the dropdown located under the parallel coordinates called "Brush mode" and choose the desired type. When done with brushing, one can press on "Reset brushes" to remove the brushes and redraw the parallel coordinates.

Each brushing technique will be thoroughly described next.

3.3.1 Direct selection. Direct selection, also known as 0-order brushing, occurs when you hover over one line in the chart. More specifically, each time the user places the mouse over a polyline, additional information about its corresponding dataset entry will be displayed in a table below.

If one clicks on a highlighted line, that line remains saved in the table for further exploration until removed. Users are allowed to select multiple lines from the chart and compare their values in the table, as shown in Figure 5. The "Reset" button deselects all the lines and the plot goes back to its initial state.

Figure 5: Direct selection of lines

3.3.2 Simple brushing. Simple brushing is the most basic form of brushing and can visually filter data by sweeping a virtual brush over each axis. Figure 6 illustrates an example of such brushing. It can be clearly seen that the central aim of simple brushing is to restrict the dimensions between a min and max value (e.g. Figure 6 can be interpreted as "cars with 4-6 cylinders, displacement between 100-240 cc, economy between 17-33 mpg, power between 80-120 hp and weight between 2000-3500 lb").

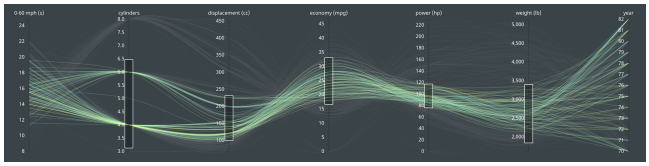

Figure 6: Simple brushing

3.3.3 Multiple brushing. The mechanics behind multiple brushing are similar to simple brushing, except that it allows users select multiple segments along one axis. For instance, in Figure 7 only

cars with 4, 6 or 8 cylinders are highlighted, action which couldn't have been performed by using only simple brushing.

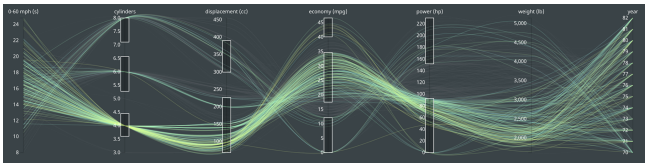

Figure 7: Multiple brushing

3.3.4 Line brushing. Line brushing is a 2D selection brushing method which can be performed by drawing lines in the areas between each 2 axes. In order to draw a line, one has to place the mouse cursor between 2 dimensions and then click & drag the line within that segment. After releasing the mouse button, the selection will contain all entries that pass through all of the active lines (Figure 8). The users can remove the lines and draw new ones by clicking anywhere close to their surrounding area.

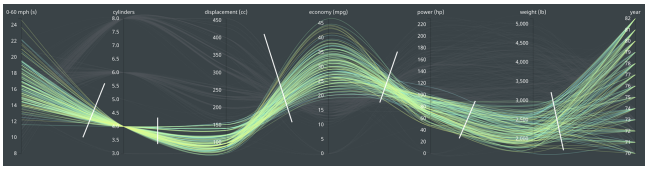

Figure 8: Multiple brushing

3.3.5 Angular brushing. As mentioned previously, angular brushing is a technique worth implementing in our application due to the strength it offers, i.e. the discovery of underlying relations between each dimension. Drawing angles is similar to drawing lines, however it involves a two-step process. First a simple line is drawn and afterwards the left or right corner of the line (where a small circle is shown on hover) is dragged up/down to expand the degrees within the angle. Angular brushing helps people identify whether there is a positive/negative correlation in-between values. As an example, in Figure 9 there can be observed a negative correlation between 1st and 2nd axis (0-60 mph(s) and cylinders) and a positive correlation between cylinders and economy. Furthermore, the outliers between power and weight can be noticed, since all but a few lines have a positive slope, hence the few others stand out.

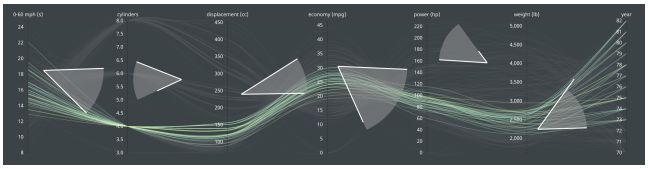

Figure 9: Angular brushing

3.4 Logical predicates

Along with brushing, users can also switch between the logical predicates AND/OR when more than one brush is active. In other words, predicate selection works when brushing was performed on more than one axis or in at least two inter-dimensional areas. The main idea behind predicates is that they create combinations of brushes using the logical operators AND/OR.

4 IMPLEMENTATION

The visualization was implemented using D3.js framework and the source code can be found **here**.

4.1 Brushing

Simple and multiple brushing are pretty straightforward to understand, since they both use the `d3-brush` module which is extremely well documented. On the other hand, line and angular brushing have a completely different approach and more external aspects had to be considered for ensuring a proper behaviour.

Line brushing starts when the user performs a click somewhere between 2 axes. When this event occurs, our application finds out between which 2 axes the line was started based on mouse coordinates. While dragging the line across the parallel coordinates, the algorithm makes sure that the ending point is within the bounds, and if everything succeeds, the line is drawn and all entries that the line crosses are highlighted. The same rules apply to angular brushing, except that the user draws an angle instead of a line and all the items that have a similar orientation are displayed ("diagonal" compatibility).

4.2 Line interpolation

This project underwent multiple iterations. Initially, the parallel coordinates were composed of straight line segments as seen in Figure 10, however after investigating how we can improve the readability of lines and reduce visual clutter we realized that a much better approach was to use curved paths (Figure 11). To achieve this, we used the monotone line interpolation method provided by D3.js.

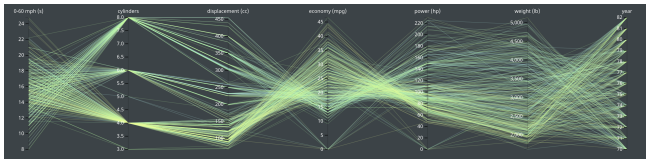

Figure 10: First iteration. Straight line segments

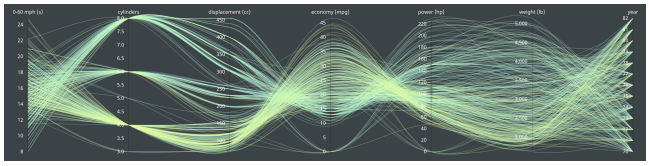

Figure 11: Second iteration. Curved paths

4.3 Color interpolation

In the first iteration, each line had the same gradient color and reduced opacity. However, after uploading larger datasets we noticed that it became much harder to perceive meaningful information within the plot (see Figure 12) and we adopted a different solution. Right now all lines have different successive colors, as in Figure 13. Implementation wise, this has been done by generating a list of consecutive colors within a specified range before rendering the parallel coordinates (in this case colors between light yellow and light blue). The list has the length equal to the number of lines that must be rendered, therefore each line gets assigned a unique color.

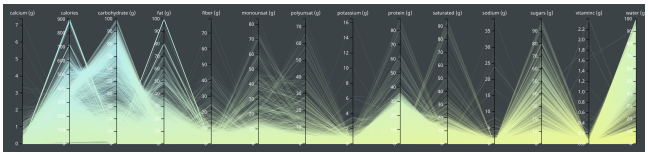

Figure 12: First iteration. Each line has the same gradient color

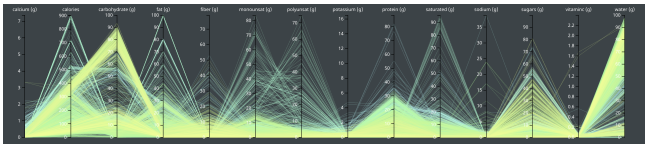

Figure 13: Second iteration. Each line has a different color

5 RESULTS

Five types of brushing techniques were implemented in order to interact with the parallel coordinates graph. To research the usability of each of them, the users were given specific technical tasks to solve that required the use of each brushing technique (**link here**).

In this section the results of face-to-face interviews and the survey will be presented in detail. The questions were related to the cars database that was provided to the users who had to download it and load to the website. The testing was done without introducing the subjects to how brushing techniques work, letting them figure it out by themselves. Our main assumption when choosing not to provide any hints was to check whether brushing techniques are intuitive enough.

First question was to *name the heaviest car*. The correct way of answering this question was to use direct selection of a polyline. All of the users performed this task correctly.

The users also performed well in the tasks asking to *name a car with a certain number of cylinders*. It proved that the subjects have good understanding of the simple brushing technique.

If it comes to multiple brushing, the performance was slightly worse. The question regarding this technique was: *'How many cars have a displacement between 0-150 cc or above 350 cc?'*. 55% of users answered to it correctly. 100% of the subjects answered the second question concerning multiple brushing correctly. Therefore, it can be concluded that the understanding of multiple brushing was also good. Additionally, it is worth mentioning that during the live feedback session many users stated that single brushing can be replaced by multiple brushing.

Next question was: *'Use line brushing to find out how many cars have a displacement > 350 cc and economy > 35 mpg'*. The intention with this question was to gain some insight on line brushing. 30% of the subjects stated that this technique did not work for them. Moreover, another 30% of the users answered the question incorrectly. Evidently, there was 60% of subjects who did not understand the technique, or were unable to use it correctly.

During face-to-face interviews we noticed that subjects had major problems when using the line brush. The common way of using this technique is to highlight the sets of lines between the two vertical axes (Figure 14 - Right). During the life testing, users tried to use this brush similarly to multiple brush, drawing lines on the axis (Figure 14 - Left). Using line brush in this way reduced its properties and led to confusion of the subjects.

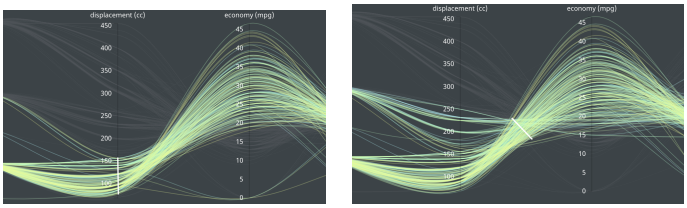

Figure 14: Left: Incorrect use of line brushing. Right: Correct use of line brushing

The last question was constructed to evaluate the use of angular brushing. The task was to *Emphasize the lines between "economy" and "power" that have a negative slope.* The users were asked to give the number of selected lines. Face-to-face testing resulted in a conclusion that subjects did not know that the line has to be dragged and that this is a way of creating an angular selection. Among the survey answers, only 30% of subjects answered correctly.

6 CONCLUSION

We managed to successfully implement an enhanced version of web-based parallel coordinates that contains multiple brushing techniques and solves the problem of visual clutter. Going through multiple implementation phases enabled us to develop a highly-interactive system that allows user-guided interaction with the visualization.

User testing showed that there is still some space for improvement. It is worth noting that the tested subjects were not introduced to brushing techniques of any kind and the majority did not know how to use line and angular brushing techniques in particular. Therefore, this part of tests needs to be revised and people require more training before fully benefiting from the features our application offers.

7 FUTURE WORK

In order to conduct better user testing, the future tests should be carried out with an introduction tutorial on how advanced brushing techniques work (especially line and angular brush). User interface

must be improved with short descriptions or hints on how to use certain brushing techniques.

Additionally, during face-to-face testing it was observed that understanding of the relationships between the data is dependent on axis order. This fact could be examined more deeply, and the future system could provide a suggestion on the most adequate order of axis.

One of the biggest challenges that needs to be solved is optimizing the rendering speed of large datasets and improving the user experience when interacting with such data. This could be done by introducing progressive rendering, which provides real-time feedback to users and instead of waiting for an entire memory-consuming chart to be displayed, they can see an animated version of the plot, where each single line being rendered shows up on screen only in a matter of seconds.

8 CONTRIBUTION

- Diana Tofan - website development, implementation of parallel coordinates and brushing techniques; responsible for "Abstract", "Introduction", "Related Work", "Method" and "Implementation" sections
- Katarzyna Zukowska - website development & design, implementation of direct line selection and table, user testing; responsible for "Introduction", "Results", "Conclusion" and "Future Work" sections

REFERENCES

[1] Dipanjan Sarkar. The art of effective visualization of multi-dimensional data. https://towardsdatascience.com/the-art-of-effective-visualization-of-multi-dimensional-data-6c7202990c57, 2018. [Online].

[2] R. Roberts, R. S. Laramee, G. A. Smith, P. Brookes, and T. D'Cruze. Smart brushing for parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1, 2018.

[3] H. Hauser, F. Ledermann, and H. Doleisch. Angular brushing of extended parallel coordinates. In *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002.*, pages 127–130, Oct 2002.