Interactive Edge-bundled Parallel Coordinates

by

Ziang Li
B.Sc., University of Victoria, 2018

A Project Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

Interactive Edge-bundled Parallel Coordinates

by

Ziang Li
B.Sc., University of Victoria, 2018

Supervisory Committee

Dr. Margaret-Anne Storey, Supervisor
(Department of Computer Science)

Dr. Charles Perin, Departmental Member
(Department of Computer Science)

ABSTRACT

Parallel coordinates are a well-researched visualization technique to represent multidimensional data. There are many variations of parallel coordinates for different application needs. This report describes a visualization solution for large multidimensional data. The report also proposes an evaluation plan to investigate non-linear data relationships through variants of parallel coordinates. Based on this proposal, a web-based application of bundled parallel coordinates was designed and implemented. This visualization supports different clustering methods and violin plots to discover data distribution. It also has a series of interaction features such as brushing, reordering of axes and zooming. A pilot study was also conducted to evaluate the perception of non-linear data relationships through variants of parallel coordinates, and the results helped the formation of a hypothesis: interactions help the discovery of non-linear data relationships, and standard parallel coordinates can better support such tasks than bundled parallel coordinates. An evaluation is needed in future work to evaluate this hypothesis.

Contents

List of Figures

ACKNOWLEDGEMENTS

I would like to thank: My supervisor **Dr.Margaret Story**, for her continuous mentoring, encouragement and support during difficult times and always being patient and kind.

Everyone in the **CHISEL** lab, the current and former members and collaborators: Soroush Yousefi, Andreas Koenzen, Ying Wang, Eirini Kalliamvakou, Jorin Weatherston, Arman Yousef Zadeh Shooshtari, Matthieu Foucault, Omar Elazhary, Trishala Bhasin, Daniel German, Neil Ernst, Charles Perin, and Cassandra Petrachenko. I am grateful to share all the good times with you and appreciate your help in my research.

A special thanks to Matthieu Foucault, Jorin Weatherston and Ying Wang, who provided valuable insights and technical support in my research. Cassie for her immediate help when I needed and the editing of this report.

Lastly, I would like to thank my parents and friends for their love, understanding and support along this journey.

DEDICATION

To my parents and family.

Chapter 1

Introduction

Exploring and analyzing multidimensional data can be a challenging task due to the increasing availability and size of data [10]. There exist many visualization solutions to help people understand and explore multidimensional data in different ways. Scatterplots (scatterplot matrix), parallel coordinates, and star glyphs are some examples of techniques that are ideal for multidimensional visualization [11]. Multidimensional data visualization has attracted attention since the 70s [12]. The need for representing information in a highly dimensional scale drives researchers to come up with new methods to visualize such information.

The parallel coordinates plot is a popular visualization technique that provides an objective representation of multidimensional data [13]. The plot maps n-dimensional data to a 2D plane with n equally spaced axes representing data attributes, with polylines connecting each data point on the axes. Users usually need to traverse through the polyline to see the trend of data, and the value is easy to compare just by looking at the position on the axis. Parallel coordinates' axes represent data dimensions, their numeric values are encoded on the axes often from high to low values when viewing from the top to bottom. The concept of parallel coordinates was first introduced by Inselberg [14] and now has become a common tool in the visualization community [13].

One of the biggest challenges with parallel coordinate visualizations is that visual clutter occurs when the size of data gets too large [13]. This will result in too many lines intersecting each other and make it difficult to see any useful information. Recent research papers on parallel coordinates have tried to tackle this problem from different perspectives. Since the core problem is dealing with a large amount of data, several papers [15, 4, 5, 6, 16, 17] introduced data mining techniques to preprocess

the data using dimensionality reduction or clustering techniques, so that the visualization outcome would be easier to interpret. During the visual exploration process, interaction techniques will enhance the user's perception of information that is being presented and will also contribute to a better user experience [18]. Some research addresses different interaction techniques to solve or smooth the cluttering problem in parallel coordinates [19, 8, 20]. Moreover, manipulating the axis layout of parallel coordinates in 2D or 3D environments [15, 21, 22] will provide users a different visual perspective. Some hidden patterns can be revealed by seeing the visualization from another angle.

In this project, I explore an edge-bundled layout of parallel coordinates inspired by Palmas et al. [23] for multidimensional data exploration. The visualization supports the discovery of data distribution by a toggle option for a violin plot. The interactive features (brushing/selecting, zooming, and reordering axis) allow direct exploration of the dataset which could help users find interesting patterns quickly. This project was developed as a web-application plugin within the Lodestone visualization platform. It uses React [1], D3 [2], PIXI [3], and other Node.js [4] libraries. This report also presents a design of an evaluation for non-linear data relationships in parallel coordinates, with results from a pilot study that leads to the generation of a few hypotheses that can be tested through future user studies.

1.1 Structure of the Project Report

This project report is organized as follows:

In **Chapter 2**, I give an introduction to multidimensional data visualization. This includes the background and some examples of common visualization techniques.

In **Chapter 3**, I talk about several related works to address clutter in parallel coordinates.

In **Chapter 4**, I introduce the edge-bundled parallel coordinates web application, its capability and user interaction techniques.

[1] *https://reactjs.org*
[2] *https://d3js.org*
[3] *https://www.pixijs.com*
[4] *https://nodejs.org/en/*

In **Chapter 5**, I discuss the design of an evaluation study in terms of the perception of non-linear data relationships in parallel coordinates.

In **Chapter 6**, I discuss limitations of this project as well as future work.

Finally, in **Chapter 7**, I summarize the purpose and implementation of this project and draw conclusions.

Chapter 2

Background

Multidimensional data visualization is a popular yet challenging task for visualization researchers [10]. Human perception often can only handle low-dimensional space, usually no more than three dimensions [11]. The goal of data visualization is to present high-dimensional data in a low dimensional space such that humans can easily understand the information behind the mass of data. In this chapter, I first describe the challenges of multidimensional data visualization. Next, I introduce several common visualization techniques to aid the exploration process of multidimensional visualization, including parallel coordinates. I also describe why traditional parallel coordinates cannot meet the increasing needs of large datasets and introduce bundling techniques to address them.

2.1 Challenges of Multidimensional Data Visualization

There are several aspects to consider when designing a visualization tool for multidimensional space. The increase of data collection methods leads to a surge in data sizes [24], such that multidimensional data often has a large volume, meaning the amount of data is also large. Scalability becomes a crucial challenge in this domain, and more specifically, perceptual scalability and interactive scalability are challenging [25]. Perceptually, our visual processing system has limited capacities to extract a large amount of information at once. Human brains can only handle limited cognitive loads to process that information. Limited screen spaces are also one of the constraints of perceptual scalability. Too much data on a limited screen space will result in the

visualization being too dense to see useful information. Screen resolution measured in pixels should be considered when designing visualizations for a large amount of data. The interactivity of visualizations also plays an important role in user experience. It takes time to process users' commands and to fetch the corresponding data back and display it to users. However, if actions take a long time to respond, they might interrupt fluent interaction and even freeze the application due to the large amount of data being processed. It is also a challenge to scale query processing for massive data while maintaining a smooth interactive experience.

2.2 Examples of Multidimensional Visualization Techniques

Many researchers have addressed the scalability challenges of multidimensional data visualization. Scatterplot matrix, radical visualization (RadViz), and parallel coordinates plot are some common examples I discuss below. It is important to realize the limitations of individual visualization techniques, so that we can design better multidimensional data visualization tools in the future.

2.2.1 Parallel Coordinates Visualization

The parallel coordinates visualization is common yet often regarded as an "expert-only" visualization [26]. The plot maps n-dimensional data to a 2D plane with n equally spaced axes parallel to each other. By connecting data points across axes using polylines, users can traverse through the plane to see the data relationships. Figure 2.1 shows an example of parallel coordinates with the Iris dataset of 150 entries. Each line is associated with one row of entries in the dataset. Together, by connecting all the lines across axes, a complete parallel coordinates plot will be available to explore.

Traditional parallel coordinates suffer from overplotting, a situation where too many data points crunch together to form "hairballs" in the visualization. Figure 2.2 shows this situation with the Cars dataset, which includes 5,076 data points with 5 variables. The colours separate cars with different fuel types. Even with 5,000 data entries, the parallel coordinates visualization is quite cluttered. A solution is needed to handle a large amount of data in parallel coordinates without compromising too much on the quality of visualization.

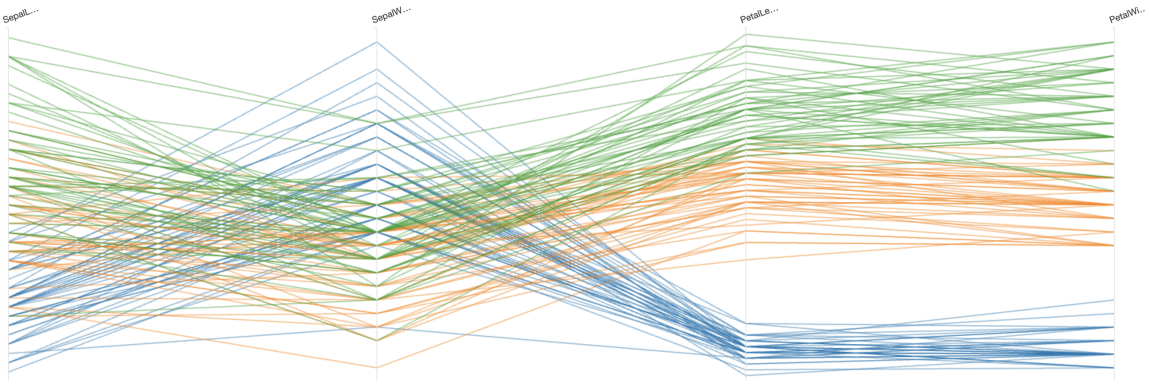


Figure 2.1: Parallel coordinates visualization with 150 data points

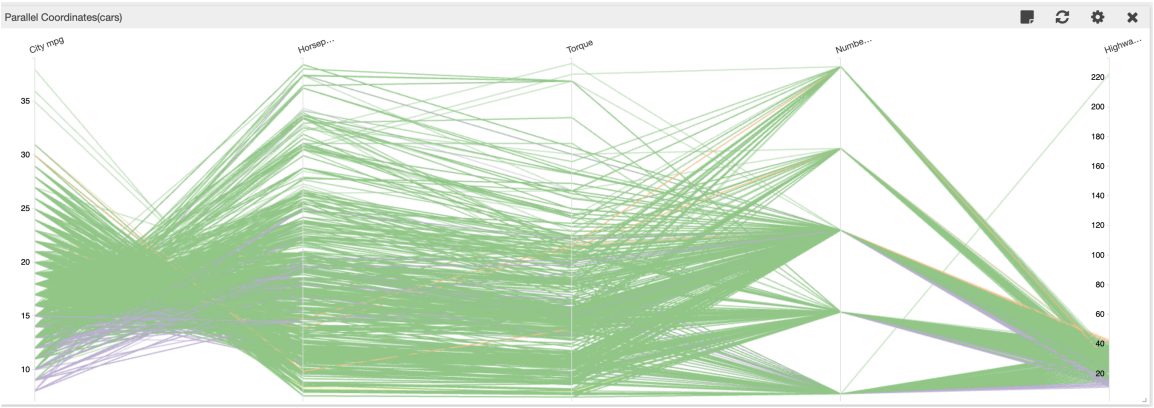


Figure 2.2: Parallel coordinates visualization with 5,076 data points

Visual clutter is one of the biggest challenges for parallel coordinates [13]. To efficiently solve this, clustering techniques were applied. Instead of showing individual lines in parallel coordinates, clusters can be highlighted in the visualizations, and the lines can be bundled to reveal more information and reduce clutter. McDonnell and Mueller [27] first explored a bundling technique on parallel coordinates. Each data-point was rendered as a poly curve and bent towards the centre of the axis. This reduces the visualization space, which then reduces visual clutter in the multidimensional dataset. Zhou et al. [5] explored curved edges in parallel coordinates and optimized their arrangement to improve visual clustering results without clustering the data. Heinrich et al. [28] described a C^1 continuous bundling technique to alleviate traditional cross-over problems from standard parallel coordinates. A recent study by Palmas et al. [23] introduces an edge-bundling layout for parallel coordinates. Their method uses density-based clustering and successfully reduced visual overhead, providing a faster overview and trend of data. For more information on the evaluation of parallel coordinates, readers can refer to a literature survey by Johansson and Forsell [29].

2.3 Scatterplots and Scatterplot Matrix

The scatterplots were designed to demonstrate the relationship of data in a two-dimensional space. It is one of the most commonly used visualizations that originated in 1833 according to Friendly and Denis' [30] analysis on the root of this visualization. The data are represented as individual points on two continuous orthogonal planes (commonly denoted as X and Y axis). With the growing demands of data analysis and the size of data, traditional scatterplots were limited to providing more insightful information. Many interaction techniques and variations of scatterplots were proposed to adapt to user needs.

Three-dimensional (3D) scatterplots have the advantage of showing another dimension that brings more possibilities in data exploration. Sanftmann and Weiskopf [2] proposed illuminated 3D scatterplots (see Figure 2.4) to address the difficulty of shape perception in traditional 3D scatterplots. Their method aimed to highlight an important structure in dense scattered data. Kosara el al. [31] demonstrated a practical application of interactive 3D scatterplots for scientific visualization. Interactions in 3D allow users to link the feature space to the actual object, giving them more freedom when exploring the data.

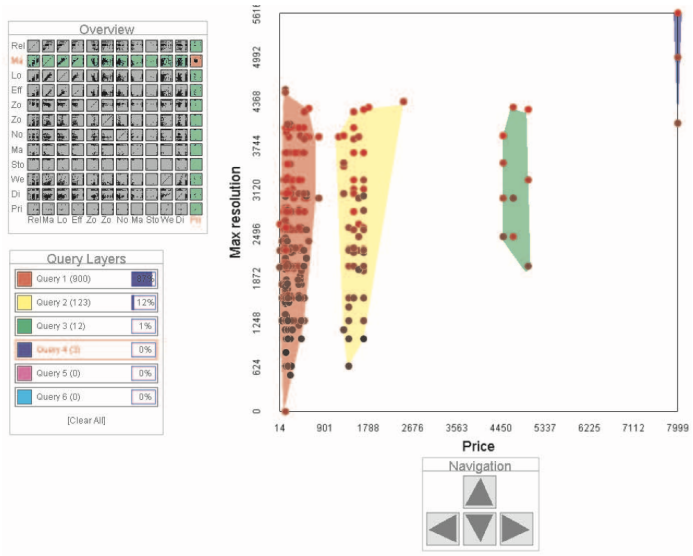


Figure 2.3: Interactive Scatterplot Matrix [1]

One scatterplot alone can only reveal information between two dimensions. A scatterplot matrix, however, can give an overview of many more data dimensions and relations between different configurations. Elmqvist et al. [1] presented an interactive method for multidimensional visual exploration using scatterplot matrix (see Figure 2.3). Transitions between different configurations were performed as animated 3D rotations which they have argued brought more semantic meaning to the nature of interactions.

2.4 RadViz

The radial visualization was first proposed as "circle segments" [32]. The basic idea of this visualization is to display data dimensions as segments of a circle. The circle is partitioned into k segments, where k is the number of dimensions in a given dataset. The result from this study shows the visualization is ideal for large amounts of multidimensional data. An example application of radViz is on the visualization of cancer genomics data: Schroeder et al. [3] summarized different techniques for visualizing multidimensional genomics data. Figure 2.5 shows a snapshot of Circos, a visualization tool that uses a radial layout to represent multidimensional oncogenomics data. The genomic coordinates of chromosomes are represented in a radial layout and enable users to explore relationships between distinct alternations.

In the next chapter, I introduce different variations of parallel coordinates from

existing literature, and discuss how they help overcome visual clutter issues through data mining and interaction techniques.

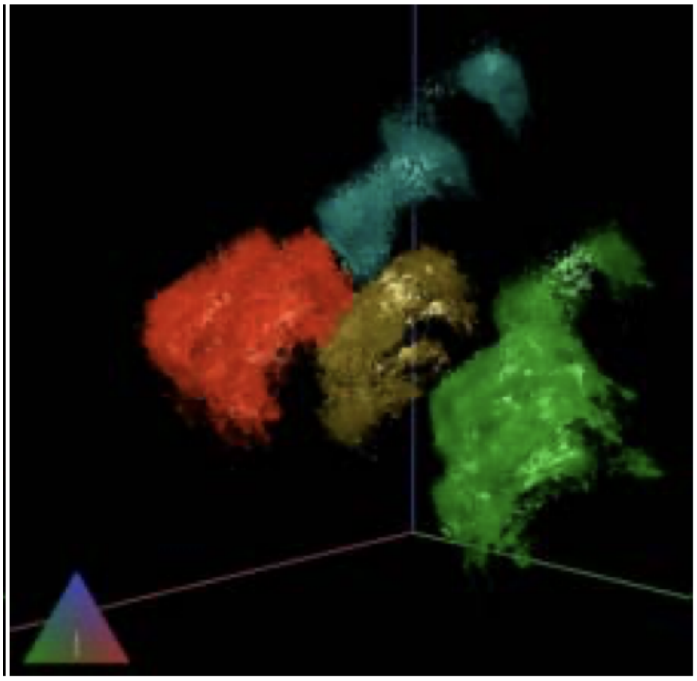


Figure 2.4: Illuminated 3D Scatterplots [2]

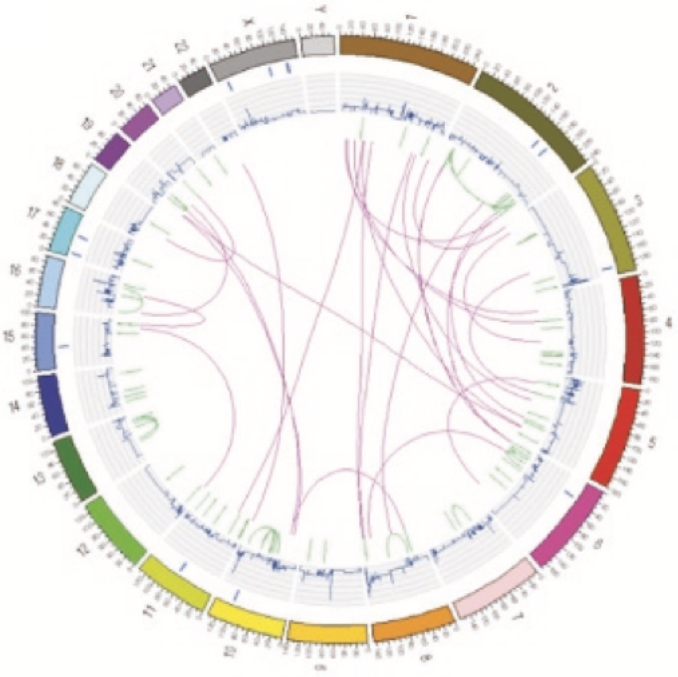


Figure 2.5: Circos plot for multidimensional genomics visualization [3]

Chapter 3

Related Work

Traditional parallel coordinates suffer from visual cluttering when the size of data gets too large. Many researchers have already proposed different variations of parallel coordinates to address this issue. Clustering, filtering, and sampling are some of the common algorithmic approaches to solve this problem. Interaction techniques such as zooming and brushing also help users navigate through the visualization. In this chapter, I discuss existing techniques to overcome clutter problems in parallel coordinates and how some of the insights can be used to drive a solution in chapter 4 of this report.

3.1 Data Mining Techniques for Clustering

Data mining techniques aim to manipulate the shape of data, such as dimensionality or size, to reduce the number of data items in a visualization such that it would be less likely to form visual clutter [10]. This section summarizes three data mining techniques: clustering, filtering, and sampling. Clustering algorithms target data items with similar attributes and group them to show a bundled line. Filtering techniques reduce the number of data items to be displayed. Sampling techniques display a subset of original data according to the calculated field, such as occlusion without removing data from the original source.

3.1.1 Clustering

Clustering is a popular technique to reduce visual clutter. Instead of showing all data items on the plot, clustering techniques group together data items with similar values

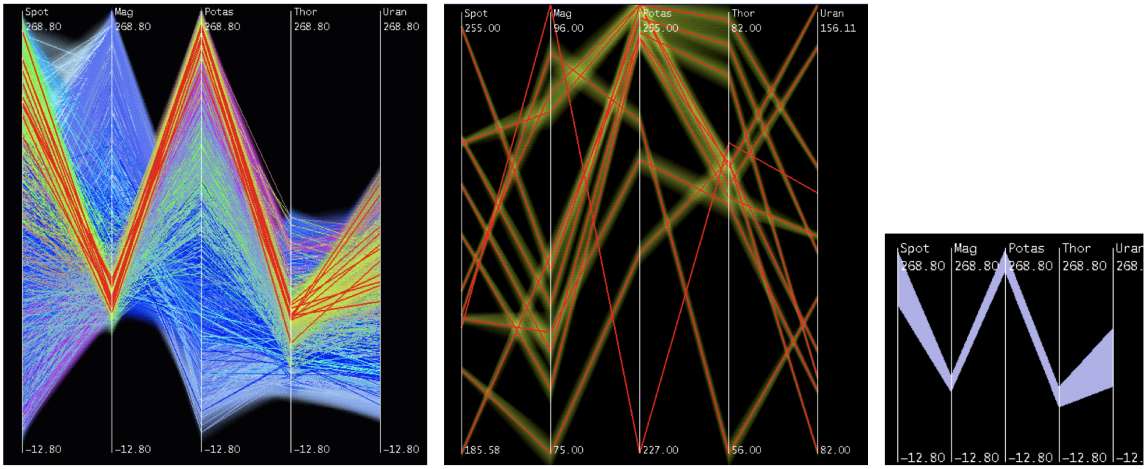


Figure 3.1: Hierarchical parallel coordinates with brushing to show specific region [4]

to show a bundled line. Johansson et al. [33] proposed a solution to construct clusters and use transfer functions to preserve high-precision textures as well as highlight important cluster characteristics of parallel coordinates. The high-precision texture is computed before plotting using graphics hardware. They achieved this by normalizing the intensity range by overlapping a maximum data item. The method has successfully handled 100,000 data items without showing significant clutter. While the method reduced visual clutter, there was no user evaluation performed. Without evaluation, we are not able to measure the benefits of certain types of visualization and will discourage widespread adoption of the method [34]. Fua et al. [4] uses a hierarchical clustering algorithm to construct a tree of nested clusters. A tree node T maintains summary information of all data points and sub-clusters rooted from it. More specifically, the number of data points in the cluster, the mean value of data points, and the minimum and maximum bounds of the cluster. Furthermore, colouring lines based on their cluster proximity gives users a clearer picture of the visualization (see Figure 3.1). Some interaction techniques like dimension zooming, drill-down and roll-up were also introduced. These interaction techniques help users navigate the visualization.

In another study, Zhou et al. [5] proposed a clustering algorithm that exploits curved lines in parallel coordinates instead of traditional polylines. Animations were introduced to enhance a better exploration process, as well as specification rules for coloring and opacity (see Figure 3.2). They demonstrated the effectiveness of their method by testing it on several datasets.

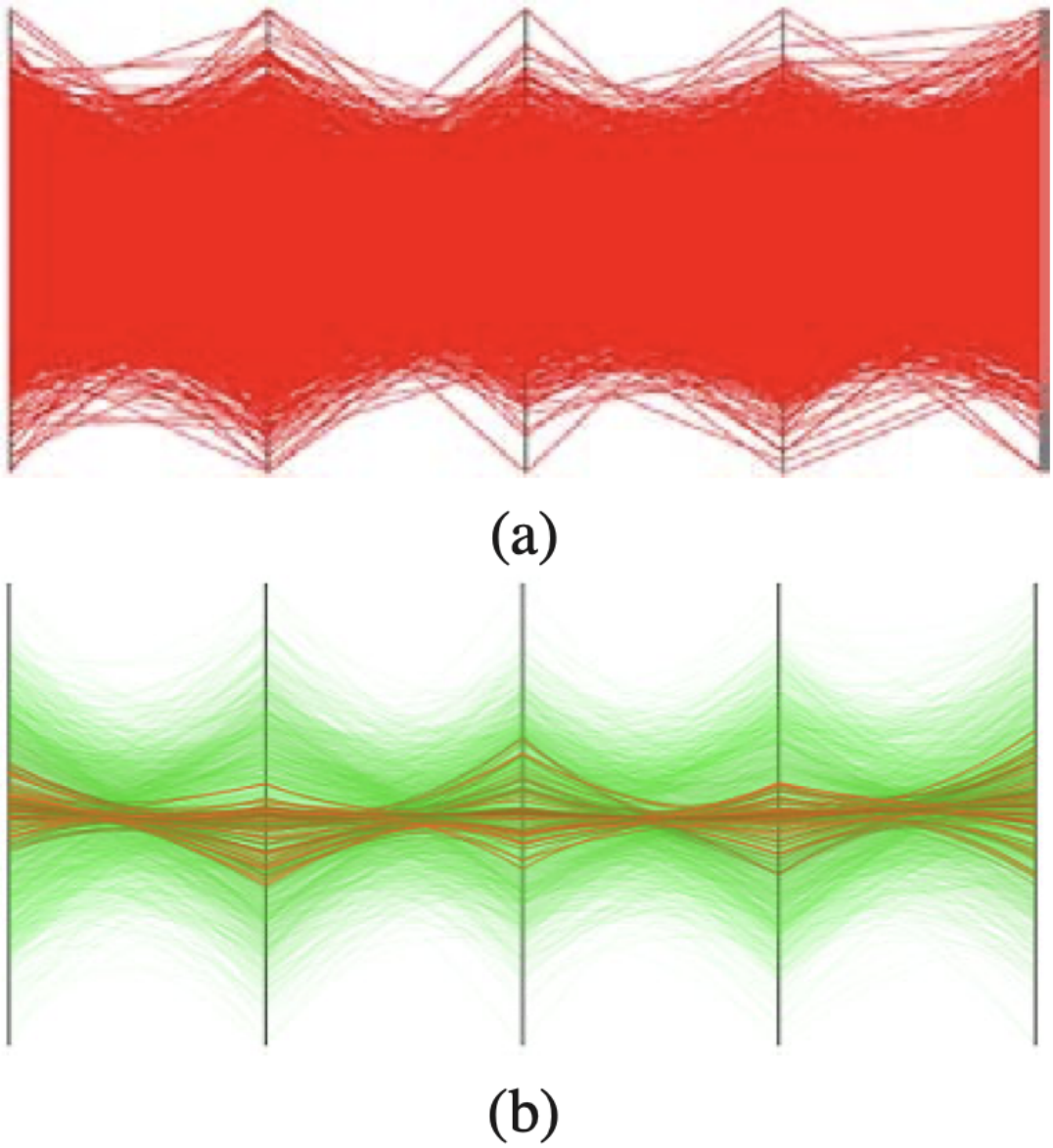


Figure 3.2: Experiments with 3,848 data items of the (a) original plot and (b) after visual clustering [5]

The results show their method can effectively reduce visual clutter and that the visualization works best after performing the clustering technique under a large dataset with the colour and opacity enhancement method in place. The colour and opacity enhancement method assigns opacity and colour to bundled lines according to the local density computed using a histogram method. Clusters become more distinguishable from each other with this method implemented. The main drawback of their method is computational efficiency; For a dataset with 7,736 data items, it will take more than two hours to compute the cluster based on their dual-core desktop. To date, the clustering technique has been proven to effectively reduce visual clutter through multiple studies [33, 4, 5]. However, most of the studies are lacking some evaluation to support their claims. A recent study [35] proposed aesthetic criteria to quantitatively access the quality of different clutter reduction techniques in parallel coordinates. Five aesthetic criteria were introduced to help select the best clutter reduction method: number of crossing lines; angle between two intersecting lines; parallelism; number of overlaps; and density of pixels. The authors concluded that the current criteria would not necessarily yield the best choice in every scenario, and the validity of the criteria could be confirmed by conducting future surveys.

3.1.2 Filtering

Another common way to perform clutter reduction is through filtering. This technique essentially removes part of the data from the display and can significantly reduce clutter. The method proposed by Rosenbaum et al. is called progressive parallel coordinates (PPC) [6]. This method was based on progressive refinement that uses recursive interval subdivision, hierarchical clustering, and wavelet transformation techniques to reduce the number of data points to be rendered. Users essentially see the filtered version of a parallel coordinates display with a limited amount of data shown (see Figure 3.3).

At the end of Rosenbaum's studies, they asked 43 participants to perform several tasks and compare the effectiveness of detecting patterns in both traditional parallel coordinates and progressive parallel coordinates. The results show that the progressive parallel coordinates display has no significant advantage in the correctness of pattern detection. However, progressive parallel coordinates required less available data, only 37.04% of data needed in order to identify patterns. They also concluded that PPC improved the overall user experience in terms of assistance and acceptance.

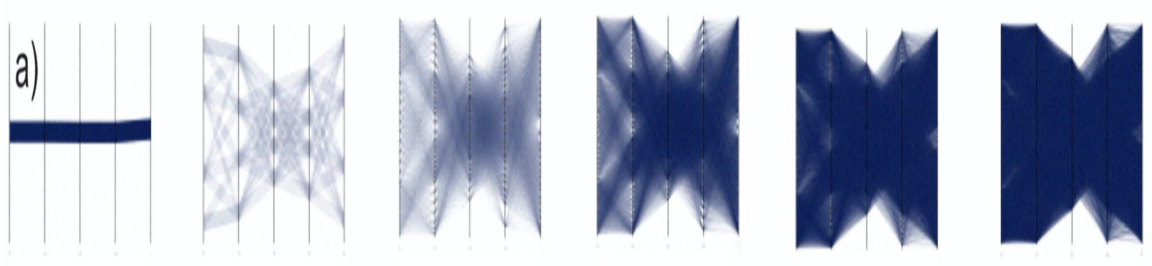


Figure 3.3: Progressive parallel coordinates with different levels of refinements [6]

Although PPC is a good technique to reveal patterns from an early stage, it also faces several drawbacks. The number of data items have been reduced to produce the visualization so it is hard to reveal detailed views when users are required to gain such information. Also as mentioned by the authors, the PPC is insufficient to handle datasets that change constantly. Artero et al. [36] investigated a method to calculate density and frequency from data. They use such information to filter out certain information from the dataset such that the parallel coordinates visualization is less likely to form visual clutter. The authors tested their method on a variety of datasets. The results show users can interactively identify and extract information of clusters, even on a large dataset with 1,000,000 records and 200 attributes. This method creates lines in the display that do not map directly to data points. Instead it displays the trend and clusters of data so it is hard for users to find out what the actual information from the data is.

3.1.3 Sampling

An interesting solution to solve the visual clutter problem is through random sampling. Two studies [17, 37] show random sampling is helpful to reduce the density of display by only displaying a portion of the data. This method is also helpful to other visualizations like scatter plots. To automatically adjust the sampling rate to adapt to users' needs, an efficient method to calculate the occlusion of intersecting lines in the parallel coordinates is needed. Ellis and Dix [7] described three algorithms, namely raster algorithm, random algorithm, and lines algorithm, to calculate the occlusion and sample for a particular region of the visualization (see Figure 3.4). They followed their studies with an empirical study to show that the random algorithm performs the best in normal cases and is relatively stable in extreme cases. This approach has been proven effective. However, by manipulating the data, we lose certain information in

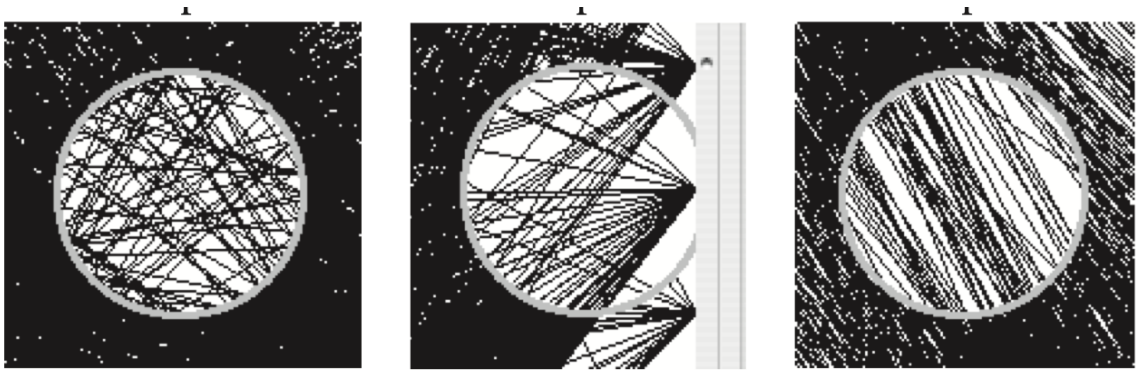


Figure 3.4: Sampling lens to see specific region of parallel coordinates [7]

the display.

Data mining techniques target the visual clutter problem at the data level and provide a quick and efficient way to reduce the amount of information that is being displayed. This approach has been proven effective in visualizing large amounts of data in parallel coordinates [33, 4, 5, 6, 36, 7]. They are relatively easy to implement in any visualization because the algorithms already exist. However, using this method alone will not be sufficient for a complete data exploration process. In order to reveal more information that might be hidden in the visualization, other techniques, such as user interactions, should also be in place to aid the exploration process.

3.2 Interaction and Rendering Techniques

Effective user interaction techniques play an important role in data visualization [38]. Without proper interaction techniques, any data representation method would not be able to demonstrate its full potential [18]. Several studies addressed interaction techniques for parallel coordinates to improve the usability of the display.

Hauser et al. [19] introduced a new brushing technique that enables a fast and flexible exploration of parallel coordinates visualizations when multiple dimensions are presented. In addition to standard brushing, users can specify a sub-set of slopes to brush and this gives them more flexibility to select fewer data points. They tested their solution in several applications and the results show it to be effective in terms of helping users explore information in a large dataset. They also suggested that the combination with other visualization techniques like scatterplots will result in a better exploration process, and more interaction techniques like axis re-ordering could be added to maximize the effectiveness of angular brushing. In another study, Guo et al. [8] proposed an interactive clustering method that allows users to have controls on the regions to be clustered. By using an attractive operator, users can choose a subset of data and drag the neighbouring parallel coordinate lines to form a cluster in a chosen location of the plot (see Figure 3.5). Such processes can be done collaboratively on a multi-touch display. They tested their solution on various datasets and the results show their method can help users successfully identify clusters thus reducing visual clutter in a large dataset with five variables and 16,384 data items. A further study involved 9 students from different departments at their institute showing the average accuracy of correct identification of clusters varied from 90% to 100%. This solution has given users the highest degree of freedom in terms of clustering. However, the

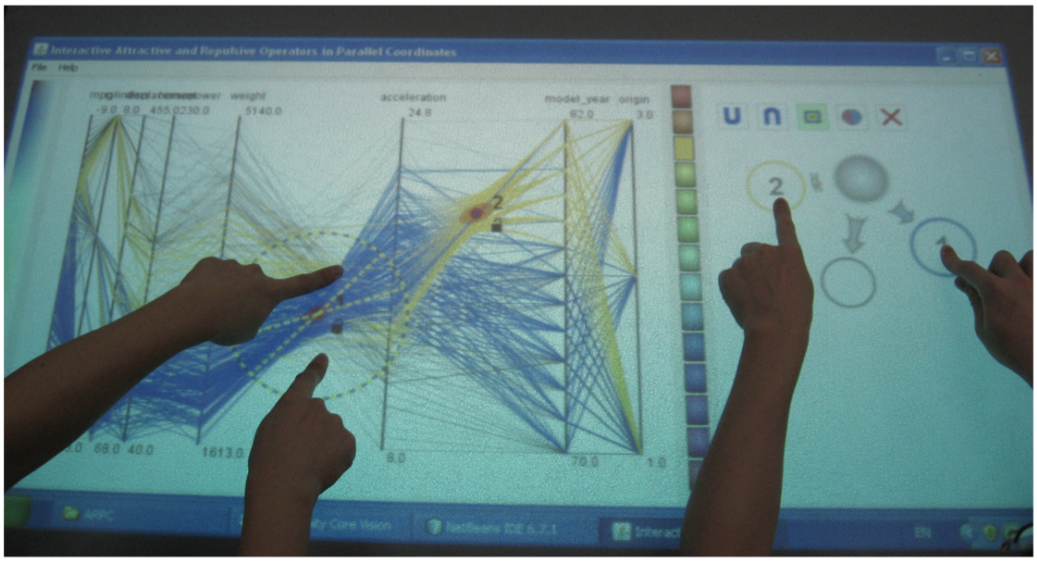


Figure 3.5: Collaborative parallel coordinates with touch screen [8]

solution works best when users can identify as many clusters as possible and perform the manual clustering process correctly. Automatic algorithms should be in place in case users fail to perform the manual steps.

Direct manipulation [39] is another technique to interactively reduce visual clutter. The goal is to manipulate a visualization and get visual feedback within 0.1 seconds. This way, the user would perceive the visual feedback as a direct response to the manipulation. Siirtola [20] described two interaction techniques to help users gain more information in a short amount of time. Polyline averaging was used to summarize overlapping polylines such that the visualization becomes more readable. The correlation coefficients between subsets of polylines were also visualized to reveal patterns. The author mentioned the usefulness of this method needs to be verified in future investigations.

Visual clutter in parallel coordinates start to form when the size of data to be visualized gets large. Data mining techniques such as clustering, filtering, and sampling can help reduce the amount of data in the visualization to minimize clutter. Interaction techniques such as brushing and axes re-ordering can also help reduce visual clutter. In the next chapter, I propose an interactive edge-bundled parallel coordinates tool for visualizing multidimensional data. The tool also addresses visual clutter for large datasets.

Chapter 4

Implementation of Edge-bundled Parallel Coordinates

In this chapter, I explain my solution to the problem set described in the previous chapter built on an application called Lodestone. Lodestone is a visualization platform that provides multiple ways of connecting to data sources and visualizations. The web-based application was developed using React.js and Typescript, and can be exported as a standalone Electron application that runs on multiple operating systems. The Lodestone project was developed with the support from the Natural Sciences and Engineering Research Council of Canada (NSERC) [1], Thales Canada [2] and Defence Research and Development Canada (DRDC) [3]. Figure 4.1 shows an overview of the Lodestone application, where users can choose a certain type of visualization to work with. Users need to configure visualizations before seeing them in the panel and configuration parameters vary for different visualizations. Figure 4.2 shows the configuration for bundled parallel coordinates.

4.1 Data Clustering

The parallel coordinates plot is a popular visualization ideal for multidimensional data, but considering the size of data we are dealing with, standard parallel coordinates cannot continue to provide the level of clarity we needed. However we still appreciate how parallel coordinates can provide an overview of all dimensions in a

[1] $https://www.nserc-crsng.gc.ca/index_eng.asp$
[2] $https://www.thalesgroup.com/en$
[3] $https://www.canada.ca/en/defence-research-development.html$

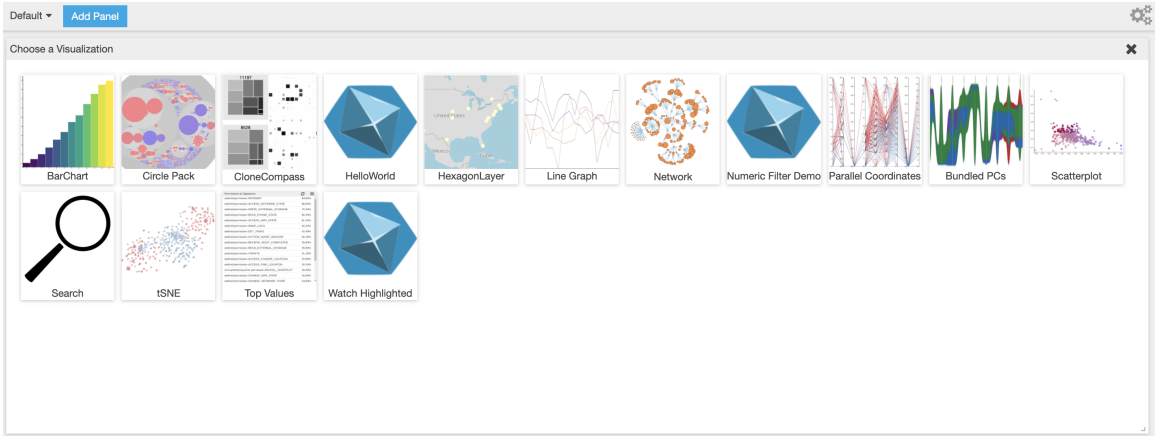


Figure 4.1: Lodestone Application

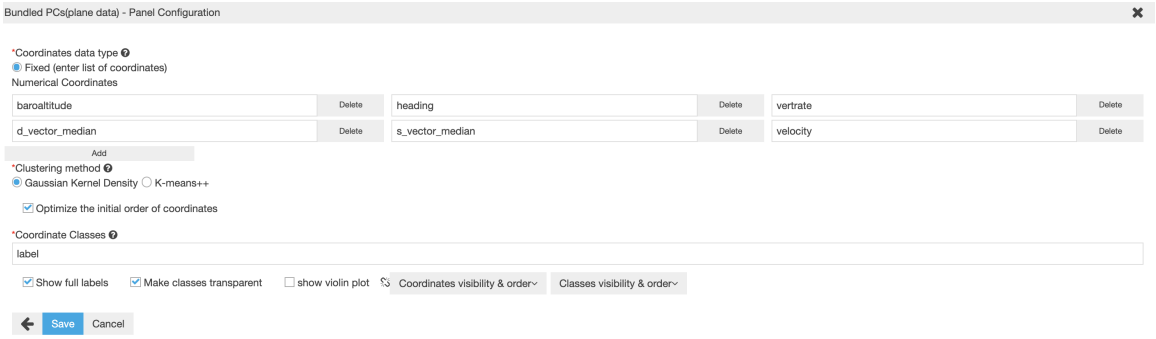


Figure 4.2: Configuration panel of bundled parallel coordinates

single view, so we started to look into different variations of the visualization.

The idea of interactive bundled parallel coordinates originated from Palmas et al.'s work [23]. Instead of rendering data points line by line, a clustering algorithm was used to aggregate data into clusters and presented as bundled lines in the visualization. The edge bundling technique is not new to the visualization community. Holten [40] applied hierarchical edge bundling to a radial visualization to reduce visual clutter and to better illustrate implicit adjacent edges. Cui et al. [41] proposed a geometry-based edge bundling technique that aims to provide informative and less cluttered visualization layouts in maps. The presentation of edge-bundled parallel coordinates makes the visual graphics more abstract and it helps give an overview of the data at first sight. In this project, I adapted the original Gaussian density clustering method from Palmas et al.'s paper [23] described below, and added another clustering method to give users a more flexible way to cluster their data, and with faster performance.

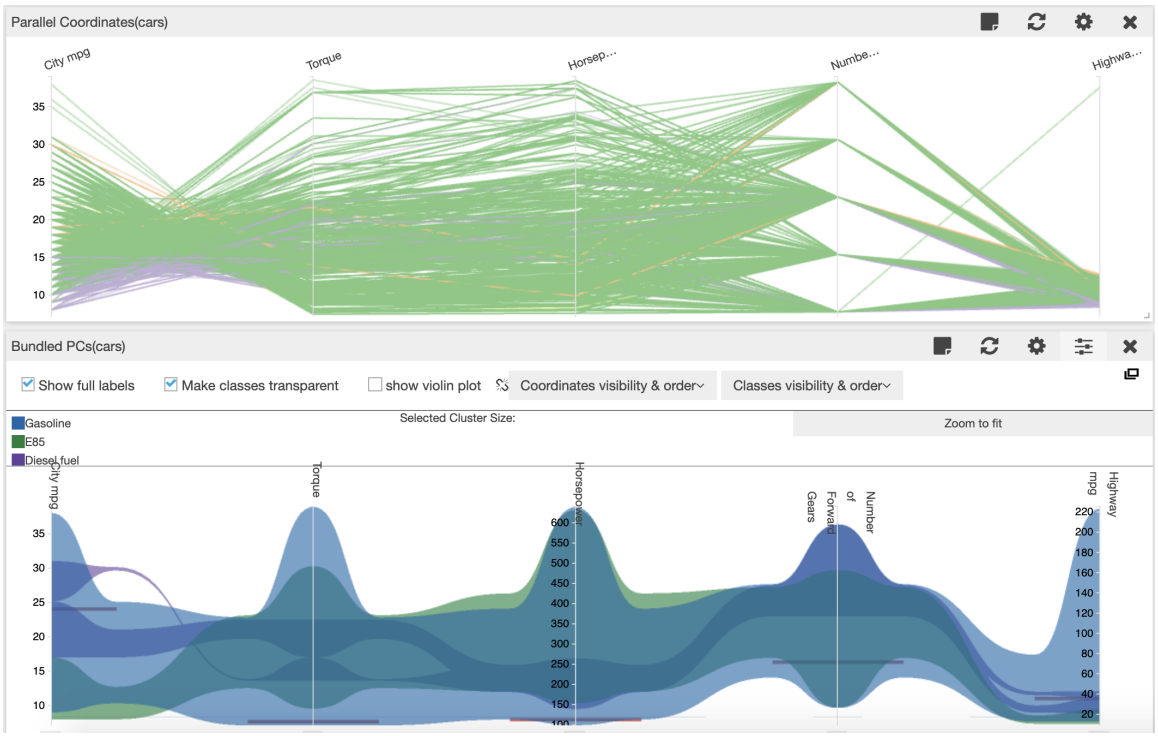


Figure 4.3: Standard parallel coordinates (top) and edge-bundled parallel coordinates (bottom) with Gaussian kernel density clustering on Cars data with 5,076 five dimensional items

4.1.1 Gaussian Kernel Density Clustering

A cluster can be defined by density. The density f(x) of x_i is defined as follows:

$$f(x) = \frac{1}{n\sigma\sqrt{2\pi}} \sum_{i=0}^{n} e^{-\frac{1}{2}(\frac{x_i - x}{\sigma})^2}, \text{where } \sigma \text{ is the standard deviation.} \quad (4.1)$$

A Gaussian kernel density estimation Node.js library [4] was used to calculate the kernel density for clustering. We use a 4-tuple to define each cluster: the local maximum, local minimum, the centre of the cluster, and the size. We use Bezier curves instead of straight lines used in traditional parallel coordinates to connect data points. Between clusters in the adjacent axis, a polygonal strip was used to connect them: A polygonal strip is a Bezier curve with two added offset curves that are perpendicular to the tangent of the original Bezier curve. The degree of offset is determined by the position of the axis in two adjacent axes pairs. Rendering a large number of Bezier curves is time-consuming and will clutter the visualization. The

[4] https://www.npmjs.com/package/kernel-smooth

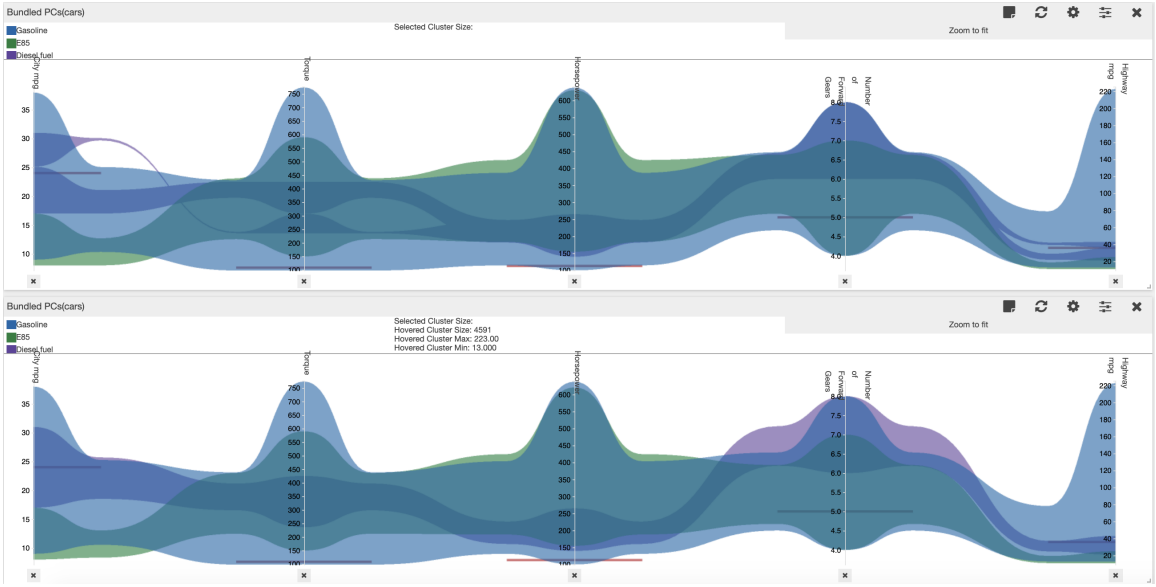


Figure 4.4: Kernal density clustering (top) and K-means++ clustering (bottom) on bundled parallel coordinates

idea of polygonal strips is to capture a certain range of Bezier curves such that the visualization is more visually appealing. A more detailed layout of bundled parallel coordinates can be found in section 3 of Palmas et al.'s original paper [23]. Figure 4.3 shows the comparison of standard parallel coordinates and edge-bundled parallel coordinates with Cars dataset of 5,076 data points.

Due to the nature of this implementation, subtle details about the data cannot be seen in the visualization. For example, in the standard 2D parallel coordinates, users can usually observe a single data point by using filtering or brushing techniques. However, in bundled parallel coordinates, it is not possible to show a single data point unless the data are so extreme such that no other data can be clustered into the same cluster. That is, one data point is essentially a cluster by itself. The default maximum number of clusters per class was set to 8. However, if a certain class cannot produce 8 clusters, fewer clusters will be shown.

4.1.2 K-means Clustering

After we implemented the original clustering method of bundled parallel coordinates from Palmas et al.'s paper [23] as a web application, we noticed that when the size of the data becomes large (>50,000 data points), the total rendering time and computing time takes more than 10 seconds, which was not ideal for an optimal user experience.

We tried streaming the data ingestion process to reduce the total processing time, however, an improvement was not apparent. Another way to optimize the running time was to find another clustering algorithm. K-means algorithm is one of the most popular clustering algorithms that was first brought up by MacQueen [42]. It is known for its simplicity and scalability to large datasets. We adopted the K-means++ algorithm from Arthur et al.'s work [43], a variation of K-means clustering that has a randomized seeding process to optimize time complexity. Users will define how many clusters they want to see within each dimension in the configuration, then the algorithm will generate the same number of clusters per class. Figure 4.4 shows K-means++ clustering on bundled parallel coordinates. Compared to density clustering, the clusters are generated differently but the time complexity is reduced to O(log k), which significantly improves the user experience.

4.2 Showing Distribution on Bundled Parallel Coordinates

One disadvantage of bundled parallel coordinates is that it displays information in an abstracted way so that some data properties, such as distribution among each axis, becomes difficult to interpret. The width of clusters does not represent how much data is inside, which rather it shows a range of possible values. In this case, a cluster with a broad minimum and maximum value could look larger in the display but have fewer data points than a narrower cluster which has a smaller range of minimum and maximum value but with more data points inside. To better display the distribution of data, we need a representation of data distribution that works nicely with bundled parallel coordinates.

4.2.1 Histogram

Histograms are a popular visualization technique to show data distribution that has been widely adopted in visualization research. Hauser et al. [19] implemented a bar chart histogram and overlay on the top of parallel coordinates to enable the discovery of data distribution on each axis. McDonnell and Mueller [27] also adopted histograms in their illustrative parallel coordinates and rendered them as faded quadrilateral strips. Their implementation allows users to see the data distribution for each cluster rather than in each axis from Hauser's method. Similarly, Janetzko et al. [9] also

overlaid a colored stacked bar chart, violin plot, and box plot on parallel coordinates to show class frequency.

The initial design of showing data distribution on bundled parallel coordinates was sketched on paper as shown in Figure 4.5. Different classes are mapped with

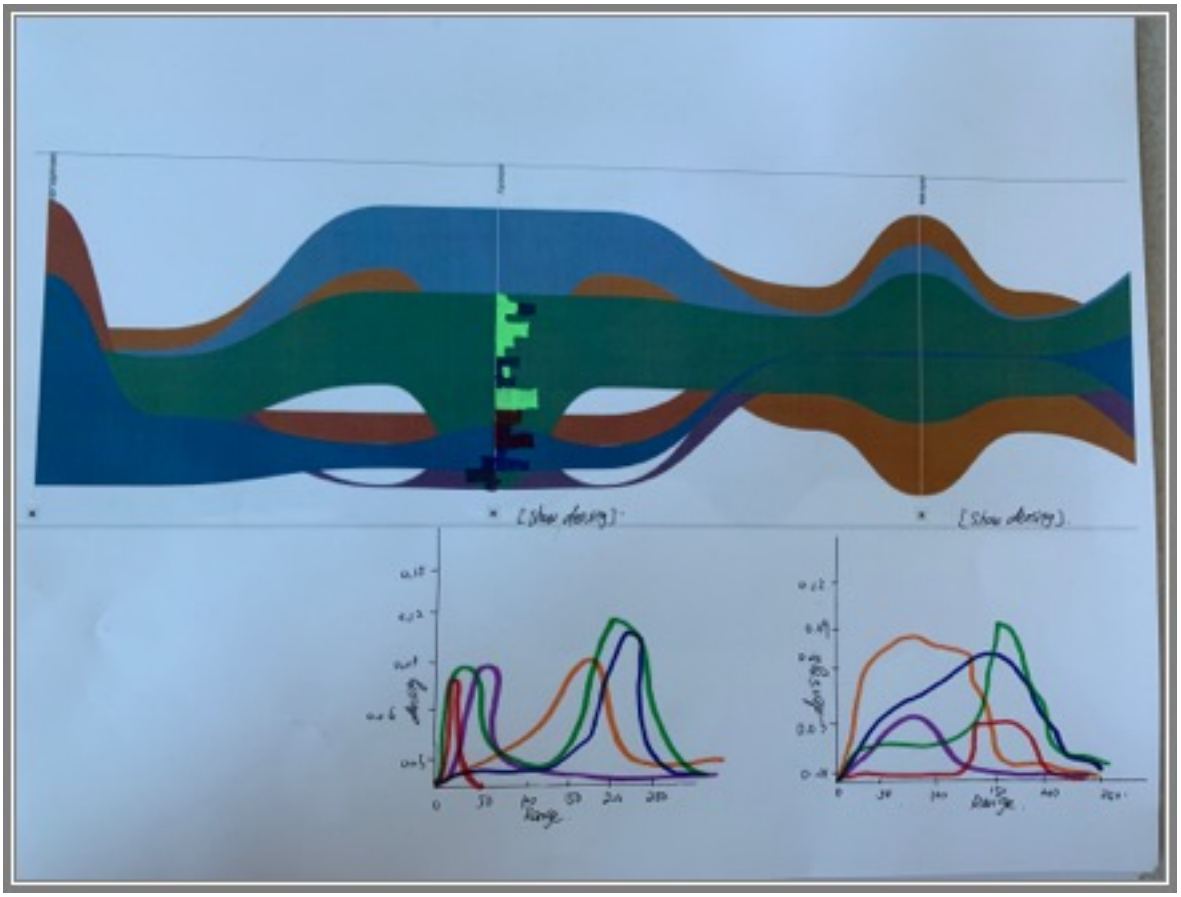


Figure 4.5: Sketch of stacked bar charts

corresponding class colors in this sketch. At the bottom, I also included a density graph to complement this method. However, after getting feedback from several colleagues, this method was abandoned due to a possible confusing presentation: too much information was shown at once, leading to cognitive overload. This process let me explore the violin plot, which is an alternative method of showing distribution.

4.2.2 Violin Plots

Violin plots enable the discovery of data density. They work similarly to the box plots except the distribution can be seen clearly from their shape.

Janetzko et al. [9] implemented various density visualizations including violin plots. Figure 4.6 shows their implementation.

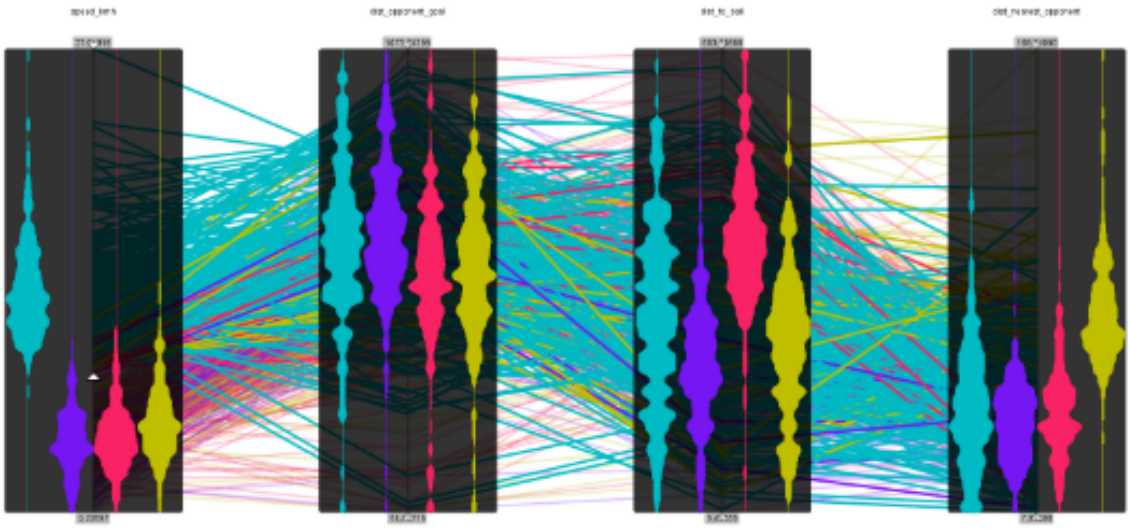


Figure 4.6: Violin plots with Janetzko's [9] implementation

The first iteration of my design was based on this idea, in which I overlaid violin plots with colours representing different classes on top of bundled parallel coordinates. Figure 4.7 shows the initial design sketch. One of the biggest disadvantages of this

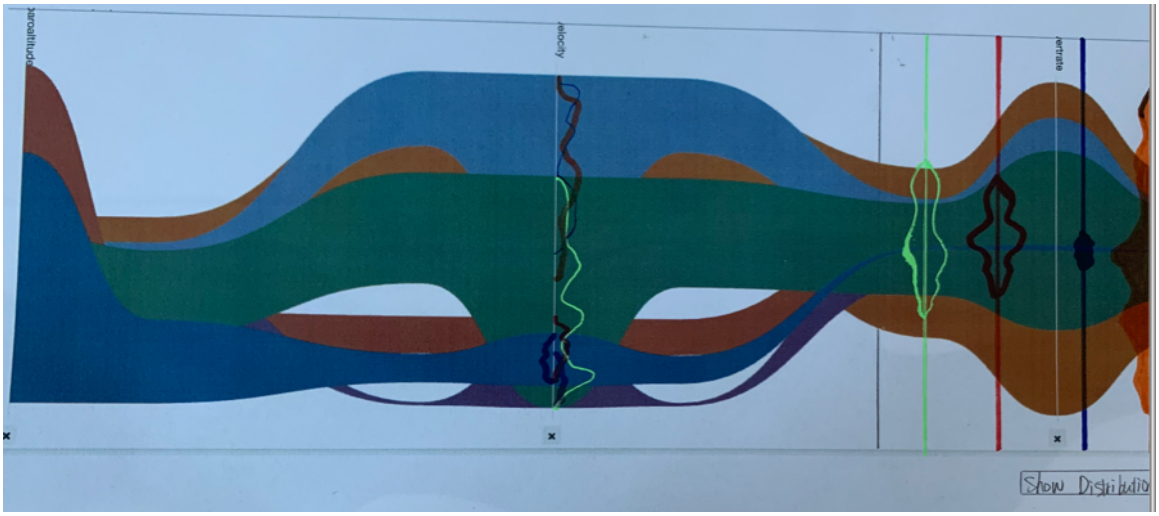


Figure 4.7: Initial idea of using violin plots in bundled parallel coordinates

approach is that when the data contains more than 5 classes, the violin plots will get expanded quickly, thus introducing a high cognitive load and occluding the main underlying visualization. Having the ability to show density on bundled parallel

coordinates should give users a general idea of where the majority of data is located. It is less important to see the detailed data distribution within specific violin plots.

4.2.3 Implementation of Violin Plots

A violin plot is generated for each axis instead of for classes to best accommodate both efficiency and aesthetics. The computation of histograms to generate violin plots happens at the client side and returns an SVG element that contains the violin plot. The biggest challenge during implementation was efficiency. When the size of data reach more than 10,000 data points, the waiting time for users becomes a major stumbling block and negatively affects the user experience. If the data is more than 100,000 data points, the browser will stop responding and the program will crash. To solve this problem, I tried two approaches: 1. Moving expensive calculation pre-computation onto the server side, and 2. changing the resolution of violin plots such that the computation can happen in logarithmic time.

The first approach was not successful. In order to migrate the calculation server side, multiple parameters need to be carried over to the server as well as scaling, position, and coordinate information. This information was not available until users provided a dataset and specific configuration information. There was no way the server could know this information beforehand so the pre-computation was not possible in this scenario.

The second solution was to change the resolution of the violin plot. In my implementation, the violin plot was generated from histograms. After the histogram was generated, a smoothing function was applied to smooth the edges of the histogram. The most time-consuming process was the calculation of histograms: The algorithm iterates through all the data points for each axis, then iterates through all the bins inside the histogram to smooth the curve. The solution here was to reduce the bin size of the histogram so that there would be less computation through the iterations.

In the implementation of the histogram, I used the D3 library [44]. One of the function parameters for the histogram is the function that determines the optimal size of the bin. Several algorithms are available in D3 to determine the size of bins that a certain histogram should have, namely Sturges', Scott's, and FreedmanDiaconis' formulas [45]. The purpose here is to reduce the computation time by reducing the number of bins in the histogram. Since Sturges' rule returns the number of bins k

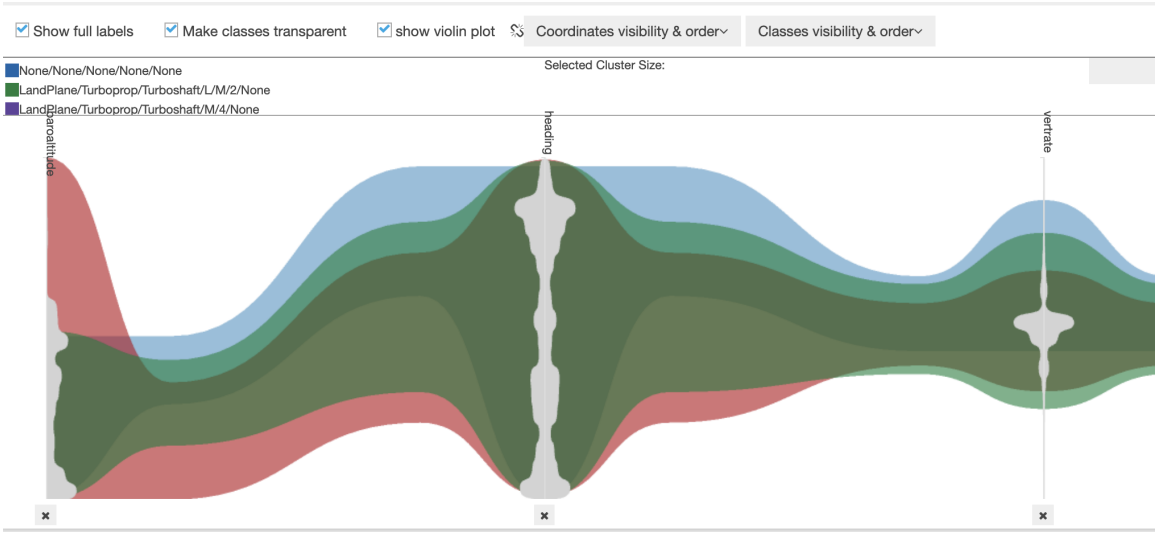


Figure 4.8: Violin plot showing data distribution on each axis

where

$$k = \left\lceil \log_2 n \right\rceil + 1, \text{n} = \text{total number of observations} \tag{4.2}$$

In this case, the resolution of the violin plot will be low compared to the other two methods. The main purpose of having a violin plot is to give users a general idea of where the majority of data is located on each axis. Details of such information are not important since users will have other interactions with the parallel coordinates to explore the data set (see Figure 4.8). The performance was improved significantly using Sturges' method.

4.3 User Interactions

Interactions assist with the data exploration process by helping users navigate a large amount of data. Standard user interactions with parallel coordinates include brushing [19] and reordering axes [46]. In bundled parallel coordinates, we also implemented several user interaction techniques.

4.3.1 Brushing and Selecting on Clusters

When users hover on a cluster, detailed information including cluster size, cluster minimum and maximum will appear on a text panel on the top of the visualization. The number of clusters can be changed by scrolling the mouse wheel or sliding on

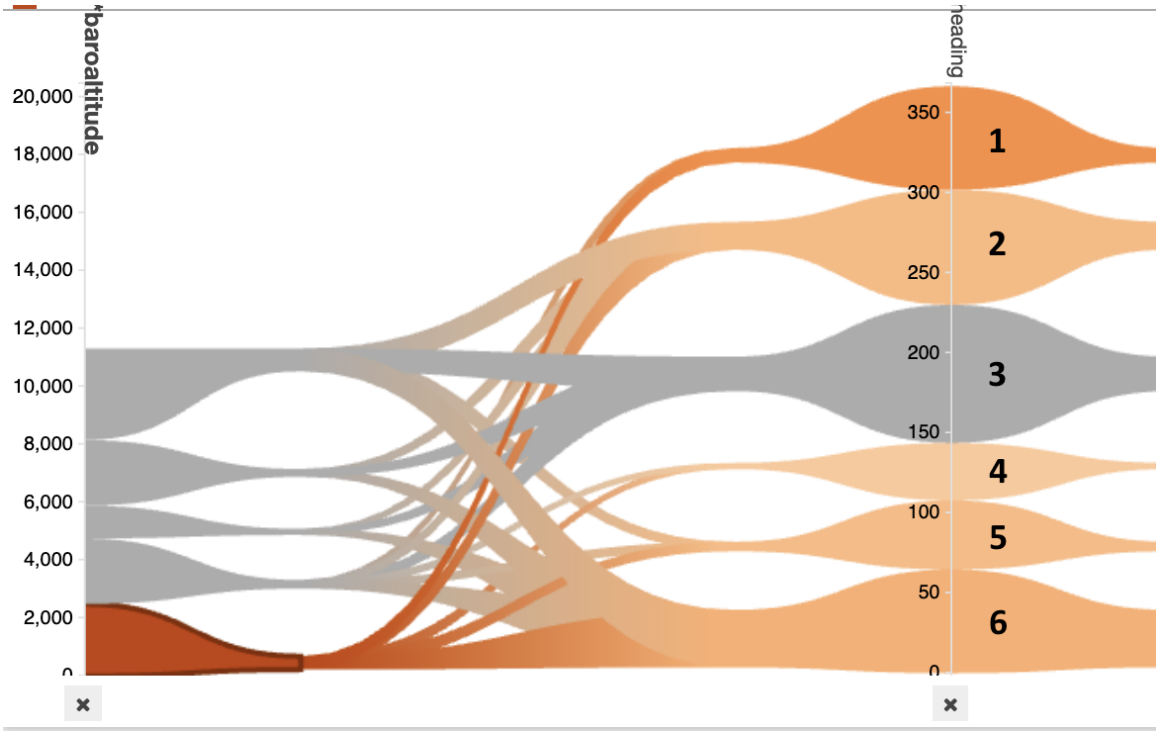
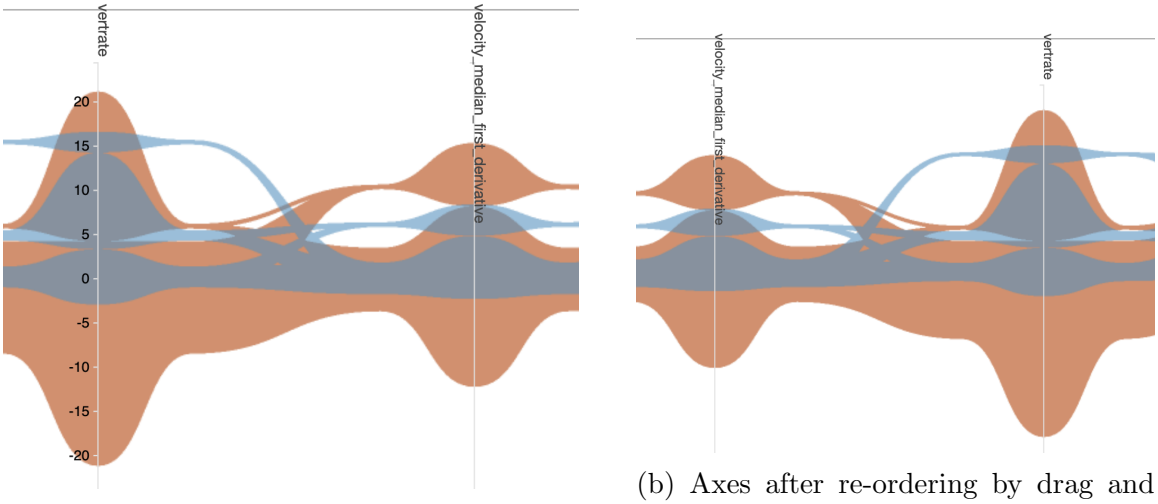


Figure 4.9: Data relationships between adjacent axes. When the bottom cluster on the left axis is selected, clusters on the right axis will have different colours to indicate relationships. For example, cluster 3 has no data value from the selected cluster; cluster 1 has the most data in this dimension from the selected cluster; cluster 2, 5, and 6 all have similar colours, and users can expand the cluster size further to investigate the relationships.

the touch pad when hovering over a cluster. The maximum number of clusters was set to 8 for the kernel density clustering method. The initial cluster can be further expanded into 8 sub-clusters to show a more detailed view. When we have a detailed view between adjacent axes, users can click on one of the sub-clusters to see the data range relationship of that cluster with the adjacent cluster (see Figure 4.9). When users select a cluster on one axis, the cluster color on the other axis will change based on how much data from the selected cluster dimension falls into other clusters in another dimension. If no data is found on the other dimension, the cluster color will be grey, otherwise, the color luminance will be faded 50 percent each time according to how much data is presented in different clusters.



(a) Axes before re-ordering

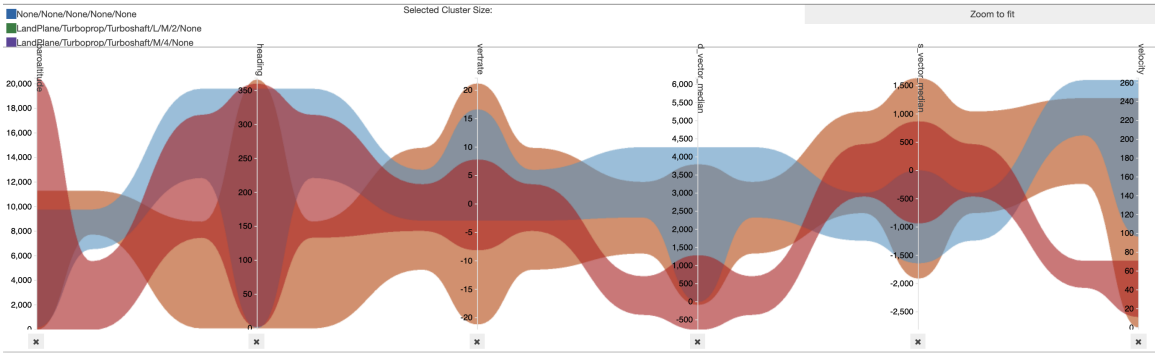
(b) Axes after re-ordering by drag and drop

Figure 4.10: Manual re-ordering of axes in bundled parallel coordinates

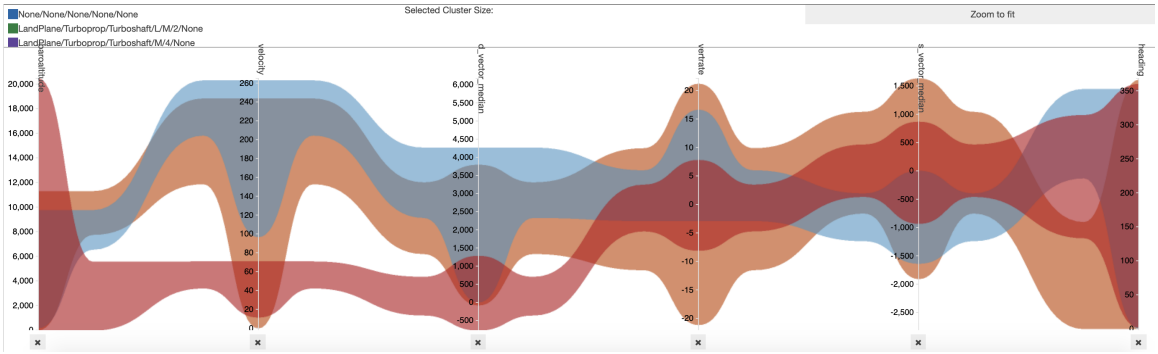
4.3.2 Re-ordering of Axes

Axis reordering is a typical interaction technique for parallel coordinates. The typical layout of the parallel coordinates only allows the discovery of relationships between adjacent axes. Re-ordering of axes by drag and drop can provide users a more insightful idea about the data through different combinations of axis orders. Our bundled parallel coordinates also supports re-ordering of axes by drag and drop. Figure 4.10 depicts the bundled parallel coordinates before and after swapping axes, which was done by drag and drop.

Another feature is the automatic re-ordering of axes. In our implementation of bundled parallel coordinates, users go through a configuration phase before they can see the visualization. We provided an option as a toggle box in the configuration to allow users to optimize the initial axes order of bundled parallel coordinates automatically. By default, this option is not selected, and the order of axes in the visualization is the same as what was set on the configuration page (see Figure 4.2). We implemented an algorithm by Lu et al. [46] which orders axes based on the similarity of each dimension and contribution of the dimension to the whole dataset. Singular value decomposition (SVD) [47] was used to calculate the contribution of each dimension to the whole dataset. In parallel coordinates, the first and last dimensions often attract more attention when presented to users at first view [46]. In this case, the algorithm first orders the left most axis as the most significant according to its contribution rank computed by SVD. The rest of the axes are ordered by their



(a) Axes before auto re-ordering



(b) Axes after auto re-ordering

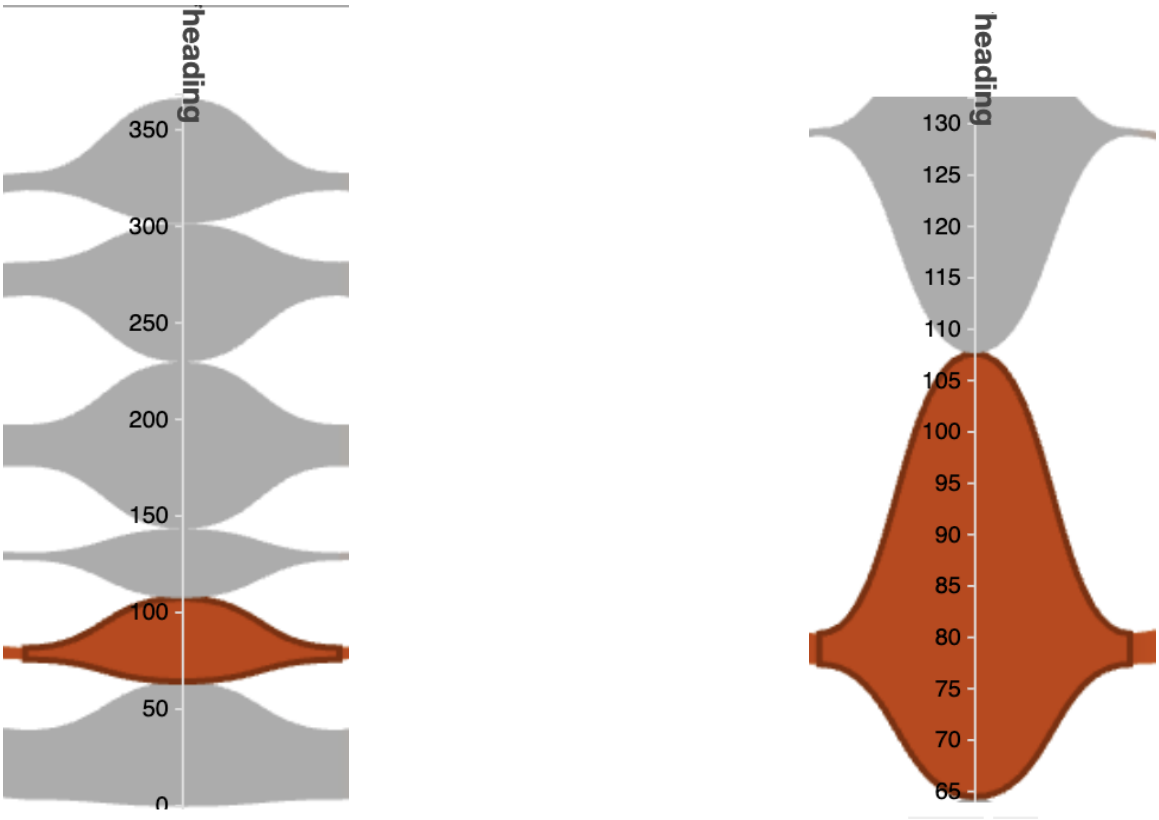
Figure 4.11: Automatically re-ordering of axes in bundled parallel coordinates

similarity based on a non-linear correlation coefficient. The Spearman's algorithm [48] was used to measure the similarity between axes. Those axes with the highest similarity will be placed adjacent to each other (see Figure 4.11).

The optimization of the initial order of axes was an attempt to help users find important and interesting patterns in the visualization more efficiently. The ordering of dimensions has a great impact on how we perceive data relationships [49]. Some may come up with a different conclusion about the data structures if the ordering of dimensions is different. It is important to take this into consideration when designing visualizations for multidimensional data.

4.3.3 Zooming

Zooming is a typical user interaction in visualization design. When it allows users to see a detailed view of a specific area, this is called geometric zooming. Another type of zooming technique, namely semantic zooming, allows users to see another type of data structure when zooming in and out to show details on demand. An example



(a) Cluster without geometric zooming

(b) Cluster with geometric zooming

Figure 4.12: Geometric zooming on clusters in bundled parallel coordinates

of an application that uses semantic zooming for ontology graph visualization can be found in Wiens et al.'s work [50]. In bundled parallel coordinates, the need for zooming is no more than the ability to see the scale of axes clearly, especially when the number of clusters is large. We do not need to show another data structure in this case, hence we use geometric zooming to show a more detailed range of values on the axis.

Users can zoom in and out on a specific cluster by scrolling the mouse wheel while the control key is pressed. Figure 4.12 shows a comparison with and without zoom on the same cluster. With geometric zooming, we can see the complete range of values within the cluster thus providing us a better understanding of the selected data.

Interaction techniques play an important role in visualization. Brushing, zooming, and re-ordering of axes in bundled parallel coordinates provide users a more flexible way to explore multidimensional data. In the next chapter, I present a study design

of my evaluation for bundled parallel coordinates as future work.

Chapter 5

Evaluating Parallel Coordinates: A Review of Studies and a New Study Design

In this chapter, I review existing research of parallel coordinates that presented an evaluation as a part of their study, and propose a new study design of an evaluation for bundled parallel coordinates as future work.

5.1 Evaluations of Parallel Coordinates

Recent studies have evaluated the efficiency of parallel coordinates visualizations by comparing them with other visualization techniques or with its variants. Lanzenberger et al. [51] evaluated parallel coordinates and stardinates, a novel visualization technique for highly structured data. The stardinates are a hybrid visualization technique combining glyph-based features with geometric representations. The axes are arranged in a circle shape similar to pie charts. Two data sets were used and 22 participants took part in the study. The results revealed that the stardinates visualization was more appropriate for structured data in analyzing details. The parallel coordinates technique, however, can provide an overview of the data more quickly. Kuang et al. [52] compared parallel coordinates with scatterplots in terms of value retrieval. The main conclusion was that the parallel coordinates technique produced more errors in value retrieval tasks when the dimensionality and density of data increased. In comparison, scatterplots produced less errors when these two properties

changed.

Pillat et al. [53] evaluated parallel coordinates and RadViz in a usability study. Five participants were asked to answer specific questions regarding relationships in the data set. Their results showed the parallel coordinates visualization was more effective for identifying outliers and relationships in subsets of data. RadViz, however, has the advantage of identifying clusters and providing a better overview of the data structure.

Identifying and analyzing data correlation is an important goal for visualization research. Several studies have looked into this problem and evaluated the parallel coordinates technique. Li et al. [54] compared parallel coordinates with scatterplots. Twenty-five participants were asked to judge the strength of linear correlation in both visualizations. Five semantic levels were used, from strong negative to strong positive. The study concluded that scatter plots were better at this task and users could identify twice as many correlation levels than parallel coordinates. Another interesting finding from their study is that users generally overestimate the strength of negative linear correlation using a parallel coordinates visualization.

Harrision et al. [55] mapped human perceptions of correlation using Weber's law in nine visualizations including parallel coordinates. The results indicated that scatter plots perform better in depicting positively correlated data than parallel coordinates. Furthermore, parallel coordinates plots can depict negatively correlated data better than positively correlated data.

Palmas et al. [23] designed an edge-bundling technique for parallel coordinates. In their design, data points were rendered as clusters instead of traditional polylines. They compared this technique with the traditional 2D parallel coordinates. The tasks included judging the strength of correlation and tracing subsets of data from one variable to another. They concluded that the edge-bundled parallel coordinates were superior in both tasks.

Forsell and Johansson [56] went beyond linear correlations and investigated non-linear correlations in parallel coordinates. Three sinusoidal relations along with two linear relations were used to investigate the performance of standard 2D parallel coordinates compared with 3D multi-relational parallel coordinates. Thirty participants were asked to identify one and all five data relationships and also to identify the relationship that was missing. The results indicated that 3D multi-relational parallel coordinates was faster when users manually explored in a complex multivariate dataset. However, the performance between the two visualizations was negligible with

simple tasks.

5.2 Design of an Evaluation Study

I propose a study to investigate the perception of non-linear data relationships between regular parallel coordinates and bundled parallel coordinates. Non-linear data relationships are a common type of data pattern that exists in everyday applications. For example, the radius of a sphere has a monotonic non-linear relationship to the volume of the same sphere; the value of your vehicle and the amount of time you owned it also has a non-linear relationship. Empirical studies on parallel coordinates often focus on linear data relationships in their evaluations [28, 5, 20, 57, 54]. There is a need to go beyond linear data relationships and explore non-linear data relationships in parallel coordinates to better utilize this visualization.

In this study design, participants would have to assign a data relationship to each axis pair in both versions of parallel coordinates by using visual inspections and user interactions. To measure the performance of each visualization, we would use three dependent variables: (1) task completion time, (2) number of errors, and (3) the confidence of the participants. These measurements are commonly used in visualization research to capture the performance of the visualization tools [58, 59].

Visualization Design and Implementation. Both versions of parallel coordinates were developed in Lodestone. The standard 2D parallel coordinates is the basic parallel coordinates that uses polylines to connect data between each axis (see the 2D parallel coordinates in Forsell and Johansson's work [56] for an example). The edge-bundled parallel coordinates were introduced in section 4 of this report. Both versions of parallel coordinates support axis-reordering by drag and drop, and brushing by clicking on a predefined interval or cluster. The default cluster size for bundled parallel coordinates was set to 8.

Hypotheses. Our study could consider the following two hypotheses:

H1. Edge-bundled parallel coordinates perform better than the standard 2D parallel coordinates for the specific tasks in this study. In particular, we expect the edge-bundled parallel coordinates to be (**1**) faster for users to identify different data relationships, (**2**) fewer errors would be produced by the users in the tasks, and (**3**) the users would be more confident of their answers. I assume that users would be able to trace and follow bundled lines more easily than regular polylines since the amount of information in the visualization is less.

H2. Interactions would help boost the performance of both visualizations. In particular, we expect with interactions, users will (**1**) spend more time on each task due to the higher interactivity, (**2**) fewer errors would be produced by the users than with no interaction, and (**3**) the users would be more confident of their answers. I assume interactions would boost the user performance in visualization tasks. Similar results were obtained by Perin et al. [60] when testing interactive visualizations.

5.2.1 Datasets

The creation of the dataset composes an important aspect of this study design. We constructed a set of synthetic data to support this goal. To eliminate interference in the data, like noise, we needed to have full control of different data relationships by using synthetic data [61]. However, we compromised the value of using empirical data, and the correlations may not fully reflect reality. Based on an initial pilot study with 6 participants (5 male and 1 female in the age range of 22-26 from the computer science department, all with some background in data visualization). We selected the following five mathematical relationships: a negative linear correlation, a positive quadratic correlation, a negative quadratic correlation, a sinusoidal correlation with one period, and a negative cubic correlation. We omitted positive linear correlations and positive cubic correlations due to their simplicity. Participants from the pilot could identify these relationships with little effort. We included one linear correlation to make the experiment more complete. The data set consists of five data correlations and one noise correlation together as a six-dimensional data set. Each dimension (correlation) has 3,000 sample points. According to our pilot study, a 3,000 data points sample was a decent amount for both versions of parallel coordinates. If the sample size is small, there is no need to use the bundling technique. On the other hand, if the sample size exceeds this amount, the 2D parallel coordinates would become too cluttered to see any information. The five correlations were constructed as follows: 1) y $= -ax +$ e 2) y1 $= ax^2 +$ e 3) $-ax^2 +$ e 4) y $= sin(\pi x) + e$ 5) $-ax^3 + e$, where e is a small random noise and a is a constant value. All variables have been normalized to the range of [-1,1].

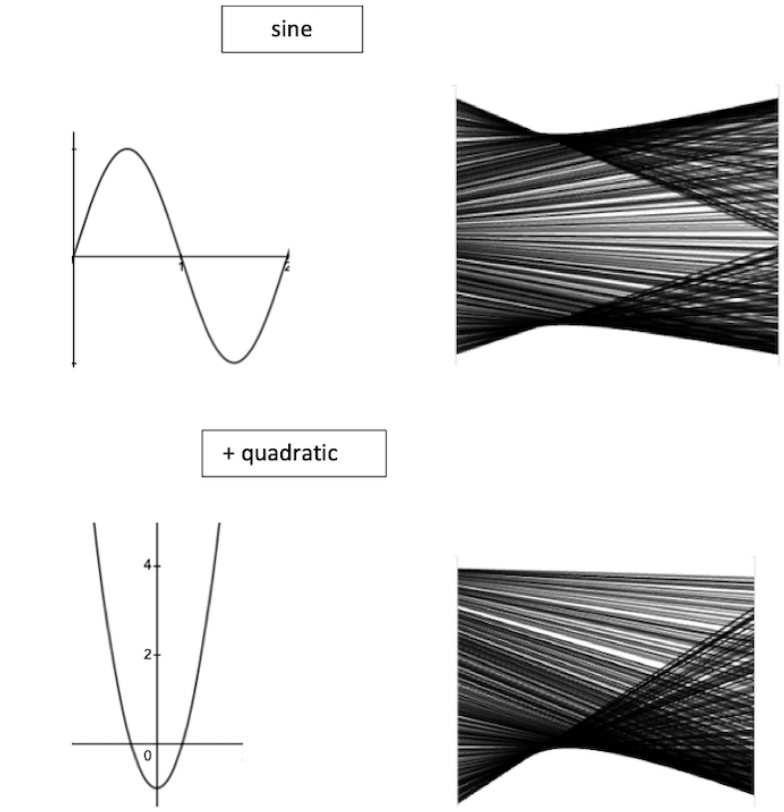
5.2.2 Tasks and Procedure

The experiment itself is a simple judgment task. We doubled each correlation to make a total of 10 correlations. We presented all variables at once in a multidimen-

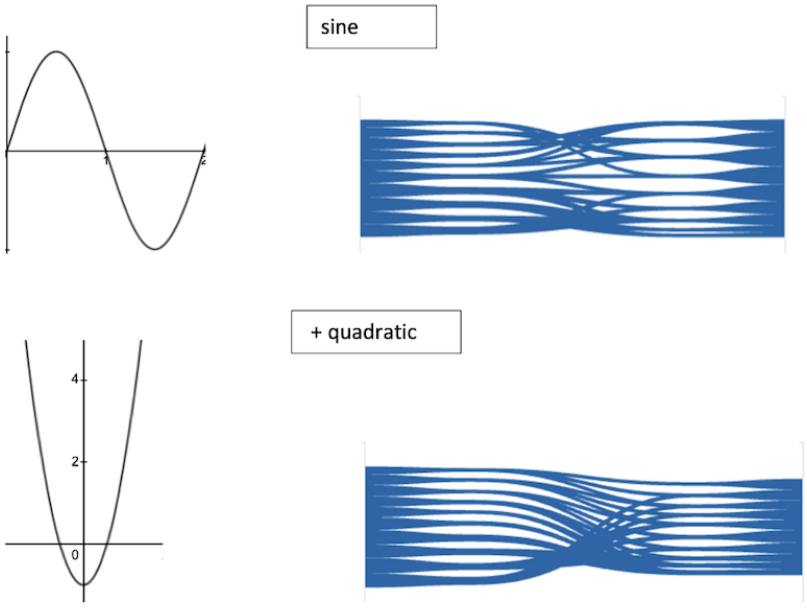
sional view. We chose this way to represent data because it is the typical way that people would see a parallel coordinates visualization. In each trial, we would ask the participant to identify a relationship between a given axes pair (e.g., X and Y). The investigator will re-order the axis to show the next task pattern. The participant then needed to choose from a multiple-choice list to give an answer. For each question, they would be given two attempts. During the first attempt, the participant cannot use any interactions. They will provide an answer by just looking at the pattern. A quiz to gain participants' initial understanding of non-linear data relationships would be given, followed by a tutorial provided at the beginning of the experiment to help participants become familiar with all the data relationships (see Figure 5.1). This information sheet will be taken away when the evaluation starts.

During the second attempt, the participants are allowed to use interactions, which might include brushing, selecting, and reordering axes. Furthermore, we would ask each participant after each attempt how confident they were with their answer, on a five-point scale from -2 to +2, with +2 meaning they are almost 100% confident about their answer, and -2 meaning they are very uncertain about their answer.

The evaluation with actual participants did not happen due to a global pandemic and the shutting down of university laboratories. In the next chapter, I discuss the limitations of the bundled parallel coordinates tool and the design of this evaluation.



(a) Sample non-linear data relationships in regular PCs



(b) Sample non-linear data relationships in bundled PCs

Figure 5.1: Sample non-linear data relationships in parallel coordinates

Chapter 6

Discussion

Through the pilot study, the preliminary results showed that users were able to identify non-linear data relationships faster and with higher accuracy using standard parallel coordinates than bundled parallel coordinates. Some participants found that "It's easier to identify patterns in the regular parallel coordinates because the lines are straight and less overlapping." Others commented, "The patterns in bundled parallel coordinates are less distinguishable, some patterns like sine and quadratic looked very similar." Due to the layout of bundled parallel coordinates, data are clustered as bundled strips instead of polylines. The bent shape of the lines created more visual overhead to users at first glance. However, in both versions of parallel coordinates, the pilot participants agreed that interactions helped them identify patterns more accurately than without them. Brushing through the axis was a common interaction that participants used in their sessions.

The pilot study demonstrated that with the given dataset and non-linear data relationships, regular parallel coordinates can better support users identifying such relationships than bundled parallel coordinates. Data relationships in real-world applications may not be as perfect as the dataset in this experiment, so the empirical performance of these parallel coordinates still needs to be investigated.

As the pilot study had only six participants and all had some visualization background, it was still not comprehensive enough and results may only be relevant to populations with similar background. A larger number of participants would be ideal but recruiting was not easy. There may be the concern of being judged or scored from the evaluation tasks that may hinder people from participating. It is important to keep in mind the trade-offs between user performance and experience when designing evaluation [62] to provide a smooth and objective evaluation experience. Some other

challenges met in this project include:

- The ingestion of large CSV files to Lodestone. As mentioned in Chapter 2, one of the challenges of multidimensional data visualization is the ability to process a large amount of data. The bundled parallel coordinates visualization was built using Lodestone. While the visualization has the capability to process a large amount of data by streaming the input and clustering data points, Lodestone still reads the whole file into the browser's memory at the data preparation stage. If large files were uploaded to the browser, it would take a long time to process, and potentially freeze the application. It was for this reason that only small to medium sized files were tested in the visualization.

- Time constraints for enhancing the visualization and conducting an evaluation. In this project, we received lots of valuable feedback from our industrial partner Thales and other collaborators. For instance, in the evaluation design, we had the opportunity to explore eye-tracking devices with Tobii eye trackers. Due to the steep learning curve of the eye-tracking data analysis tool and limited time constraints, we presented the simpler evaluation design we present in this report. It would be interesting to use eye-tracking devices to understand users focus point when solving certain tasks in the future.

In terms of the limitations of the tool design, the first limitation is that the tool does not support categorical values in non-numeric format. The axes are set to only display numeric values. Second, the tool does not support the identification of data outliers very well. Outliers have to be extreme to be able to stand out in both versions of parallel coordinates. Another limitation is the lack of ability to interact with other visualizations in Lodestone. It would be useful to have the ability to link multiple visualizations like a dashboard to provide better insights. The design of the evaluation study uses synthetic datasets with manually created data relationships. However, in order to better assess the perception of non-linear data relationships in real-world applications, empirical data should be considered in the evaluation. The dependent variables to be measured in this design are also limited. More advanced measurements and techniques such as eye-tracking data could be added to further support some hypotheses.

Chapter 7

Conclusion

In this project, I designed an interactive edge-bundled parallel coordinates tool and implemented it as a web application in Lodestone. This visualization is a solution to address visual clutter in standard parallel coordinates when the goal is to visualize a large amount of data. This visualization supports two types of clustering methods: density-based clustering and proximity-based clustering. Various types of interaction techniques are implemented to support users' direct explorations of multidimensional data. I conducted a preliminary pilot study and the results indicated that it might be possible that the regular parallel coordinates helped users better identify non-linear data relationships than bundled parallel coordinates, and interactions were helpful in those tasks.

In terms of future work, the bundled parallel coordinates need to better support the identification of outliers. The ability to visualize categorical values is also desired. Finally, the evaluation should look into empirical data to better evaluate the performance and usability of bundled parallel coordinates.

Bibliography

[1] Niklas Elmqvist, Pierre Dragicevic, and Jean-Daniel Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE transactions on Visualization and Computer Graphics*, 14(6):1539–1148, 2008.

[2] Harald Sanftmann and Daniel Weiskopf. Illuminated 3d scatterplots. In *Computer Graphics Forum*, volume 28, pages 751–758. Wiley Online Library, 2009.

[3] Michael P Schroeder, Abel Gonzalez-Perez, and Nuria Lopez-Bigas. Visualizing multidimensional cancer genomics data. *Genome medicine*, 5(1):9, 2013.

[4] Ying-Huey Fua, M.O. Ward, and E.A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. *Proceedings Visualization '99 (Cat. No.99CB37067)*.

[5] Hong Zhou, Xiaoru Yuan, Huamin Qu, Weiwei Cui, and Baoquan Chen. Visual clustering in parallel coordinates. *Computer Graphics Forum*, 27(3):1047–1054, May 2008.

[6] Rene Rosenbaum, Jian Zhi, and Bernd Hamann. Progressive parallel coordinates. *2012 IEEE Pacific Visualization Symposium*, Feb 2012.

[7] Geoffrey Ellis and Alan Dix. Enabling automatic clutter reduction in parallel coordinate plots. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):717–724, Sep 2006.

[8] Peihong Guo, He Xiao, Zuchao Wang, and Xiaoru Yuan. Interactive local clustering operations for high dimensional data in parallel coordinates. *2010 IEEE Pacific Visualization Symposium (PacificVis)*, Mar 2010.

[9] Halldór Janetzko, Manuel Stein, Dominik Sacha, and Tobias Schreck. Enhancing Parallel Coordinates: Statistical Visualizations for Analyzing Soccer Data. *Electronic Imaging*, 2016(1):1–8, 2017.

[10] D.A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, 2002.

[11] Gintautas Dzemyda, Olga Kurasova, and J Zilinskas. Multidimensional data visualization. *Methods and applications series: Springer optimization and its applications*, 75:122, 2013.

[12] Pak Chung Wong and R Daniel Bergeron. 30 years of multidimensional multivariate visualization. *Scientific Visualization*, 2:3–33, 1994.

[13] Julian Heinrich and Daniel Weiskopf. State of the Art of Parallel Coordinates. In M. Sbert and L. Szirmay-Kalos, editors, *Eurographics 2013 - State of the Art Reports*. The Eurographics Association, 2013.

[14] Alfred Inselberg. The plane with parallel coordinates. *The visual computer*, 1(2):69–91, 1985.

[15] Jimmy Johansson, Patric Ljung, Mikael Jern, and Matthew Cooper. Revealing structure in visualizations of dense 2d and 3d parallel coordinates. *Information Visualization*, 5(2):125–136, Jun 2006.

[16] A.O. Artero, M.C.F. de Oliveira, and H. Levkowitz. Enhanced high dimensional data visualization through dimension reduction and attribute arrangement. *Tenth International Conference on Information Visualisation (IV'06)*.

[17] G. Ellis and A. Dix. Density control through random sampling: an architectural perspective. *Proceedings Sixth International Conference on Information Visualisation*.

[18] Ji Soo Yi, Youn ah Kang, and John Stasko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1224–1231, Nov 2007.

[19] H. Hauser, F. Ledermann, and H. Doleisch. Angular brushing of extended parallel coordinates. *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002.*

[20] Harri Siirtola. Direct manipulation of parallel coordinates. *CHI '00 extended abstracts on Human factors in computing systems - CHI '00*, 2000.

[21] E. Fanea, S. Carpendale, and T. Isenberg. An interactive 3d integration of parallel coordinates and star glyphs. *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*

[22] Wei Peng, M.O. Ward, and E.A. Rundensteiner. Clutter reduction in multidimensional data visualization using dimension reordering. *IEEE Symposium on Information Visualization.*

[23] Gregorio Palmas, Myroslav Bachynskyi, Antti Oulasvirta, Hans Peter Seidel, and Tino Weinkauf. An edge-bundling layout for interactive parallel coordinates. In *IEEE Pacific Visualization Symposium*, 2014.

[24] Joris Sansen, Gaëlle Richer, Timothée Jourde, Frédéric Lalanne, David Auber, and Romain Bourqui. Visual Exploration of Large Multidimensional Data Using Parallel Coordinates on Big Data Infrastructure. *Informatics*, 2017.

[25] Rajeev Agrawal, Anirudh Kadadi, Xiangfeng Dai, and Frederic Andres. Challenges and opportunities with big data visualization. In *Proceedings of the 7th International Conference on Management of computational and collective intElligence in Digital EcoSystems*, pages 169–173, 2015.

[26] Harri Siirtola, Tuuli Laivo, Tomi Heimonen, and Kari-Jouko Räihä. Visual perception of parallel coordinate visualizations. In *2009 13th International Conference Information Visualisation*, pages 3–9. IEEE, 2009.

[27] K T Mcdonnell and K Mueller. Illustrative parallel coordinates. *Computer Graphics Forum*, 27(3):1031–1038, 2008.

[28] Julian Heinrich, Yuan Luo, Arthur E Kirkpatrick, Hao Zhang, and Daniel Weiskopf. Evaluation of a bundling technique for parallel coordinates. *arXiv preprint arXiv:1109.6073*, 2011.

[29] Jimmy Johansson and Camilla Forsell. Evaluation of parallel coordinates: Overview, categorization and guidelines for future research. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):579–588, Jan 2016.

[30] Michael Friendly and Daniel Denis. The early origins and development of the scatterplot. *Journal of the History of the Behavioral Sciences*, 41(2):103–130, 2005.

[31] Robert Kosara, Gerald N Sahling, and Helwig Hauser. Linking scientific and information visualization with interactive 3d scatterplots. 2004.

[32] Mihael Ankerst, Daniel A Keim, and Hans-Peter Kriegel. Circle segments: A technique for visually exploring large multidimensional data sets. In *Visualization*, 1996.

[33] J. Johansson, P. Ljung, M. Jern, and M. Cooper. Revealing structure within clustered parallel coordinates displays. *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*

[34] Catherine Plaisant. The challenge of information visualization evaluation. *Proceedings of the working conference on Advanced visual interfaces - AVI '04*, 2004.

[35] Irina Yatskiv and Anastasija Hismutova. Clutter reduction in parallel coordinates using aesthetic criteria. *Frontiers in Artificial Intelligence and Applications*, 312:81–100, 01 2019.

[36] Almir Olivette Artero, Maria Cristina Ferreira de Oliveira, and Haim Levkowitz. Uncovering clusters in crowded parallel coordinates visualizations. In *Proceedings of the IEEE Symposium on Information Visualization*, INFOVIS '04, pages 81–88, Washington, DC, USA, 2004. IEEE Computer Society.

[37] Alan Dix and Geoff Ellis. by chanceenhancing interaction with large data sets through statistical sampling. *Proceedings of the Working Conference on Advanced Visual Interfaces - AVI '02*, 2002.

[38] Evanthia Dimara and Charles Perin. What is interaction for data visualization? *IEEE transactions on visualization and computer graphics*, 26(1):119–129, 2019.

[39] Ben Shneiderman. The future of interactive systems and the emergence of direct manipulation. *Behaviour & Information Technology*, 1(3):237–256, 1982.

[40] Danny Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on visualization and computer graphics*, 12(5):741–748, 2006.

[41] Weiwei Cui, Hong Zhou, Huamin Qu, Pak Chung Wong, and Xiaoming Li. Geometry-based edge clustering for graph visualization. *IEEE transactions on visualization and computer graphics*, 14(6):1277–1284, 2008.

[42] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[43] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

[44] M. Bostock, V. Ogievetsky, and J. Heer. D data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, Dec 2011.

[45] MP Wand. Data-based choice of histogram bin width. *The American Statistician*, 51(1):59–64, 1997.

[46] Liang Fu Lu, Mao Lin Huang, and Jinson Zhang. Two axes re-ordering methods in parallel coordinates plots. *Journal of Visual Languages & Computing*, 33:3–12, 2016.

[47] GH Golub and CFV Loan. Matrix computations 1996 3rd baltimore. *Md, USA Johns Hopkins University Google Scholar*.

[48] Charles Spearman. The proof and measurement of association between two things. 1961.

[49] Mihael Ankerst, Stefan Berchtold, and Daniel A Keim. Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In *Proceedings IEEE Symposium on Information Visualization (Cat. No. 98TB100258)*, pages 52–60. IEEE, 1998.

[50] Vitalis Wiens, Steffen Lohmann, and Sören Auer. Semantic zooming for ontology graph visualizations. In *Proceedings of the Knowledge Capture Conference*, pages 1–8, 2017.

[51] Monika Lanzenberger, Silvia Miksch, and Margit Pohl. Exploring highly structured data: a comparative study of stardinates and parallel coordinates. In *Ninth International Conference on Information Visualisation (IV'05)*, pages 312–320. IEEE, 2005.

[52] Xiaole Kuang, Haimo Zhang, Shengdong Zhao, and Michael J McGuffin. Tracing tuples across dimensions: A comparison of scatterplots and parallel coordinate plots. In *Computer Graphics Forum*, volume 31, pages 1365–1374. Wiley Online Library, 2012.

[53] Raquel M Pillat, Eliane RA Valiati, and Carla MDS Freitas. Experimental study on evaluation of multidimensional information visualization techniques. In *Proceedings of the 2005 Latin American conference on Human-computer interaction*, pages 20–30. ACM, 2005.

[54] Jing Li, Jean-Bernard Martens, and Jarke J Van Wijk. Judging correlation from scatterplots and parallel coordinate plots. *Information Visualization*, 9(1):13–30, 2010.

[55] Lane Harrison, Fumeng Yang, Steven Franconeri, and Remco Chang. Ranking visualizations of correlation using weber's law. *IEEE transactions on visualization and computer graphics*, 20(12):1943–1952, 2014.

[56] Camilla Forsell and Jimmy Johansson. Task-based evaluation of multirelational 3d and standard 2d parallel coordinates. In *Visualization and Data Analysis 2007*, volume 6495, page 64950C. International Society for Optics and Photonics, 2007.

[57] Mats Lind, Jimmy Johansson, and Matthew Cooper. Many-to-many relational parallel coordinates displays. In *2009 13th International Conference Information Visualisation*, pages 25–31. IEEE, 2009.

[58] Alfie Abdul-Rahman, Min Chen, and David H Laidlaw. A survey of variables used in empirical studies for visualization. In *Foundations of Data Visualization*, pages 161–179. Springer, 2020.

[59] Sheelagh Carpendale. Evaluating information visualizations. In *Information visualization*, pages 19–45. Springer, 2008.

[60] Charles Perin, Frédéric Vernier, and Jean-Daniel Fekete. Interactive horizon graphs: Improving the compact visualization of multiple time series. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 3217–3226, 2013.

[61] Daniel A Keirn, R Daniel Bergeron, Ronald M Pickett, H Levkowitz, et al. Test data sets for evaluating data visualization techniques. In *Perceptual issues in Visualization*, pages 9–22. Springer, 1995.

[62] John M Carroll. Human-computer interaction: psychology as a science of design. *Annual review of psychology*, 48(1):61–83, 1997.