TU WIEN Informatics

Einfluss des Seitenverhältnisses auf Parallele Koordinaten

BACHELORARBEIT

zur Erlangung des akademischen Grades

Bachelor of Science

im Rahmen des Studiums

Medieninformatik und Visual Computing

eingereicht von

Mirjam Kronsteiner
Matrikelnummer 11808233

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller
Mitwirkung: External Lecturer Dipl.-Ing.in Dr.in techn. Johanna Schmidt

Wien, 10. Oktober 2024

Mirjam Kronsteiner

Mirjam Kronsteiner | Eduard Gröller

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Informatics

Influence of Aspect Ratio on Parallel Coordinates

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Media Informatics and Visual Computing

by

Mirjam Kronsteiner
Registration Number 11808233

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller
Assistance: External Lecturer Dipl.-Ing.in Dr.in techn. Johanna Schmidt

Vienna, October 10, 2024

Mirjam Kronsteiner Eduard Gröller

Erklärung zur Verfassung der Arbeit

Mirjam Kronsteiner

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.
Ich erkläre weiters, dass ich mich generativer KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Im Anhang „Übersicht verwendeter Hilfsmittel“ habe ich alle generativen KI-Tools gelistet, die verwendet wurden, und angegeben, wo und wie sie verwendet wurden. Für Textpassagen, die ohne substantielle Änderungen übernommen wurden, haben ich jeweils die von mir formulierten Eingaben (Prompts) und die verwendete IT- Anwendung mit ihrem Produktnamen und Versionsnummer/Datum angegeben.

Wien, 10. Oktober 2024

Mirjam Kronsteiner

Mirjam Kronsteiner

Danksagung

Ich möchte mich herzlich bedanken bei Dipl.-Ing.in Dr.in techn. Johanna Schmidt für die ausführliche Betreuung bei der Erarbeitung meines Themas sowie der kontinuierlichen Unterstützung bei der Umsetzung der Theorie- und Praxisarbeit.

Weiters danke ich meiner Familie sowie meinen Freund*innen und Kolleg*innen, die mir während dieser Arbeit mit wertvollen Beiträgen zur Seite standen, und die mir während meines gesamten Studiums viel Kraft und Unterstützung gegeben haben.

Acknowledgements

I sincerely thank Dipl.-Ing. Dr.in techn. Johanna Schmidt for her extensive supervision during the conceptualization of my topic and her continuous support in the implementation of the theoretical and practical sections.

I also thank my family, friends and colleagues who provided me with valuable input during this thesis and were a big source of strength and support throughout my studies.

Kurzfassung

Parallele Koordinaten sind eine außergewöhnliche Visualisierungsmethode, welche vielversprechende Einsatzmöglichkeiten für die Visualisierung von großen, vielfältigen multivariaten Datensätzen bietet. Häufig sollen solche Daten in web-basierten Visualisierungen oder Dashboards dargestellt werden. Im modernen Web gängige Konzepte wie Interaktivität, benutzerdefinierte Einstellungen, oder Responsive Design, wobei sich eine Webseite an jede beliebige Bildschirmauflösung anpassen können sollen, bewegen uns dazu, die Rolle des Seitenverhältnisses beim Gestalten von Visualisierungen zu berücksichtigen. Ein webbasiertes Werkzeug wurde implementiert und eine statistische Analyse von Winkelparametern in Parallele-Koordinaten-Diagrammen durchgeführt. Diese deutet auf einen signifikanten Einfluss des Seitenverhältnisses auf die Darstellung von Parallelen Koordinaten hin, und zeigt, dass Diagramme im Querformat robuster gegenüber Änderungen des Seitenverhältnisses sind als im Hochformat.

Abstract

Parallel coordinates are a unique visualization technique that presents promising opportunities for the visualization of large and diverse multivariate datasets. Applications such as web-based visualizations and dashboards are common use cases for this type of data. Prevalent concepts in the modern web are responsive design - the ability of a web page to fit any screen resolution - as well as interactivity and customizability, requiring us to consider the role of aspect ratio in the design of visual displays. We implemented a web-based tool and conducted a statistical analysis of angle parameters in parallel coordinates plots. Our results indicate a significant influence of aspect ratio on the display of parallel coordinates, and show that landscape orientations are more consistent across different aspect ratios than portrait orientations.

Contents

CHAPTER 1

Introduction

A prominent area of research in information and data visualization constitutes the display of multivariate, or multi-dimensional data. With the advancement of internet connectivity and storage capacities, more and more data is being collected and stored, resulting in large datasets with many distinct attributes. Trying to visualize this data requires innovative solutions, especially if we wish to convey more than two or three dimensions in one graph.

A well-known visualization technique for multivariate data is the parallel coordinates plot (PCP). Axes, each representing a single dimension (column) of the dataset, are placed in parallel, and their data ranges are scaled to be the same size. A data sample (row) is represented as a polyline that intersects each axis at a given point on the scale. An advantage of this type of plot is that it can be used to display a high number of dimensions. Unlike other multivariate visualization techniques, where a certain hierarchy or order of dimensions has to be assigned before drawing the plot, a PCP allows for flexible reordering of axes. On the other hand, a common criticism of parallel coordinates is that they are unintuitive. It will take first-time users a while until they are able to read the plot and to recognize certain patterns. This may discourage designers from utilizing them for applications where the goal is to clearly and quickly convey certain information. To address this, the original concept of PCP has been developed further by many researchers, introducing new display methods and features. For example, automatic axis ordering or bundling techniques aim to reduce visual clutter, allowing for previously hidden patterns and outlier data to be discovered. Interactive techniques such as brushing or highlighting let users visually explore datasets.

We specifically look at the use case of PCPs being included in a web-based visualization or dashboard, where the dimensions are dependent on the user's device screen size, or where the size of the plot can be changed interactively by the user. An unsuitable aspect ratio can impact the effectiveness of a visualization, or even lead to visual errors. While many dashboard design guidelines have been proposed, dealing with suggestions for plot selection and layouting, not many address the problem of variable aspect ratios. To understand this issue, we conducted a literature review in which we researched various visualization techniques that have been developed, and looked at different perspectives on the role of aspect ratio in visualization.

We then implemented an interactive parallel coordinate viewer. The core feature of our implementation is the ability to dynamically resize the plot. This tool can be used to interactively explore a dataset using a PCP and check how the plot behaves if the aspect ratio is changed. An example of the same plot in two different aspect ratios is shown in Figure 1.1. Additionally, using our application, we can systematically simulate what a plot of a given dataset would look like in several predefined aspect ratios and use this data for further analysis.

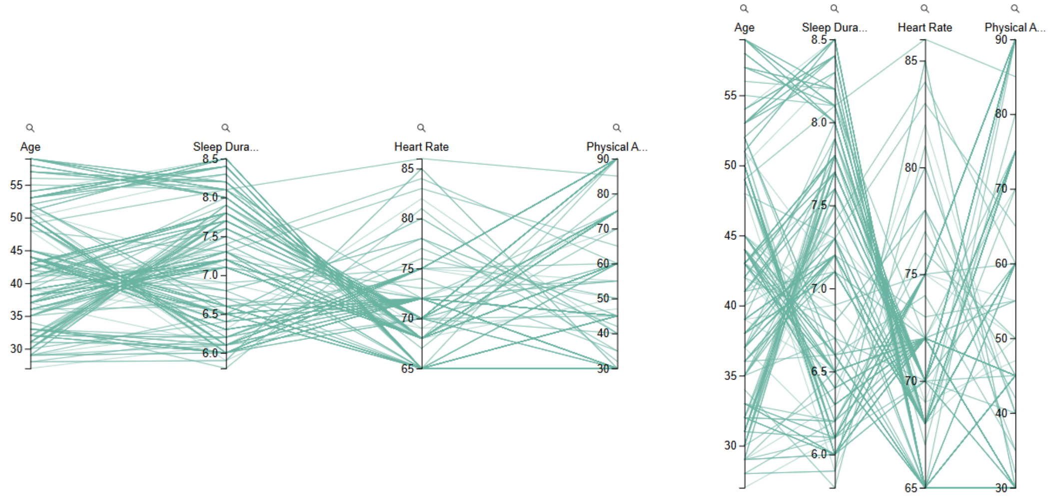


Figure 1.1: A subset of the “Sleep Health and Lifestyle” dataset [Tha23] visualized in our PCP application, using two different aspect ratios.

Based on the results of our statistical analysis, we formulated design guidelines for parallel coordinates, specifically targeted at designers of multiple-view dashboards that incorporate parallel coordinates.

CHAPTER 2

Literature review

We conducted a review of related work, starting with a general look at the history and motivations behind information visualization, and continuing with an overview of different multivariate visualization approaches with a focus on parallel coordinates. We discuss how different visualization techniques can be assessed and compared to each other. Finally, we look at some existing literature related to the topic of aspect ratio in visualization.

2.1 Overview of visualization and information visualization

Visualization allows us to comprehend large amounts of data, find properties within datasets that were not evident previously, and form hypotheses based on these observations. Data visualization can reveal data patterns and special characteristics both on a local and a global scale, and allow for quality control by exposing issues such as value errors, artifacts, or flaws of the data collection method [War04].

Referring to a *"graphical representation of data or concepts"* [War04] the term *data visualization* has only emerged in recent decades. Previously, data visualization was defined as *"constructing a visual image in the mind"* [War04]. Still, both of these definitions are important to take into account when studying computer-based visualization. When we are viewing a graphic, there is some information that only "lives" in our head, and some that is only explicit in the digital domain [SMM12].

Examples of data being presented in a structured way similar to what we today consider a chart, can be found as early as the middle ages. From the 18th century onwards, visualizations became much more sophisticated, being primarily used in scientific works, starting with time-series charts. These graphics were often used instead of tables if it was necessary to display a large number of variables and data points [Tuf01], and often doubled as data-transmission devices, which were made by hand and commonly contained

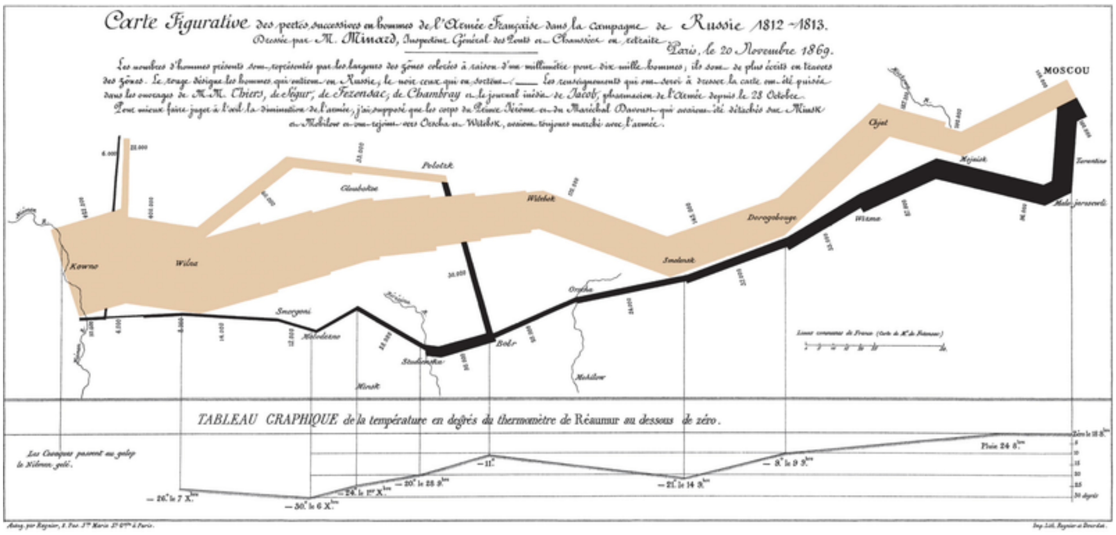


Figure 2.1: Charles Joseph Minard's visualization of Napoleon's invasion of Russia. Image taken from Raposo et al. [RTB20].

additional guiding elements such as grid lines, to ensure precision as well as to increase the accuracy of reading data from the diagram [Cle93]. Figure 2.1 shows a well-known example of a multivariate visualization from 1869 drawn by Charles Joseph Minard, displaying army movements and strength during Napoleon's invasion of Russia. This visualization is notable because it conveys six variables in an easily understandable way, including troop strength and losses, positions given as latitude and longitude, distance and direction of travel, significant dates, and temperature [Tuf01].

In the 19th and 20th century, people started to use "decorative" graphics for commercial and political purposes, with the goal of conveying certain statistical data to a broad audience. These visualizations often prioritized creativity and visual appeal, and maybe the communication of a selected message, over data accuracy. A common pitfall is the representation of one-dimensional data using two-dimensional area. We do not perceive differences of area proportionally to the actual differences of the data. This is expressed in the *"Lie Factor"* [Tuf01]. The human perception of visual elements such as position or area has been the subject of many studies, with researchers developing perception models that can serve as guidelines for accurately conveying quantitative or qualitative differences in a visualization.

Nowadays, more data than ever is being collected and stored, and massive volumes of data from many different domains are available to us. On the one hand, this provides unprecedented opportunities to gain valuable new insights. On the other hand, it also confronts us with an "information overload problem", which occurs if data is not selected correctly for the current task, processed inappropriately, or presented inappropriately [KAF$^+$08].

To address challenges when analyzing large amounts of data, several fields of research have emerged which focus on the theory of visualization. A strong emphasis has been placed on using methods from scientific fields such as psychophysics or cognitive psychology to develop theories and models of *vision science*, which can be used to choose optimal visual representations [Cle93].

Information Visualization (InfoVis) is a field of research that primarily deals with abstract data, unlike Scientific Visualization (SciVis), which focuses on data that has some kind of geometric or spatial representation, and as a result, an inherent mapping in the 2D or 3D space. Examples of scientific visualizations are volume or flow visualization. On the other hand, InfoVis is used to visualize data with no such spatial reference, often referred to as *abstract data*, for example, business data or demographic data. This data is often sourced from large databases and is *multivariate*, meaning it contains more than three dimensions. Traditional 2D or 3D visualizations such as scatterplots, line plots, bar charts, or histograms may not be sufficient for representing a multivariate dataset in its entirety. [Kei02, KAF+08].

Visual Analytics is an area of research closely associated with InfoVis. It focuses on developing interactive solutions based on established visualization techniques, with the goal of enabling visual data exploration and model building [KAF+08]. This knowledge discovery process follows the "Visual Information Seeking Mantra" introduced by Shneiderman: *"Overview first, zoom and filter, then details on demand."* [Shn96]. Since interaction is not the primary focus of this work, Visual Analytics will not be a major topic. However, the visualization tool that we developed does contain interactive elements, so Visual Analytics models and approaches have been taken into account.

2.2 Types of data

When we are looking for a visualization technique for a certain dataset, it is crucial to first understand what type, or types, of data it contains. Classifying data into clean and distinct categories is not always possible. Fundamentally, we can think of two types of data: *entities*, which are individual data values or objects, and *relationships*, or relations, between entities. Both entities and relationships can have *attributes* which can be in any scale of measurement [War04].

2.2.1 Scales of measurement

Stevens [Ste60] introduced a taxonomy consisting of four different measurement scales, focusing specifically on how they influence sensory communication. Each of these scales is different in how they are perceived by a user, and what mathematical or statistical operations can be performed on them. Because each visualization technique utilizes different scales of measurement, we must carefully consider their individual properties when choosing a display method.

- *Nominal* data includes categorical items, such as labels. It can be numerical or non-numerical, but it has no inherent order.
- *Ordinal* data is numeric data which can be ordered in a sequence. Each set of items has a rank quality, where one is higher or lower than the other.
- *Interval* data defines a gap between two values, for example, elapsed time.
- A *ratio* scale requires that there is a zero value, allowing for a comparison of ratios. Multiplication by a certain factor always results in the same relative change. For example, the Kelvin temperature scale satisfies this requirement, while the Fahrenheit and Celsius scales do not, they are considered interval scales.

In modern programming, interval and ratio scales are combined into the concept of real-number data. Similarly, integer data represents values on an ordinal scale, which are discrete and ordered.

Operations performed on the data may also be considered data themselves. Mathematical operations on numbers, or transformations of the database, may have to be visualized as well, for example through the use of animation [War04].

2.2.2 Scale vs physical information

Cleveland [Cle93] presents a model of graphical perception which differentiates between quantitative and categorical data. Data can be shown in two different ways in a visualization: *scale* information describes the units of measurements or category names which are typically used to label the axes, while *physical* information represents features such as the position of an element or the icon, size, or color assigned to the element.

When choosing a visualization technique, it is important to understand that different types of data have certain *physical* representations with which they are commonly associated. For example, the attribute of size, e.g. in a bar chart, is usually linked with quantitative information, using it to show categorical data may cause users to misunderstand the graph as representing a quantity, or a ranking between the categories. On the other hand, the color attribute usually depicts data categories. While it may be used to visualize discrete categories of ordinal data, e.g. in a choropleth map, it is not as effective for conveying metric intervals [War04].

2.2.3 Dimensions

Data may also be classified into the number of dimensions contained within the dataset. One-dimensional data may be as simple as a string of text, or a list of values. Temporal data can be considered one-dimensional data [Kei02], or as its own data type [Shn96].

Common visualization techniques for two dimensions include the x-y plot (scatterplot) or line plots. These plots can easily visualize both abstract and scientific 2D data. Maps are a special case of an x-y plot where the longitude and latitude are the x and y axes [Kei02].

In scientific visualization, three-dimensional data commonly contains digital models of real-world objects. Researchers of 3D computer graphics have developed advanced methods for digitally visualizing these objects. However, the visualization of abstract 3D data presents a significant challenge in the InfoVis field [Shn96].

While the terms *multivariate* and *multidimensional* are often used interchangeably in the visualization literature, some researchers argue that *multidimensional* data is more closely associated with SciVis. The visualization of scientific sample data with more than three dimensions is an emerging area of research. On the other hand, InfoVis deals with *multivariate* data, which is usually abstract data [dB04].

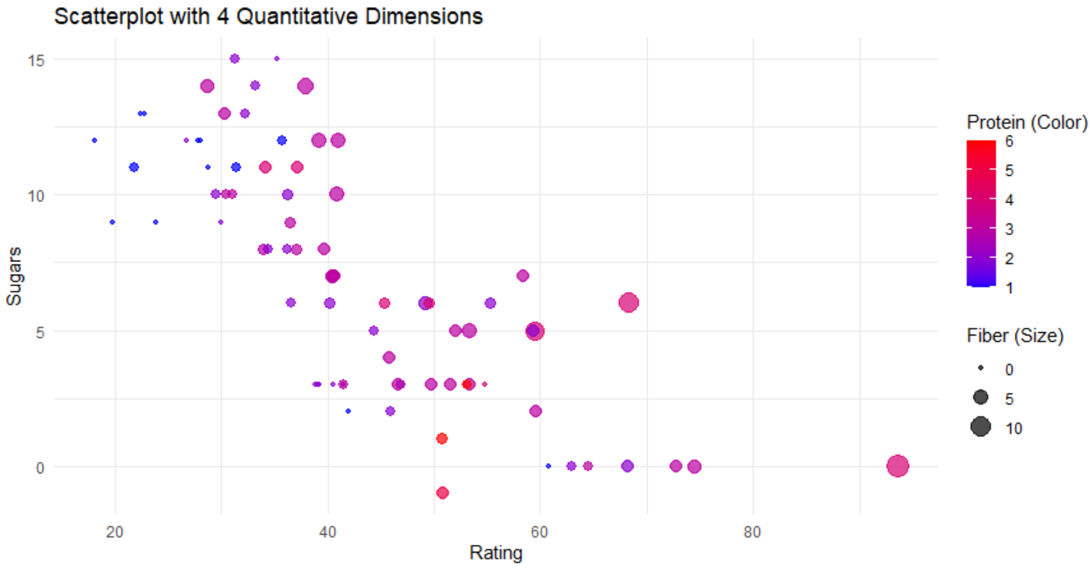
2.3 Visualizing multivariate data

Multivariate data contain more than three dimensions, and describe “homogenous sets of items by values of their attributes” [CvW11], with each attribute having its own associated domain. For example, a row might be a person, with columns describing attributes like gender, age, height, etc. Common tasks when working with multivariate data include “finding patterns, clusters, correlations among pairs of variables, gaps, and outliers.” [Shn96]

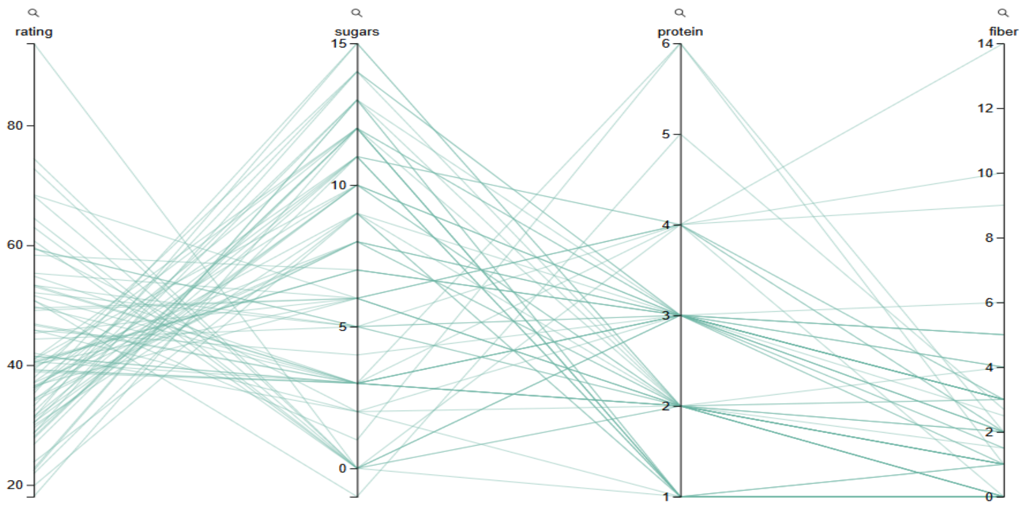
To demonstrate an example where multivariate visualization techniques can be advantageous over “traditional” techniques, we try to visually assess the “80 Cereals” dataset [Cra17] for correlations between different quantitative columns. First we create a scatterplot (Figure 2.2a). We can plot up to four dimensions by also using color and point size along with the x- and y-axes. In this plot, we can perceive the (negative) correlation between *ratings* (x) and *sugars* (y). We can also tell that *sugars* and *fiber* (point size) are slightly negatively correlated. Judging the correlation of *fiber* and *protein* (color) is possible, but it requires some focus, and it is difficult to compare exact values of the two dimensions. On the other hand, in Figure 2.2b, we see a PCP of the same columns. Here, each pair of axes is represented using the same graphical elements. We do not have to choose two dimensions that should be, for example, the x- and y-axis. While reading parallel coordinates might take some practice, an experienced user can quickly tell that the first two axes have a relatively strong correlation, while the other axis pairs are more weakly correlated. We can also read precise values from all four axes instead of having to estimate and interpolate color or point size.

2.3.1 Challenges when visualizing multivariate data

Multivariate data are often found in systems that automatically record large numbers of parameters for the purpose of data mining, resulting in datasets with numerous rows and columns. Extracting new knowledge from such datasets is often impossible with a simple



(a) A scatterplot of four quantitative attributes.



(b) A PCP of the same four attributes.

Figure 2.2: Four quantitative attributes from the “80 Cereals” dataset [Cra17] visualized using different techniques.

textual or tabular view, since only a small fraction of the data can be viewed at once [Kei02]. As a result, it was necessary to develop new methods for displaying the data.

Columns in multivariate datasets can contain a mix of numerical and categorical columns, in very different scales. Many visualization techniques are based on the comparison of sets of items by their relation. This may be difficult if the data contains different data types [BBK+18].

A multidimensional visualization may include or leave out any of the dimensions in a dataset, and arrange them in any order. This leads to the *curse of dimensionality*, as the number of columns increases, so does the number of possible mappings. Interesting relationships between certain dimensions may be overlooked [BBK+18]. Conflictingly, it is often not possible to know what properties we want to highlight in a graph, as we have not discovered these properties. Without this knowledge, we may choose an unsuitable visualization technique and end up missing important information [vW06]. This paradox affects all visualization in some way, but is especially important to keep in mind when working with multivariate data.

Fundamentally, the mapping of high-dimensional data into a two-dimensional visual representation is not a trivial task. Graphical elements have to be carefully chosen to avoid cognitive overload, and to make sure different dimensions can be easily distinguished. Inevitably, some details will be lost [Cha06]. This may prevent viewers from discovering certain characteristics of the data. However, visualization designers can address these issues by including interactivity [vW06], for example, by letting the user choose columns to be displayed in a secondary visualization.

2.3.2 Examples of multivariate visualization techniques

Keim and Kriegel [KK96] defined four categories of multivariate visualization approaches: geometric, icon-based, pixel-oriented, and hierarchical and graph-based techniques.

Geometric techniques use certain geometric projections in combination with statistical methods to find interesting and expressive ways to display data. One such technique is a scatterplot matrix, shown in Figure 2.3. Dimensions are represented as rows and columns, each cell contains a scatterplot of two given variables. Parallel coordinates are a popular multivariate visualization method, they are discussed along with similar methods in Section 2.4. These methods are able to display any number of dimensions, but also quickly become cluttered and impractical. To address this, the class of geometric techniques also includes *projection pursuit* methods, which aim to automatically discover combinations of dimensions that provide the most insight into the data [KK96]. How "interesting" a subset of dimensions is, can be determined through *dimensionality reduction* methods such as principal component analysis (PCA) [EDF08], which extract a set of principal components while preserving variance [JFJW09]. Generally, geometric methods are suitable for large and high-dimensional datasets, and are most effective for pattern and outlier detection. Potential issues are visual cluttering, and loss of information through an improper selection of the axis order.

The idea of *icon-based* techniques is to visualize data by associating them with a certain type of icon based on their dimension. A unique, well-known example are Chernoff Faces (Figure 2.4a). Each data item is represented as a human face, each dimension is a specific facial feature. These features are drawn differently, depending on the value of the given dimension [Che73]. Similarly, the stick figure technique (Figure 2.4b) maps dimensions to limbs of a stick figure, the data values dictate the lengths and angles of limbs [PG88].

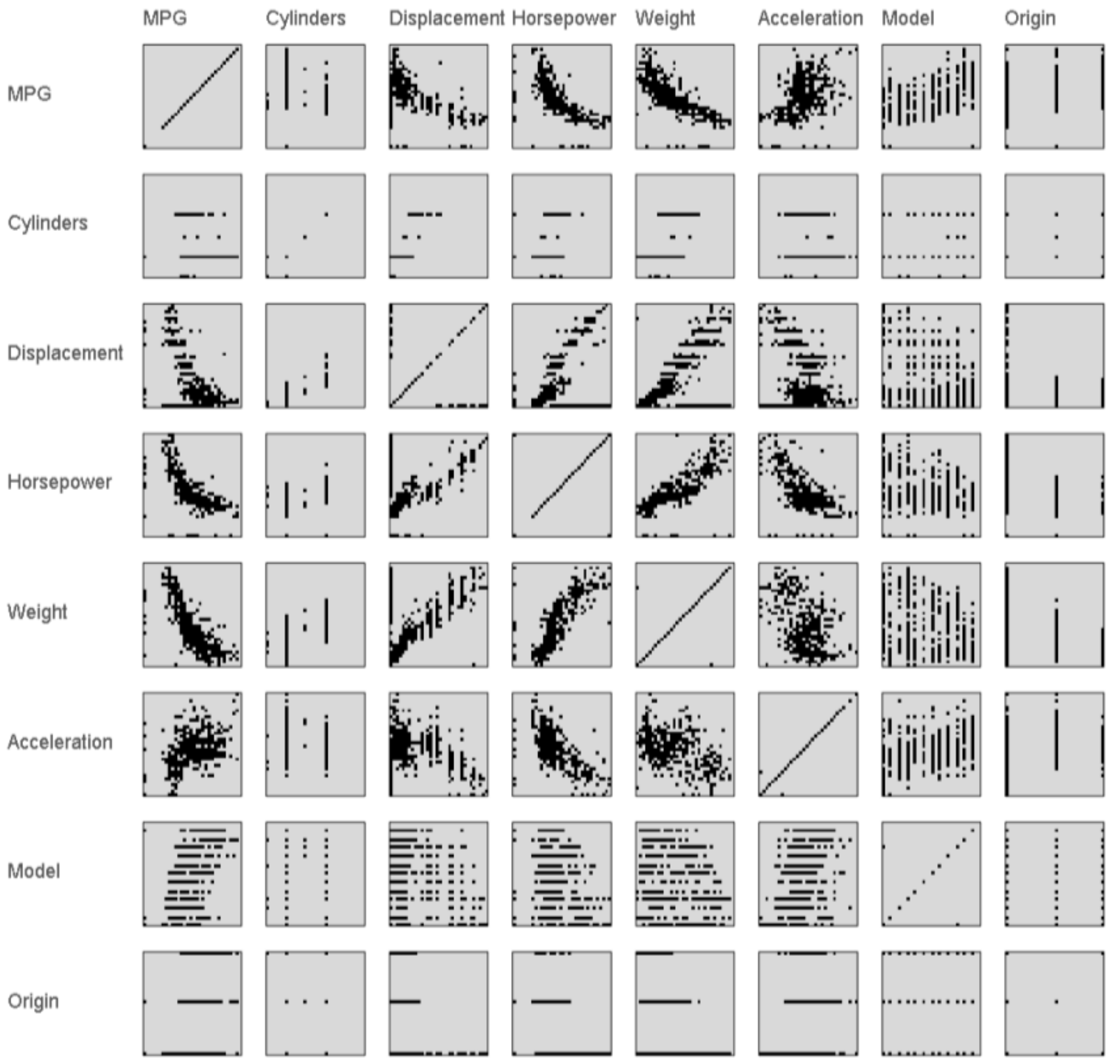
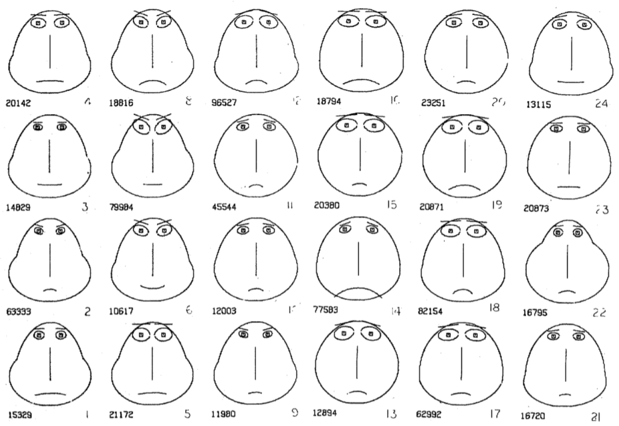
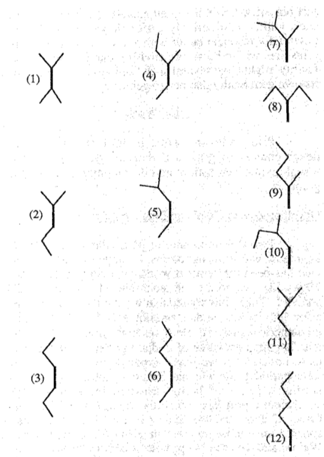


Figure 2.3: A scatterplot matrix gives a quick overview of all axis pairs. Each cell contains a scatterplot of the two dimensions given along the x- and y-axis. Image taken from Elmqvist et al. [EDF08].

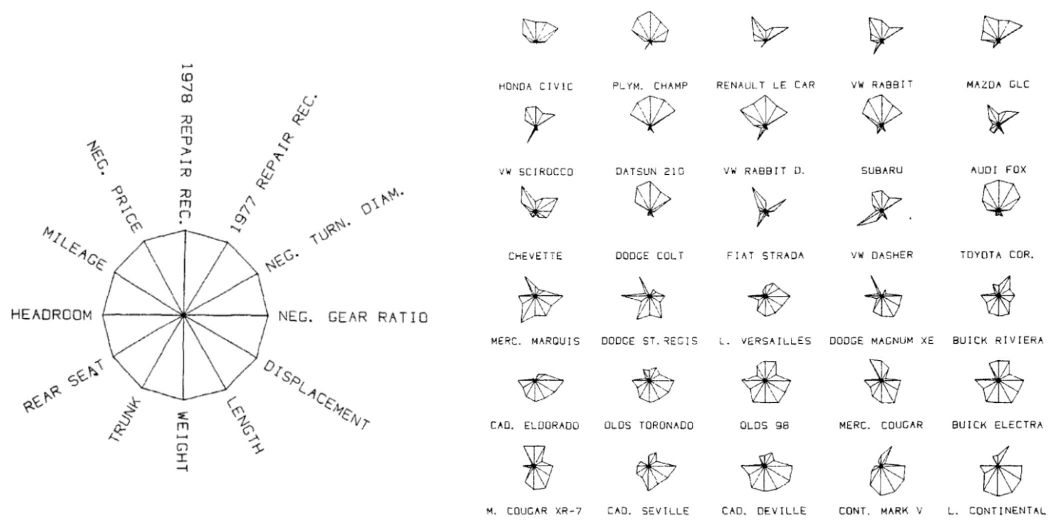
In both these techniques, two dimensions are already represented in the x and y axis - they determine the width and height of the figures - while the rest are conveyed in the figures themselves [KK96]. Star plots, also known as radar charts, seen in Figure 2.4c, can also be considered an icon-based method, since each data item is represented as a star-shaped symbol [Cha83]. In this method, axes are arranged in a star pattern, with lines connecting them, similar to parallel coordinates. Ideally, all axes are ordered in the same direction, so different samples can be compared based on the area of the star [PPLF24].



(a) Chernoff Faces. Each face represents a row in the dataset, each facial feature an attribute. Image taken from Chernoff [Che73].



(b) Stick figure technique. Each figure represents a row, each limb an attribute. Image taken from Pickett and Grinstein [PG88].



(c) Star plot/radar chart. Axes are arranged in a circle, resulting in a unique shape for each row. Image taken from Chambers [Cha83].

Figure 2.4: Examples of icon-based multivariate visualization techniques.

An example of a two-dimensional *pixel-based* visualization method can be seen in Figure 2.5. Correspondingly, a multivariate pixel-based visualization consists of several windows representing different dimensions. Every data item is mapped to one pixel in each window, which is colored according to the value of that dimension. These techniques prevent visual clutter and overlap, as each element has a strictly defined position and size. They are capable of supporting large datasets, only being limited by

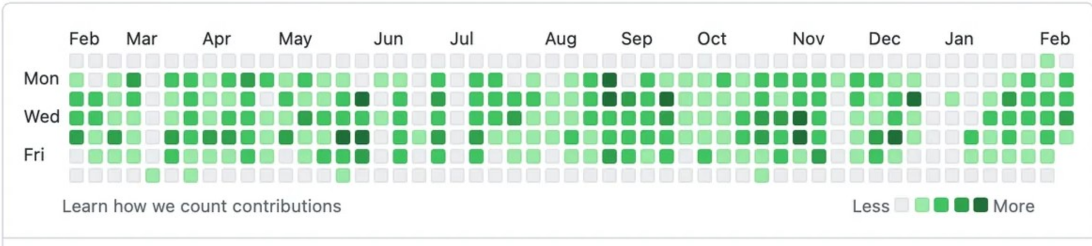


Figure 2.5: The GitHub activity feed is an example of a pixel-based visualization. Each pixel represents a date, the number of contributions on a certain date is conveyed using a color encoding. Image taken from GitHub, Inc. [Git24].

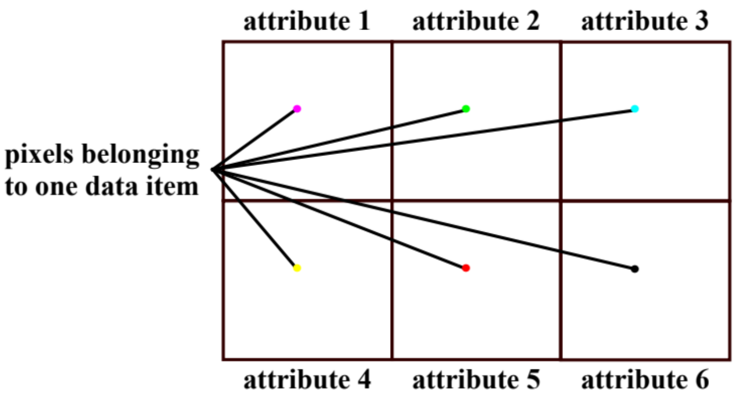
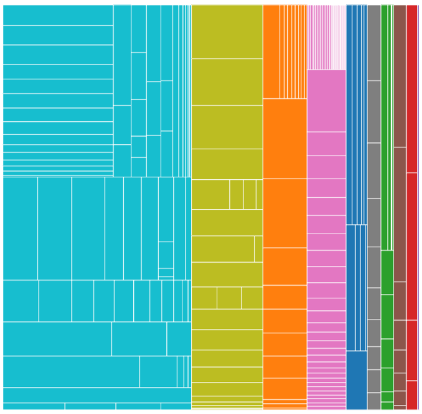


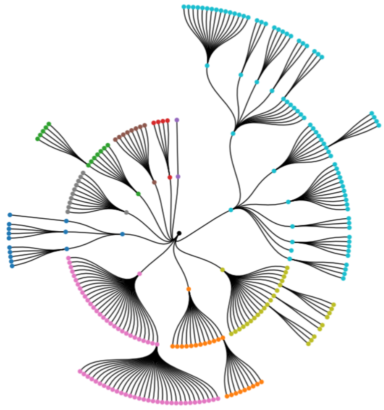
Figure 2.6: Query-independent pixel-based multivariate visualization technique. Each row is represented using one fixed pixel in each rectangle, which represents an attribute. Image taken from Keim and Kriegel [KK96].

the screen size in pixels. One challenge associated with pixel-based techniques is the ordering of the data. Figure 2.6 shows a *query-independent* pixel-based method, where the entire dataset is visualized, with the order of items being determined by thea specific attribute. On the other hand, *query-dependent* methods visualize subsets of the data based on a user query for a certain value or value range [KK96].

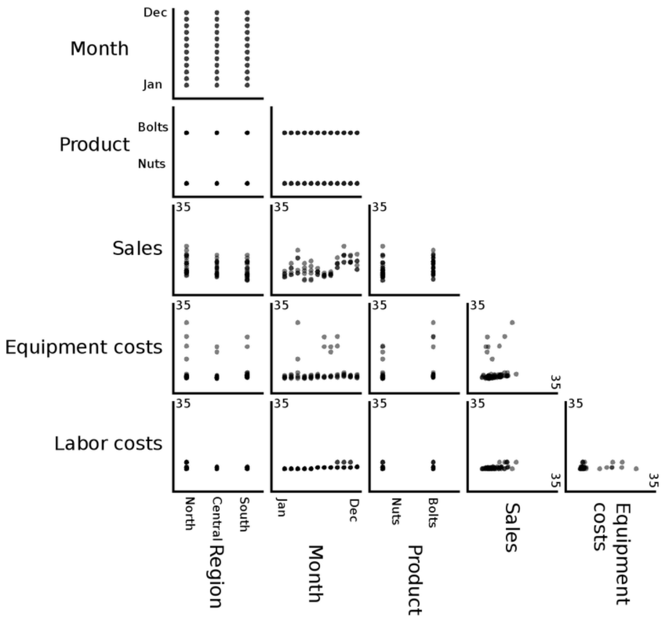
A variety of techniques exist that can be used to visualize *hierarchical* data structures. Classic examples are *space-filling* methods like treemaps (Figure 2.7a), or *node-link* representations such as radial trees (Figure 2.7b) [ZS21]. For multivariate hierarchical data, the standard approach is to subdivide k-dimensional space into 2D-subspaces which are then arranged in a hierarchical structure. An example of this is the dimensional stacking method [LWW90] seen in Figure 2.7c.



(a) Treemaps visualize hierarchy and proportions of values using area. Image taken from Zheng and Sadlo [ZS21].



(b) Radial trees visualize a hierarchical data structure. Image taken from Zheng and Sadlo [ZS21].



(c) Dimensional stacking allows visualization of multiple categorical variables on one horizontal or vertical axis. Image taken from Im et al. [IML13].

Figure 2.7: Examples of hierarchical visualization techniques.

2.4 Parallel coordinates

Parallel coordinates, also known as parallel coordinates plots (PCPs), are one of the most popular multivariate visualization techniques. The idea can be found as early as 1880. Figure 2.8 shows a visualization of rankings of US States across multiple attributes, developed by Henry Gannett [Fri08]. In 1885, Philbert Maurice d'Ocagne introduced a

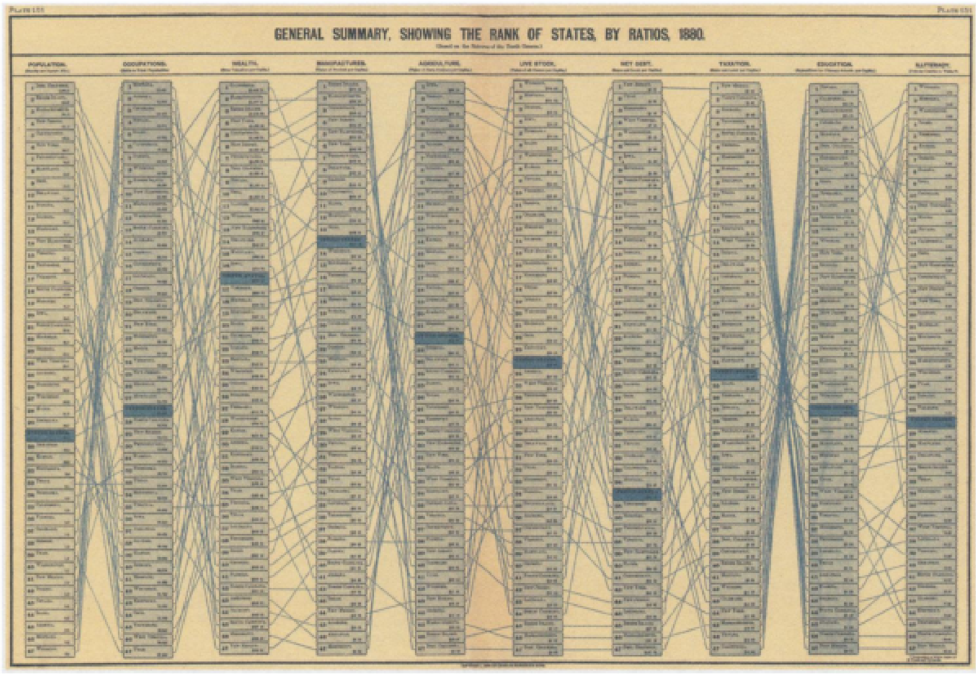


Figure 2.8: Henry Gannett's 1880 graphic ranking US States by different attributes, utilizing parallel coordinates. Image taken from Friendly [Fri08].

similar concept as part of his work on Nomography [d'O85, HW13]. Figure 2.9 shows an example of a *nomogram*, and how it may be used for calculation. In 1985, Alfred Inselberg introduced what we know today as *parallel coordinates* [Ins85], having developed the idea while studying multidimensional geometry and wishing to find a way to visualize multidimensional objects, instead of simply working with mathematical equations. Inselberg

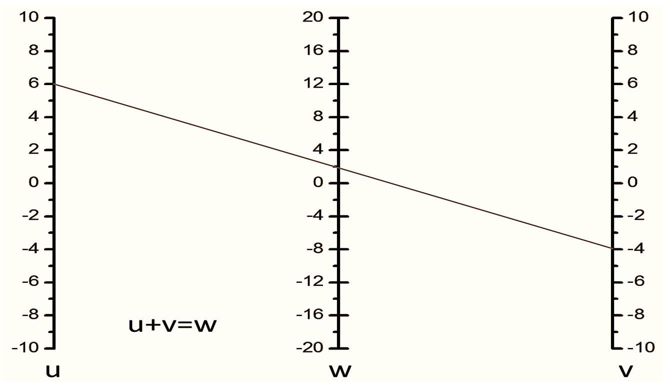
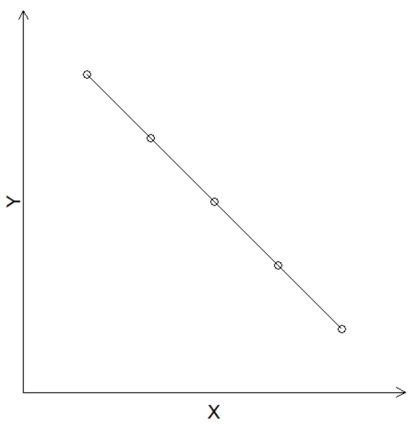


Figure 2.9: A nomogram used for mathematical calculations, in this example, addition. Image taken from Esaulov [Esa20].

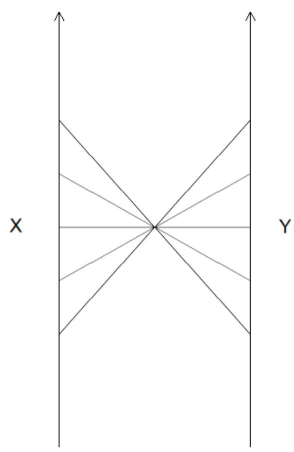
states that he was initially not aware of d'Ocagnes work, but points out that nomography was intended mainly for computational applications, while parallel coordinates are meant to be specifically a visualization technique [Ins09]. Beyond the idea of visualizing multi-dimensional geometry, Wegman introduced the parallel coordinates plot as an InfoVis technique that has become popular in exploratory data analysis [Weg90, HW13].

2.4.1 Implementation of parallel coordinates

To understand the basic motivation behind PCPs, we first examine a traditional orthogonal line plot (Figure 2.10a). Showing the relationship between two dimensions is trivial, a third can simply be added by using a 3D representation, or for example, by using color or symbols. For PCPs, the x- and y-axes are moved so that they are parallel to each other (Figure 2.10b). A data sample is now represented as a polyline which intersects each axis at the point corresponding to the coordinate of that dimension. With this new layout, we may add as many axes as needed, by arranging them next to each other. Just as in an orthogonal plot, we can still visually infer the relationship between two dimensions by recognizing the pattern of lines. Figure 2.11 shows common patterns in a 2D scatterplot, and the equivalent patterns in a parallel coordinates system. An interesting property of parallel coordinates, shown in Figure 2.12, is the *point-line duality*: a point (x, y) in Cartesian coordinates is represented as a line in the parallel coordinate domain, and the other way around [Ins85].



(a) A line plot, using an orthogonal coordinate system.



(b) A PCP, using a parallel coordinates system.

Figure 2.10: The same data in an orthogonal line plot and an equivalent PCP.

Heinrich et al. [HW13] defined a *parallel-coordinates system* as the axis layer, and a *parallel-coordinates plot* as the sample layer (lines representing data samples). In a *composite parallel-coordinates plot*, an additional visualization layer may be added on top of the plot.

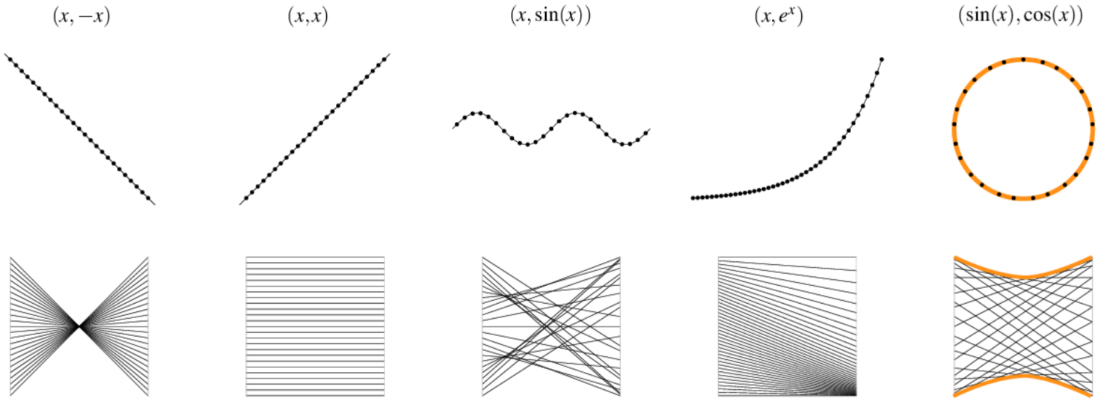


Figure 2.11: Patterns in Cartesian coordinates (top row) and parallel coordinates (bottom row). Image taken from Heinrich et al. [HW13].

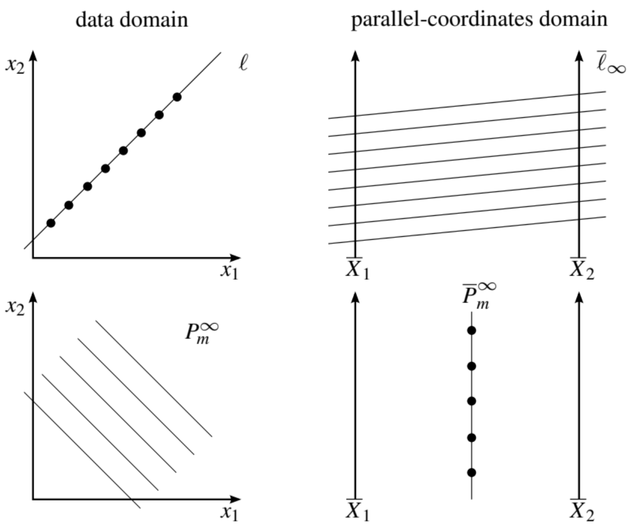


Figure 2.12: A comparison of the same patterns in a Cartesian coordinate data domain (left) and parallel coordinate domain (right), showcasing the *point-line duality*. Image taken from Heinrich et al. [HW13].

The axes in a parallel coordinates system may be oriented horizontally or vertically [AA01], although a vertical orientation is found more commonly. Classically, PCPs represent data items as polylines, consisting of one vertex for every dimension, connected by straight line segments. Various approaches have been introduced that use curved lines instead. This can be for the purpose of encoding additional information [The00], or for better readability [HW13].

PCP visualizations are often paired with interactivity. Common interactive techniques are highlighting selected lines, brushing, and reordering or flipping of axes. Highlighting is crucial, since polylines are usually not labeled, and often overlap with each other in dense datasets. It allows users to identify individual samples, and observe their values across all dimensions [Wei19]. Similarly, brushing makes it possible to isolate a subset of data, which can reveal valuable information in large datasets. For example, in Figure 2.13, cars where the *power* attribute is within a certain range are selected, and we then can see how the other variables are distributed for those samples. Brushing is often used in multi-view visualizations to link data samples across multiple plots [HW13].

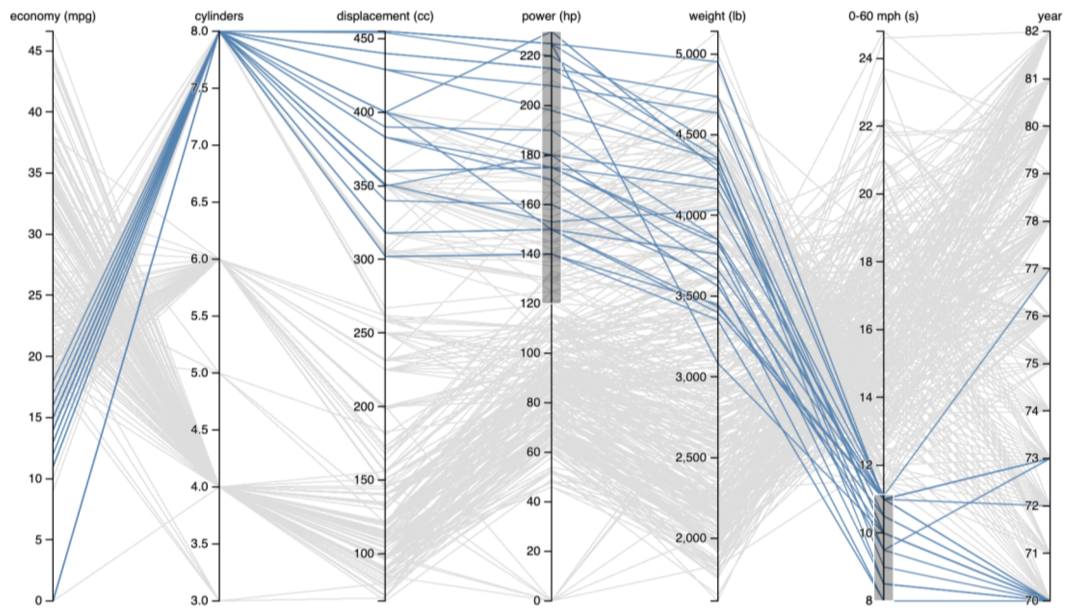
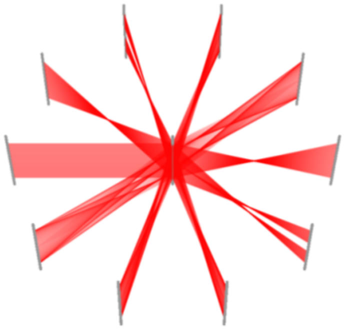


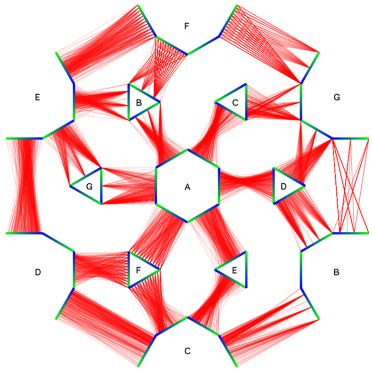
Figure 2.13: Brushing technique: highlighting a subset of data based on a range of values from a given axis. Image taken from Weidele [Wei19].

A major challenge when using parallel coordinates is the *axis order problem* [HW13]. Only the relationship between two adjacent axes can be displayed at a time, meaning we would have to redraw the plot many times to view every possible pairing, or else we might miss an interesting observation. Allowing the user to manually rearrange, flip, add, or remove axes can be helpful in the knowledge discovery process. When dealing with a large number of dimensions, an automatic ordering method may be necessary.

Dimension arrangement is a major topic of research concerning not just parallel coordinates, but multidimensional visualization in general. Ankerst et al. [ABK98] proposed an ordering method based on the *similarity* between each pair of dimensions. Several similarity measurement methods have been introduced, with a commonly used metric being Pearson's correlation coefficient (PCC) [LHZ16]. For example, axis pairs with the highest PCC can be placed next to each other, making it easier to perceive correlations.



(a) Multi-relational 3D parallel coordinates combine a specific axis with all other dimensions. Image taken from Johansson et al. [JWFLC08].



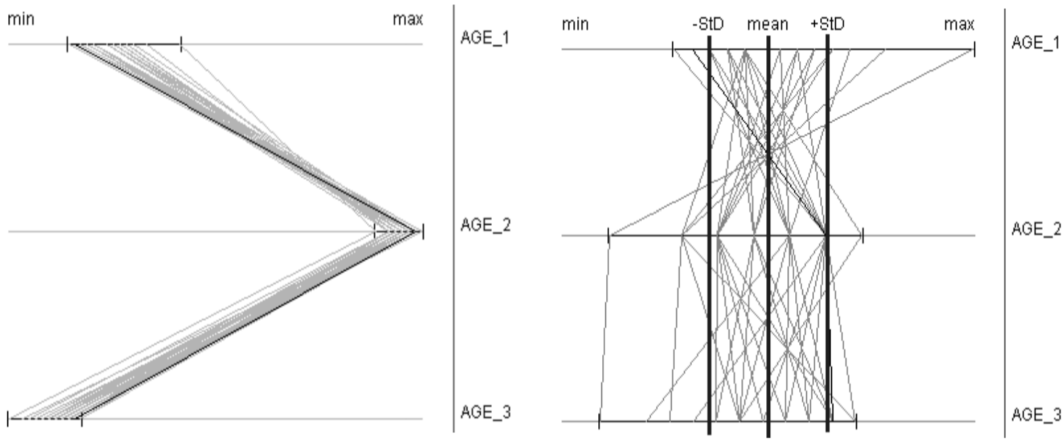
(b) Many-to-many relational parallel coordinates contain duplicate axes, showing all possible axis combinations at once. Image taken from Lind et al. [LJWC09].

Figure 2.14: Examples of relational PCP techniques. Compared to traditional PCPs, more axis combinations can be judged at a glance. This can be advantageous for discovering correlations.

Other techniques may instead focus on highlighting outliers or clusters [JFJW09]. Peng et al. [PWR04] noted that the axis order problem is closely related to visual clutter reduction techniques, and defined a good axis order as one which minimizes visual clutter. For parallel coordinates, they noted that a clutter-reduced axis order allows for better perception of patterns and outliers. Another approach is to use dimensionality reduction methods such as PCA, although this can be unintuitive for the user and may hide interesting properties [JFJW09].

Multi-relational methods such as the many-to-many relational parallel coordinates introduced by Lind et al. [LJWC09] present another potential solution to the axis ordering problem. Instead of having to create multiple plots to show different permutations of axes, all possible combinations are displayed in one plot. Figure 2.14b shows this concept. The letters label the different axes and a color coding is added to clarify axis direction. This method was found to be more effective than traditional PCPs for judging correlations between dimensions. A downside of this technique is that it can only handle a very limited number of axes, and requires one axis (the one located the center) to be prioritized.

Not just the axis arrangement must be carefully selected, but also the scaling of the axes themselves. If axis values are in very different ranges, such as in Figure 2.15a, or outliers are present, judging correlations between objects becomes difficult. This may be resolved by scaling axes using a normalization technique such as the mean (Figure 2.15b) [AA01]. In this example, all axes were given in the same unit of measurement, *age*. However, a characteristic of parallel coordinates is that each axis may be in a different scale and range. While PCPs are considered more suitable for continuous data, categorical data can also be plotted, for example, by assigning each category a numerical value. Displaying a mix of categorical and continuous data may require a more advanced visualization technique [Sch22].



(a) A PCP before normalizing the axes. (b) The same PCP with normalized axes.

Figure 2.15: Normalizing axes using the mean to reveal information in datasets containing distinct ranges of values. Images taken from Andrienko and Andrienko [AA01].

2.4.2 Applications of parallel coordinates

A large focus of study has been the relationship between different visualization techniques and analysis tasks. For example, Johansson [JWF15] studied various plot types that have been introduced over the years, and presents an overview of which techniques evidently performed better or worse at certain tasks. The list of analysis tasks, which they used as the basis of their study, included the visual analysis of correlations and clusters, discovery of outliers and patterns, value retrieval, and tracing of lines.

Andrienko and Andrienko [AA01] formulated a list of tasks for which parallel coordinates were effective. Generally, these fall into two categories: evaluations of distribution, similarities, and relationships between objects (represented as polylines), as well as evaluations of correlation between dimensions (represented as axes).

A common criticism of PCPs is that they are unintuitive. First-time users are often unable to infer information from the graph at first glance. This is due to the use of a parallel coordinates system, which is very distinct from the more common orthogonal Cartesian coordinate system. Parallel coordinates visualize attributes, relationships, and patterns in a way that users may not be as familiar with. In a user study, Siirtola et al. [SLHR09] found that participants who had some basic experience with graphing software, but were unfamiliar with parallel coordinates, were able to solve simple tasks using PCPs after just a short time. In the same study, first-time users remarked that they found PCPs “messy”, and that they could not distinguish graphical elements at first. This problem may be exacerbated by *overplotting*, which occurs if too many samples (lines) are plotted. Figure 4.1a shows an example where overplotting occurs: a highly dense dataset is plotted without applying any *clutter reduction techniques*. These techniques are divided into *data-driven* techniques, where the dataset is preprocessed in some way, and *screen-based* approaches, which adapt the display method to deal with the given data [HW13]. Some examples of screen-based methods are density- and frequency-based plots [JWLJC06], as well as bundling techniques [PBO+14].

Density-based plots can be used to reduce visual clutter, especially in very large datasets. Instead of plotting each item as a line, the *line density* at each point, which is given by the proximity of lines [HW13], is calculated and plotted. An overview of different density estimation techniques that may be used for such plots, was given by Moustafa [Mou11]. One such method is *edge-bundling*, shown in Figure 2.16. Unlike other density-based methods, which show only aggregated information, this approach preserves the visual representation of data items as lines. It was found to be more effective for correlation and cluster judgements, as well as subset tracing, than traditional PCPs [PBO+14] .

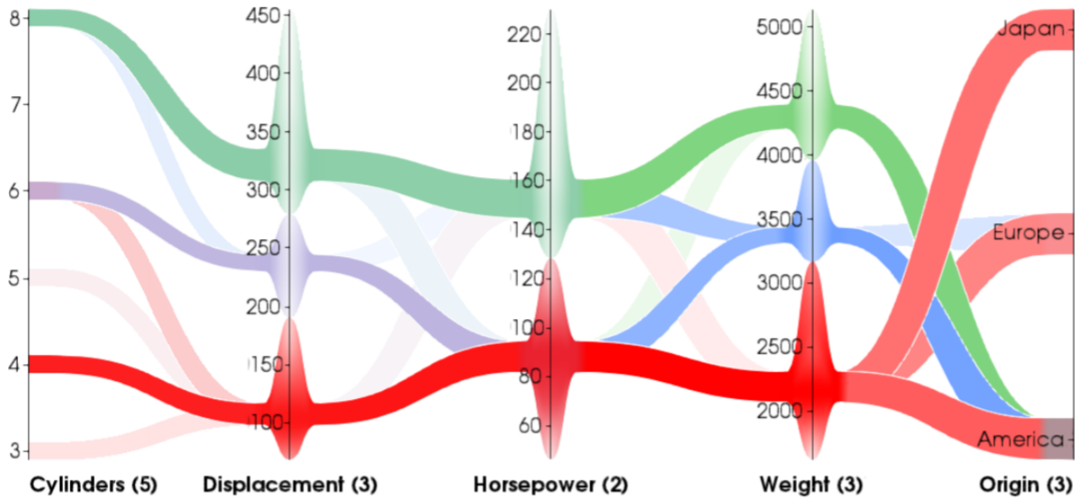


Figure 2.16: The edge bundling technique groups data based on line density, while preserving correlation and cluster information. Image taken from Palmas et al. [PBO+14].

A similar challenge to overplotting is presented by the *line tracing problem*. This occurs if two samples have identical values in one or two adjacent dimensions, meaning they touch at an axis or overlap on a line segment. In the classic polyline display method, the two samples cannot be distinguished, but a curve model, shown in Figure 2.17, can create a more readable representation.

An advantage of PCPs is their ability to handle different types of data, even a mix of numerical and categorical data, in one display. However, one data type that can be difficult to deal with is *time*. Parallel coordinates closely resemble time-series plots, but actually capturing time information in a PCP is not trivial as just adding an axis [HW13].

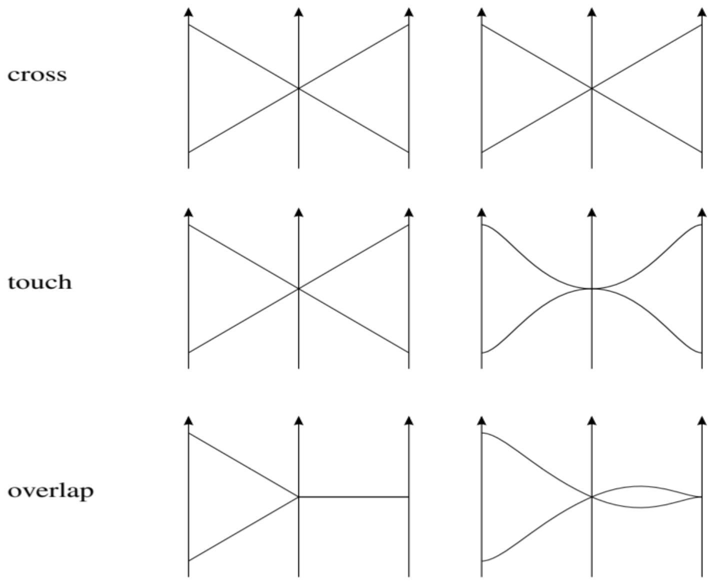


Figure 2.17: Using curved lines to mitigate the line tracing problem. Image taken from Heinrich et al. [HW13].

To visualize how a multivariate dataset changes over time, researchers proposed to use 3D visualization methods [WLG97] or density-based representations [JLC07]. Most of these approaches require some tradeoff or loss of information, so the visualization method has to be selected carefully.

Several different approaches have been introduced to handle temporal and spatial information. Multivariate samples may be given over a sequence of time steps, or at different locations. This is often the case in scientific fields such as biology or functional genomics, or in simulation applications. In order to fully display the datasets, a series of duplicate plots would be necessary [HW13]. Three-dimensional techniques try to address this. An example can be seen in Figure 2.18. 3D plots allow for clutter reduction, but make it more difficult to perceive correlations and patterns [JWLJC06, JWF15]. Furthermore, they may cause occlusion and distortion due to the projection from 2D to 3D [HW13]. Another 3D-based approach constitute multi-relational 3D parallel coordinates, shown in Figure 2.14a. These were shown to be as effective for pattern recognition as 2D parallel coordinates, but can only handle a small number of variables [JWFLC08].

PCPs may also be used to display hierarchical or conditional data. Generally, visualization of conditional data is paired with interactivity: the user can choose to expand or collapse sections in order to only see certain subsets of the data. In PCPs, this can be implemented for example by having *parent* and *child* axes that can be individually expanded or collapsed, also known as *drill-down* and *roll-up* operations [AOS15]. Another approach is to show or hide dimensions based on whether certain conditions are met within the data, combined with, for example, a brushing method as an input mode [FWR99, Wei19].

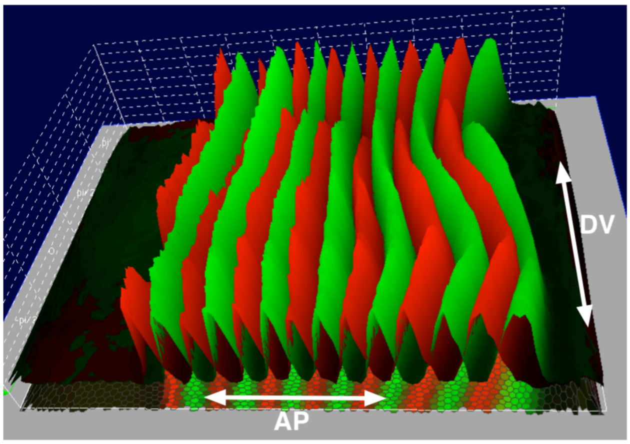


Figure 2.18: A 3D-based parallel coordinate technique. A third dimension is added to represent time. Image taken from Ruebel et al. [RWK+06].

Researchers have increasingly found use cases for PCPs in industry and other applications. Some more recent examples include climate model analysis [SST+12] and visualization of machine learning models [HK24]. While new visualization techniques are developed continuously, research also includes evaluations of the practical applicability of PCPs, and comparisons to other visualization techniques. Furthermore, it aims to find new solutions to problems like axis ordering and clutter reduction [JWF15].

2.5 Evaluation of visualization techniques

Evaluation defines what makes a visualization effective for a certain task. Wehrend [WL90] formulated a list of operations that a user may want to perform while viewing a visualization, including identifying, locating, distinguishing, categorizing, clustering, ranking, comparing, associating, and correlating. Other taxonomies such as that of Zhou and Feiner [ZF98] include both the *visual tasks* performed by the viewer as well as the task of visualizing something with a certain *presentation intent*. A visual technique may be chosen based on the set of visual tasks, which help to achieve the presentation intent.

The development of a new interactive system usually goes through multiple cycles in the *process of interaction design*. This process encompasses four steps [PRS15]:

1. Definiton of prerequisites
2. Finding alternative solutions
3. Prototype development
4. Evaluation

In the visualization field, especially the final *evaluation* step presents challenges. An effective visualization technique is often defined as one that enables *insight* into data, however, insight is not an easily quantifiable or measurable metric. It is complex, unpredictable, accumulative, and depends on existing domain knowledge of the user [Nor06].

2.5.1 The visual perception process

Cleveland [Cle93] introduces a model for graphical perception which describes the process of understanding, or *decoding*, a visualization. This process consists of *pattern perception*, which is the decoding of *physical* information, and *table look-up*, the decoding of *scale* information.

Pattern perception is divided into three stages. *Detection* is the time it takes for the user to discover the graphical element that marks a data point. In parallel coordinates, this would be a line, in a scatterplot, it could be a point or symbol. *Assembly* is the process of grouping and filtering the elements in the chart, and determining which are relevant for the current operation. *Estimation* is the assessment of multiple elements, and making a judgment of their relation to each other, e.g. whether they are equal, bigger or smaller than each other, or about their ratio.

Table look-up begins with *scanning*, where users determine the location of a point as well as the extent of the axis. Users then *interpolate* the position of the point in relation to the axis as a fraction. Finally, during *matching*, users read the labels and convert the location to a value, interpolating between two ticks on the axis if necessary.

The *latency*, or how long the task takes, determines the effectiveness of each step. The latency can be infinite if the task fails, for example, if a graphical element cannot be detected due to being hidden by another element. Generally, the effectiveness of a visualization is determined by how fast each step can be performed, or how accurate the initial estimation is to the actual data.

2.5.2 Determining the effectiveness of visualization techniques

Tufte [Tuf01] analyzed visual perception, and points out that it can be a very individual process that differs for every observer. Additionally, the effectiveness of a visualization technique can be highly dependent on the type of analysis task, a technique may be ideal for solving a certain problem, but completely unfit for another one.

Distinct types of research are required to drive innovation in a scientific field such as visualization research. *Problem-driven* research aims to find solutions to real-world problems, usually by studying actual users. *Technique-driven* research primarily focuses on developing new techniques, without a connection to a specific user requirement. [SMM12]. As we have discussed previously, a lot of knowledge about visual perception can be gained from other scientific disciplines such as psychology [Cle93]. Yet, a crucial step in the development of new visualization methods is to conduct empirical studies.

A common element of a visualization evaluation study is the measurement of performance, such as the time a user takes to read a chart, or the error rate of answering specific questions about the data. Subjects are also often asked for verbal feedback. However, not all aspects of usability and effectiveness may be evaluated just through these methods [APM+11].

The most common method for evaluating visualizations is to conduct controlled experiments on benchmark tasks. They are usually precisely defined, conducted under very specific conditions, and only assess objective, measurable metrics such as task completion time, or error rate. North [Nor06] emphasizes the importance of moving away from such tightly controlled experiments, and finding more open-ended and qualitative methods to measure insight. These measures of insight should not just be taken into account for the evaluation, but during the entire design cycle of a visualization technique [PvW09].

Interestingly, in several of the research works that were analyzed as part of this thesis, which feature user studies, the subjects themselves were experts in visualization, or professionals in another relevant field [PvW09, SR06, PVF05]. This was mostly the case in studies which evaluate a specific visualization method, or try to develop a solution for a specific technological use case. Working with domain experts is crucial in this case, many visualization techniques can be quite complex and require domain knowledge to be used effectively [SMM12].

On the other hand, studies where the goal was to gain a deeper understanding into visual perception often used a more diverse range of subjects. A common approach more recently has been the use of crowdsourcing. Researchers such as Heer and Bostock [HB10] used Amazon's Mechanical Turk (MTurk) platform [Ama18] to study visual perception. Crowdsourcing can be a way to effectively collect large amounts of quantitative performance data on a diverse set of users. It is not suited for tasks that cannot be easily split into small *microtasks*, and that require subjective or qualitative judgments [KCS08]. Additionally, some time has to be spent removing inaccurate results, or developing methods to filter out users who may not correctly perform the task.

2.5.3 Visual area judgments

One of the topics that Heer and Bostock [HB10] looked at, is how accurately users are able to visually perceive differences between areas. This is relevant for visualizations that use area to show comparisons between ordinal data, or to visualize some kind of hierarchy.

Stevens' power law [Ste60] expresses the relationship between the physical intensity of a stimulus, expressed as ϕ, and its perceived intensity ψ:

$\psi = k \cdot \phi^n$

where k is a constant depending on the units used. There is a bias, expressed as the exponent n, that measures how much users either underestimate or overestimate the actual magnitude. An exponent of 1 means an accurately perceived stimulus. Stevens stated that the exponent of *visual area* is 0.7 while that of *visual length* is 1. This means that when presented with two 2D shapes where B is twice as large in area as A, humans will generally estimate B as less than double the size of A. Cleveland et al. [CHM82] conducted an experiment where the average exponent of visual area came out to be 0.95, confirming that changes in area are usually underestimated. However, the results ranged from 0.58 to 1.27, suggesting that accuracy of area perception varies strongly per person.

Cleveland and McGill [CM84] proposed an order of *elementary tasks* of graphical perception, based on how accurately they are judged: position is the most accurate one, followed by length, then area. Direction and angle are included with length. Finally, volume, shading and color are the least accurately perceived stimuli.

Heer and Bostock [HB10] tested various graphical encodings. They were able to replicate the results of Cleveland and McGill. Position judgments performed best, followed by length and then area. On top of this, they tested how aspect ratio would affect judgments of rectangular area. For this, they used two different display conditions: two rectangles by themselves, as well as rectangles in a treemap layout (Figure 2.19). Interestingly, for both conditions, comparisons between two squares were judged least accurately. In general, both display conditions showed the same results, suggesting that rectangular treemap elements are judged similarly to stand-alone rectangles. A more detailed analysis of area judgments in relation to treemaps has been conducted by Kong et al. [KHA10].

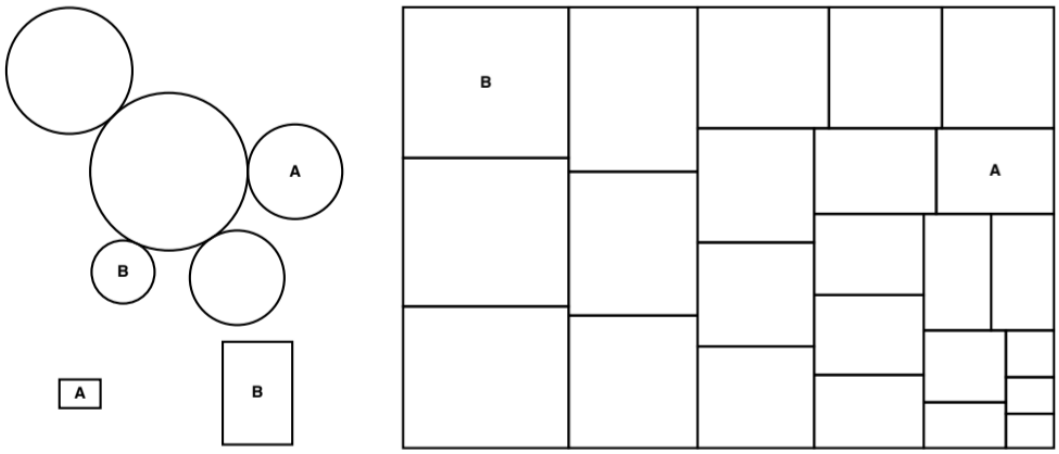


Figure 2.19: Center-aligned rectangles (bottom left), Treemap (right). Image taken from Heer and Bostock [HB10].

2.6 Relevance of aspect ratio in data visualization

The aspect ratio of a 2D chart, also sometimes called the *shape parameter*, is defined as the ratio of the height and width of the *data rectangle*, the area between the minima and maxima of data along each axis. An aspect ratio of 1 represents a square data rectangle. The aspect ratio plays a crucial role in graphical perception, and greatly influences how effectively a visualization can be "visually decoded" [CMM88]. Certain interesting properties of a dataset may only be perceived at specific aspect ratios, while being hidden at others. Figure 2.20 shows an example of two line plots with different aspect ratios visualizing the same dataset, which shows the number of sunspots over time. Looking at the upper image, we can tell that this process is cyclical. In the lower image, an important observation becomes visible: within each cycle, the number usually rises more quickly than it shrinks [Cle93]. Bertin [Ber11] recognized the importance of maximizing *angular legibility* in line charts, or the ability of the viewer to correctly perceive the rate of change in a line.

2.6.1 Banking to 45°

Plenty of researchers dealt with the topic of aspect ratio in visualization, and proposed different ways of choosing an optimal value. Cleveland et al. [CMM88] reviewed some of the existing literature, specifically focusing on line plots. Some researchers proposed either specific fixed aspect ratios, or multiple fixed aspect ratios that should be chosen from depending on the data, the medium, or other factors. Cleveland et al. claimed that these earlier works did not present sufficient scientific evidence or conduct empirical studies. Some authors did recognize the importance of looking at the orientation (angle) of different line segments to calculate an ideal aspect ratio. Based on this, Cleveland et al. introduced their own principle. As we saw in Figure 2.20, the aspect ratio is poorly chosen if the slopes of individual line segments cannot be judged accurately. An ideal aspect ratio is given if the midangle (average orientation) of line segments is centered around 45° for positive slopes and -45° for negative slopes. This is given if the *median absolute slope* of all line segments is 1. This principle has become known as *banking to 45°* [WWF$^+$19], the banking technique introduced by Cleveland et al. [CMM88] is called Median Absolute Slope procedure (MS).

Several other methods have been found, which can be used to find an ideal aspect ratio that fulfills the 45° principle. Cleveland [Cle93] introduced Average Absolute Orientation (AO) and Weighted Average Absolute Orientation (AWO) methods, which use the orientation of line segments rather than the slope. Presented by Heer and Agrawala [HA06], Global Orientation Resolution (GOR) and Local Orientation Resolution (LOR) methods attempt to maximize angles between all, or successive, pairs of line segments. Along with this, they developed the approach of *multi-scale banking*, which combines multiple banking techniques, resulting in a set of aspect ratios that are then shown in a combined graphic. Multi-scale banking techniques help accentuate both local and global features. An example is shown in Figure 2.21. Talbot et al. [TGH11] pointed

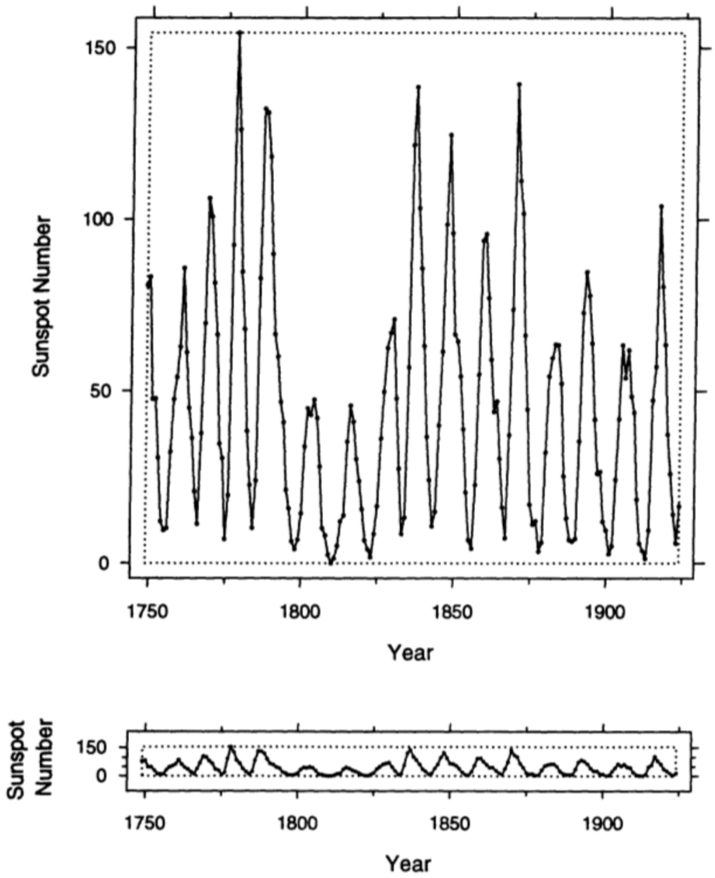


Figure 2.20: The same line plot in two different aspect ratios, showing the number of sunspots over a certain time period. Only if looking at the lower image, we discover that the cycles tend to rise faster than they fall. Image taken from Cleveland [Cle93].

out that the previous methods are not *parameterization invariant*, meaning that the calculated aspect ratio is not always the same regardless of how the line segments are parameterized. They proposed an arc-length (AL) based approach, where the total length of line segments is minimized while area of the plot is kept constant. This method does not explicitly bank to 45° or any other angle. Wang et al. [WWZ+18] compared all previous methods, and expanded Heer and Agrawala's work by introducing L_1-LOR, which is parameterization invariant in certain cases. They proved that L_1-LOR performs better than previously mentioned techniques. Additionally, they elaborated on multi-scale banking by introducing their own *dual-scale banking* technique, and demonstrated that such a technique is very effective for pattern perception.

While *banking to 45°* has held up until today, and is still being developed further, there has been a desire to generalize this approach and find techniques that can be applied to different types of 2D charts, as opposed to just line plots. Fink et al. [FHSW13]

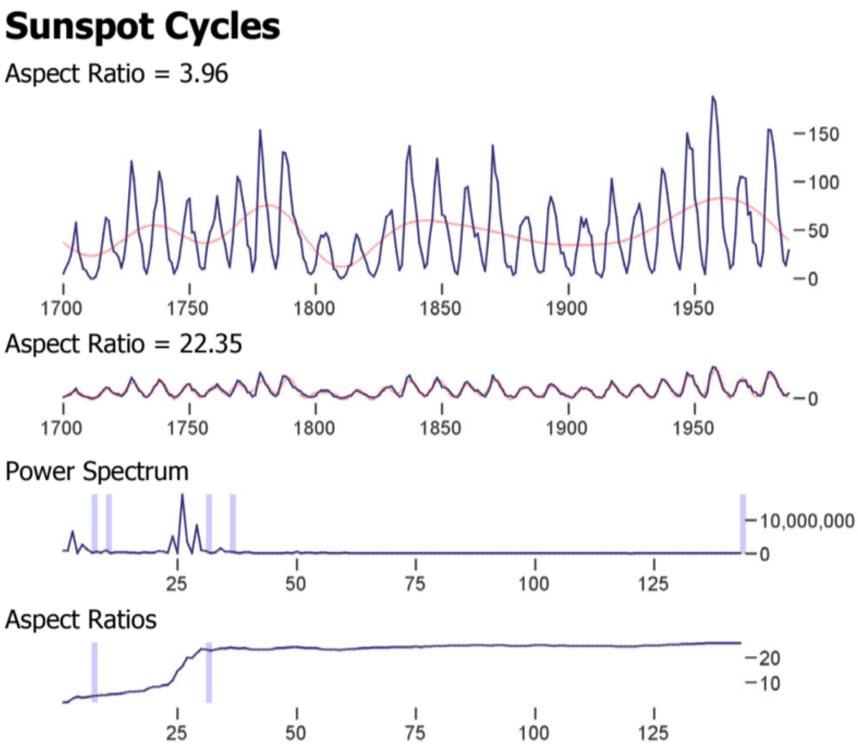


Figure 2.21: Multi-scale banking technique, where several aspect ratios are included to reveal different properties of the data. Image taken from Heer and Agrawala [HA06].

introduced a method for scatter plots, where the aspect ratio is chosen so that the Delaunay triangulation of the plot has certain geometric properties. Ideally, this creates more visible clusters and geometric features in the visualization. Other approaches for scatter plots calculate local polynomial regression (LOESS) curves [CL96] or isolines [TGH11], to essentially create a line plot, and then use one of the established methods. Wang et al. [WWF^{+}19] introduce an image-based approach that is supposed to be independent of the type of plot, using Kernel Density Estimation (KDE).

Research on aspect ratio selection methods for other types of plots, aside from line plots, has not been as common. The results presented in this thesis show a strong impact of the selected aspect ratio on the visual perception of trends and patterns in the data.

2.6.2 Aspect ratio in web visualization and multiple-view systems

Today, an increasingly important area of use for visualization is in web-based visualization and multiple view (MV) systems. Websites need to be properly displayed on various screen sizes, and support both landscape and portrait orientations. This is addressed by *responsive design*, and more specifically, *responsive visualization.* Hoffswell et al. [HLL20] performed an analysis of news articles which feature visualization, and which responsive techniques they use to deal with changes in screen size. The most common, simple methods were *no changes* and *resize* (while retaining all elements), other prevalent approaches were to remove or reposition content.

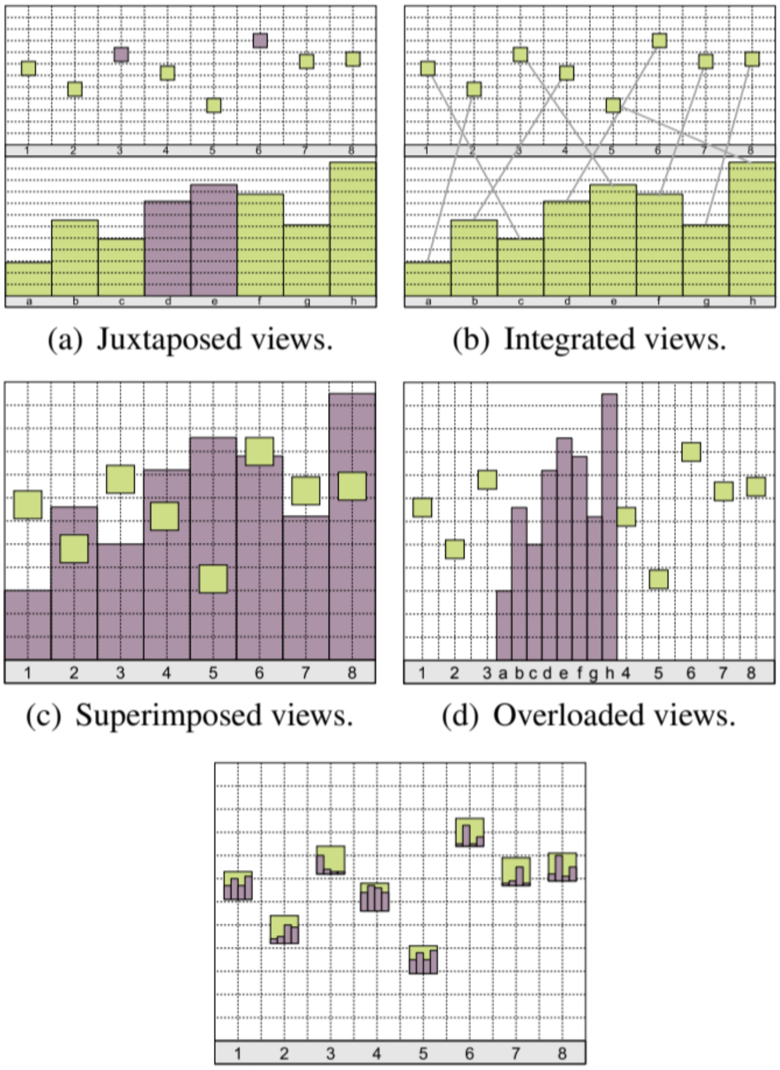
The topic of area judgment, which was discussed in Section 2.5.3, is highly relevant for MV systems. These combine two or more visualizations, which may all be different types of views, into one comprehensive representation, in order to "support the investigation of a single conceptual entity" [WBWK00]. MV dashboards are commonly found online, for example in intelligence or analytics applications [BFAR+22]. So far, we have discussed design approaches only in the context of specific visualization techniques. Dashboards must be consistent and comparable across multiple separate views, requiring their own, complex set of design guidelines [SCB+19]. Several researchers have proposed *dashboard design* frameworks, often focusing on a specific genre of dashboards. Dashboard design guidelines generally emphasize the prevention of visual clutter, a conscious choice of data attributes and elements, grouping of views, and a consistent and organized design [BFAR+22].

Organizing individual elements of an MV system in a way that maximizes usability and effectiveness is a challenging task for designers, and researchers have studied ways to automate this layout arrangement process [LLW+24]. MV tools are also often used in combination with responsive display methods. Simple proportional rescaling or reordering of elements may lead to overlaps or cause content to be too small. This means that MV components each have to be adaptable to different orientations and aspect ratios [ZCH+24].

A common requirement for dashboard applications is an interactive design mode, where users should be able to add, remove and adjust the order of elements according to their needs. However, this presents a conflict with design guidelines and best practices. A trade-off may be a "recommendation system" where a layout is chosen from an existing set of designs based on user input [CZL+21]. This could be used to limit possible aspect ratios to a finite set of values, making it easier for visualization designers to consider the impact of different aspect ratios. Kristiansen et al. [KGB22] use an algorithm to automatically identify potential adjustments to a dashboard, for example, combining or splitting up views. These recommendations are presented to the user after each action, allowing the dashboard to be improved iteratively.

An interesting topic in relation to this are composite visualizations. Rather than being arranged beside each other, like in an MV system, they feature multiple types of *visual structures* in one combined view [JE12]. These may be especially helpful for analytical tasks [DCM+23]. Figure 2.22 shows several types of composite visualizations. While the study of composite visualizations has not been as extensive as research on other techniques, they offer a novel approach to MV visualization, and may provide an interesting perspective for the problem of responsive dashboard layout. For example, two or more views may be either individual or combined, depending on the current available aspect ratio. Different combinations could be suggested to the user using a recommendation system.

Research on treemaps, which was discussed earlier, reveals important details on how users perceive area and hierarchy in rectangular layouts. This should be taken into account if dealing with MV visualizations, since MV systems may require visual comparison of values across different charts.



(a) Juxtaposed views. (b) Integrated views.

(c) Superimposed views. (d) Overloaded views.

(e) Nested views.

Figure 2.22: Types of composite visualizations of a scatterplot (green) and a bar chart (purple). Image taken from Javed and Elmqvist [JE12].

CHAPTER 3

PCP Application

This thesis specifically focuses on the relevance of aspect ratio of PCPs. We aimed to understand how aspect ratio influences the display of parallel coordinates. For this, we implemented a web-based application that lets us render a PCP from a dataset and allows us to export statistical data for further analysis.

3.1 Used technologies

We implemented our own parallel coordinate viewer as a web-based application using HTML/CSS and JavaScript. While many libraries and online tools already exist for generating PCP, there are several reasons for creating a customized solution. Firstly, we wanted to make sure that the PCP can be dynamically resized without significant re-rendering time, while keeping proper axis scale and proportions between elements. Furthermore, to analyze the angles of line segments across different aspect ratios, we needed the ability to access the exact position of each polyline-axis intersection to be able to calculate each angle. We decided on a vector graphics based solution, enabling clean and inexpensive resizing while also allowing us close control over each individual line element [App12].

D3.js [BO24b] is a widely used JavaScript library for creating web-based visualizations. We used the latest version 7.9.0. D3 is considered a low-level library; rather than predefined charts, it provides a set of *primitives* - graphical elements that can be used to construct advanced visualizations. Elements created using D3 can be appended directly to the document object model (DOM) as scalable vector graphics (SVG) objects, making them highly dynamic and responsive.

Statistics.js [Pla17], a simple JavaScript library providing utilities for statistical data analysis, was used for correlation calculations.

The application was implemented using IntelliJ IDEA as an integrated development environment (IDE) and tested in several browsers including Microsoft Edge, Google Chrome, and Mozilla Firefox.

Additional analysis was performed using the spreadsheet application OpenOffice Calc, as well as the R statistical computing language in the RStudio IDE, in combination with the ggplot2 library [Wic16] to help visualize analysis results. Furthermore, Python was used to preprocess datasets.

3.2 Implementation

Our implementation consists of two main components: visualization and analysis. First, we implemented a dynamic and interactive parallel coordinate visualization. Figures 3.1 and 3.2 show screenshots of our tool. In this example, the "80 Cereals" dataset [Cra17] was imported, and the columns *sugars* and *vitamins* are selected for the detail view. The sample with the name *Total Whole Grain* is currently highlighted, which is shown in the output in the top left in Figure 3.1. In the detail view (Figure 3.2), the two highlighted axes are visualized. To the right, statistical properties of angles of all line segments between the dimensions *sugars* and *vitamins* are displayed.

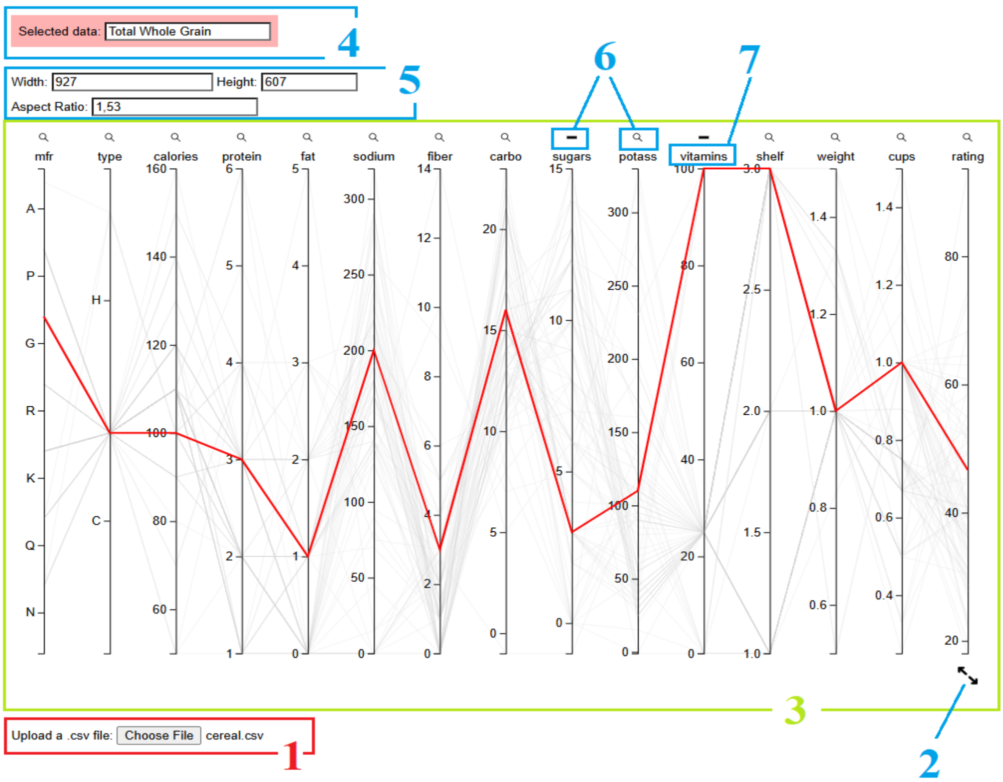


Figure 3.1: The main plot view, showing the entire dataset.

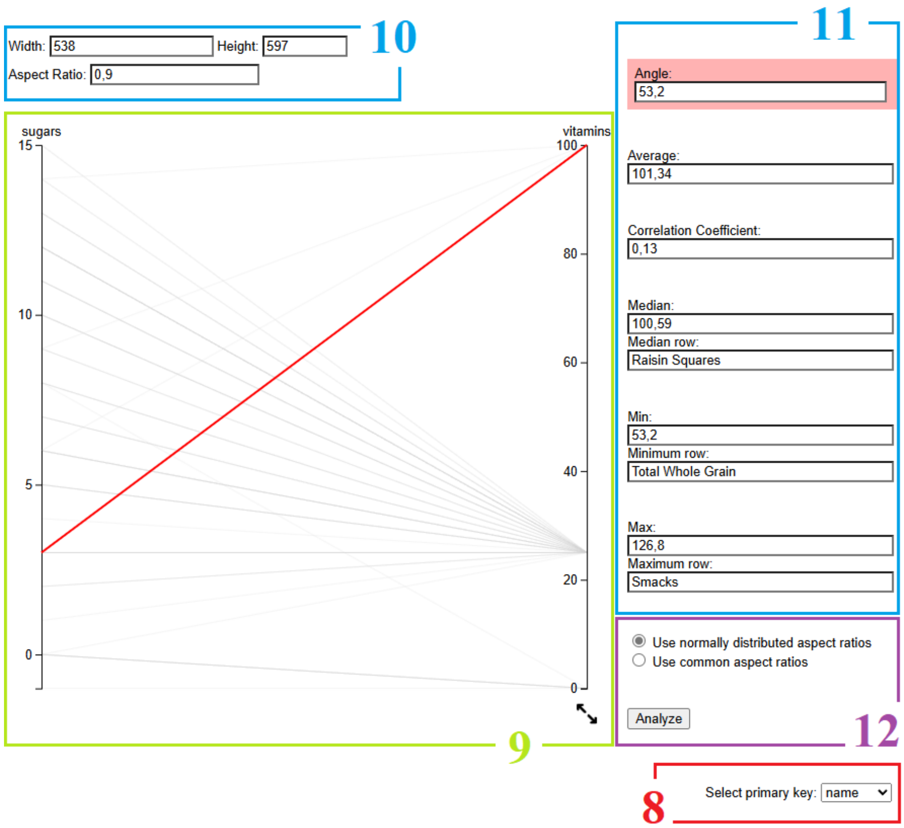


Figure 3.2: The detail view, where two columns from the dataset are shown in focus.

The upload button (marked as 1 in Figure 3.1) can be used to import a dataset in a .csv file format. The entire dataset is then graphed in the main plot window (marked as 3). The application automatically selects the first column of the dataset as the primary key which will be used to identify rows. If this column is not the desired primary key, another column can be chosen in the key select input (2).

The main diagram is interactive. If the user hovers over one of the polylines, the entire sample is highlighted, and the primary key value (such as the name of the sample) is displayed in the data display (4). In the main plot, axes can be reordered by left-clicking the name displayed on top (7) and dragging the axis left or right. Axes can be removed from the plot by right-clicking the name. Currently there is no way to re-add deleted axes, the application will have to be reloaded. Above each axis name, a magnifying glass icon (6) is displayed. The user can click this icon to select axes to be displayed in the detail window (9 in Figure 3.2). Only two axes can be focused at a time. If the detail view is loaded, lines that the user hovers over in either window, will be highlighted in both.

Both views can independently be resized by moving the resize handle in the bottom right corner of each window (8). This “drag-and-drop” resize operation is shown in Figure 3.3. It works similar to resizing a desktop window, with horizontal and vertical guidelines to show the new dimensions of the plot. The plot size is limited to a certain maximum width and height, calculated based on the dimensions of the browser window.

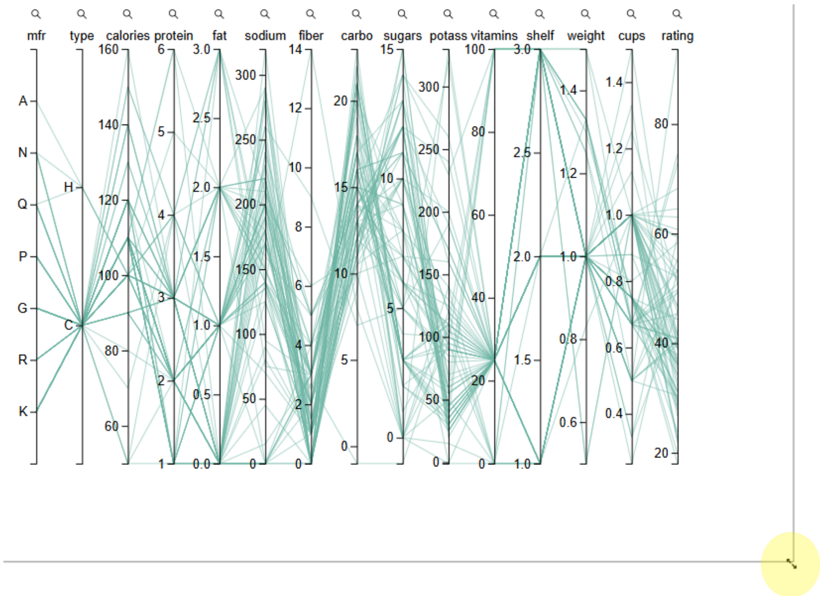


Figure 3.3: Drag-and-drop resize operation in progress, with the mouse location highlighted in yellow. Horizontal and vertical guidelines show the new size of the plot.

The current width, height, and resulting aspect ratio of each view are displayed above the plot windows (5 in Figure 3.1, 10 in Figure 3.2). Width and height can be set manually, by entering a value into the input, the plot will be resized accordingly. It is important to note that we calculate aspect ratio as *width/height*.

The purpose of the detail view is to be able to more closely inspect two specific axes. On the right side in our application (11), several statistical values are displayed. The first field shows the *angle* of the currently highlighted line segment. We measure this as the angle between the line and the first of the two connected axes, marked in yellow in Figure 3.4. A completely horizontal line would have an angle of 90°. Naturally, this angle is dependent on the aspect ratio of the plot.

If the detail view is loaded, and everytime it is resized, we determine all angles of all lines given the current aspect ratio. To calculate the angles, we utilize the same methods we use to draw the x- and y-axes, which utilize functions from the D3 library. Instead of rendering the results, we store the absolute positions of the line on each axis, then pass them to a separate function for calculating the angle based on the two positions.

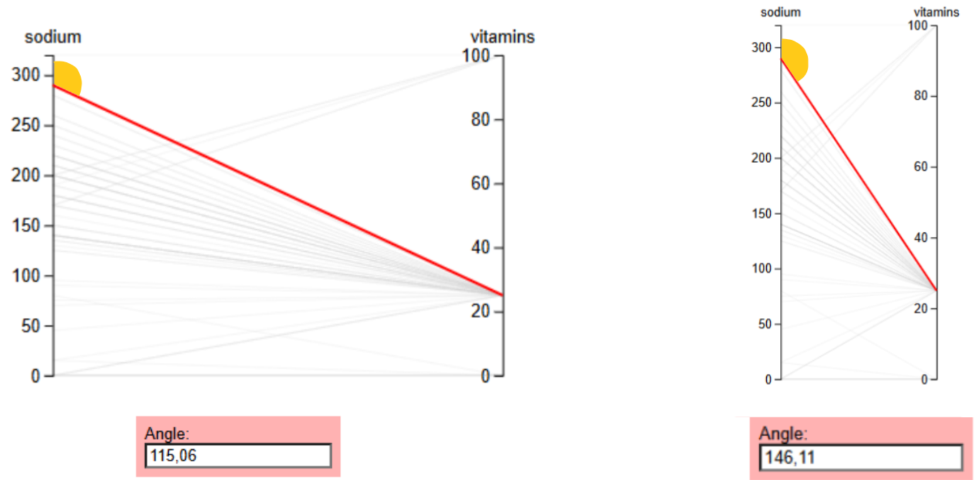


Figure 3.4: Two axes rendered in our application, in two different aspect ratios. We define the *angle* of a line as the top angle between the axis and the line segment connecting the two axes (annotated in yellow).

Once we have a list of angles, we determine statistical values and show them in the corresponding fields (11):

- Average: the mean of all angles
- Min: the minimum angle
- Max: the maximum angle
- Median: the median in the list of all angles.

We also display primary keys of the samples with the minimum/maximum/median angle.

Finally, the field *correlation coefficient* shows the correlation between the two axes, which is independent from the current aspect ratio. If both dimensions are nominal, PCC is used to calculate this value. Otherwise, *Spearman's rho* is used, a rank correlation measure that is also suitable for ordinal scales [Pla17].

Once a dataset has been loaded, a statistical analysis of the data is created. The radio buttons (12 in Figure 3.2) can be used to choose between different predefined sets of aspect ratios to be used for the analysis. If "Use normally distributed aspect ratios" is selected, a list of 16 random, normally distributed values is used. Otherwise, a set of 16 common aspect ratios of images is used [Fir24].

An analysis object, which describes statistical values of two given axes in one specific aspect ratio, is represented as a JavaScript object (Figure 3.5) with the following properties:

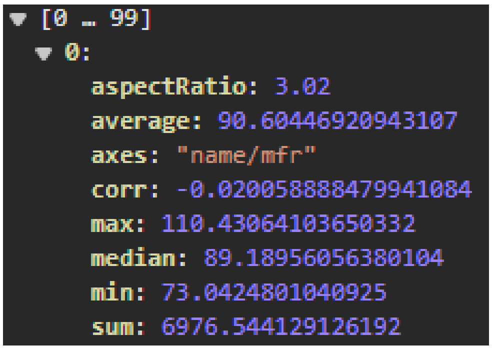
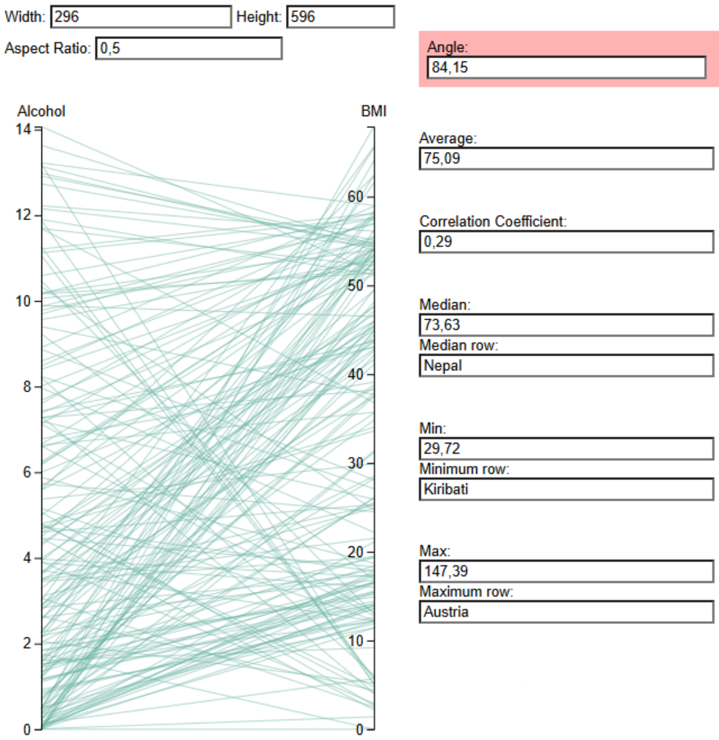


Figure 3.5: Properties of an *analysis* JavaScript object of two axes in one aspect ratio.

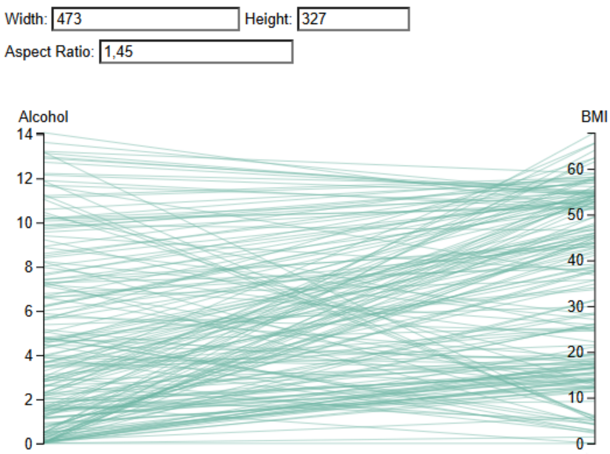
- axes: the names of both axes, given as a string
- aspectRatio: the aspect ratio used for the current calculation
- corr: the correlation coefficient
- average: the average of the list of angles
- min: the minimum value in the list of angles
- max: the maximum value in the list of angles
- median: the median in the list of angles
- sum: the sum of all angles.

We calculate these values for each possible combination of axes and each aspect ratio in the selected predefined list, resulting in the final analysis file containing a list of objects. For a dataset with n columns, an analysis file contains $\binom{n}{2} * 16$ rows. This file can be downloaded in a .csv format by clicking the “Analyze” button (12 in Figure 3.2). These results can then be used for further analysis using different tools.

Figure 3.6a shows how the detail view may be used to explore a dataset. Two dimensions, *Alcohol* and *BMI*, in the “Life Expectancy (WHO)” dataset [Raj18] are selected. In the statistical value output on the right side, we see that these variables have a moderately positive correlation, with a PCC of 0.29. The minimum angle, meaning the line with the largest incline, belongs to the sample *Kiribati*, telling us that this country has the highest BMI relative to alcohol consumption. Vice versa, the country with the maximum angle, Austria, has the highest alcohol consumption relative to BMI. Nepal represents the median in the list of all countries sorted by angle.



(a) The plot at an aspect ratio of 0.5, resulting in a portrait orientation. We can visually identify line segments with noticeably low or high angles, revealing a high relative difference between the two variables for the selected sample (represented by a line). To the right, we see the statistical properties of angles.



(b) The same plot as in (a), resized to a landscape orientation.

Figure 3.6: Two dimensions, *Alcohol* and *BMI*, in the "Life Expectancy (WHO)" dataset [Raj18], visualized in the detail view of our PCP application, with two different aspect ratios.

CHAPTER 4

PCPs and Aspect Ratio

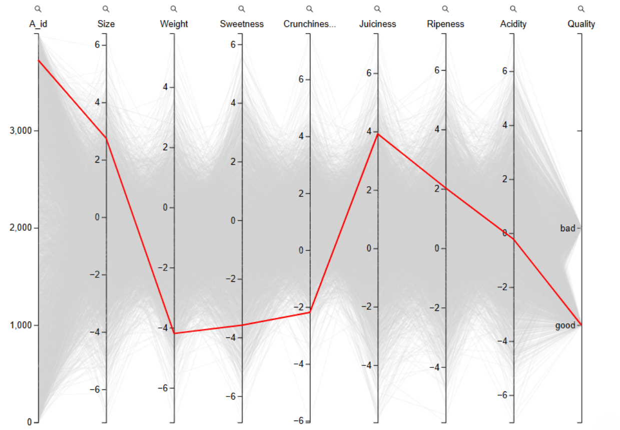
4.1 Statistical analysis of angles

The goal of our analysis was to understand how changing the aspect ratio of a PCP influences the graphical representation. To achieve this, we examined the angles between all line segments representing a sample in a PCP.

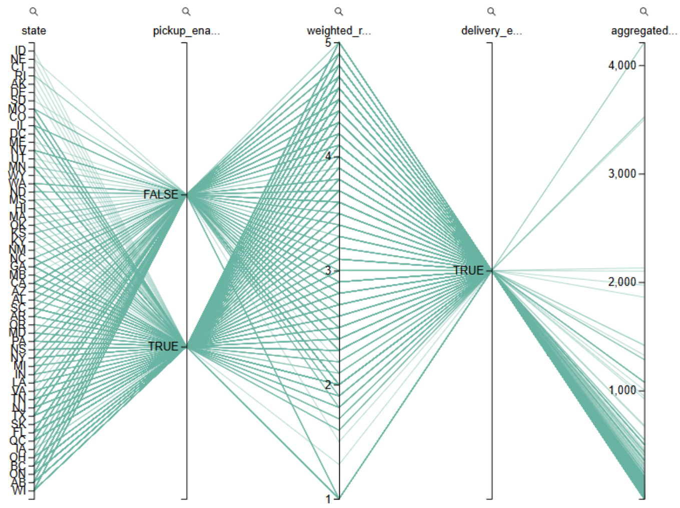
4.1.1 Dataset selection

Publicly available datasets were downloaded from Kaggle [Kag24] and imported into our application. It was found that some datasets were either unsuitable for our analysis purposes, or could not be effectively visualized using our current implementation. Some examples of this were:

- Large, dense datasets such as the "Apple Quality" dataset [Elg24] present the issue of overplotting. This can be mitigated using some of the techniques described earlier, such as interactive highlighting (Figure 4.1a), brushing, or different methods of scaling. With the current implementation, dataset could not be used for analysis as the calculation of so many angles was computationally too expensive.
- Datasets with many categorical dimensions. Usually, not a lot of interesting information could be extracted from the diagram. Especially dimensions consisting of only a few categories, such as boolean data types, caused a lot of line bundling. We can see this in the "1500 North American Restaurants" dataset [Kum24] of which a subset of columns is shown in Figure 4.1b. This dataset even has one column whose value is always *TRUE*. More advanced display methods could help with this, such as brushing, or simply filtering out columns that are not useful, but for the purpose of this thesis, we stuck with the simple method of using a band scale [BO24a] for categorical data.



(a) PCP of the "Apple Quality" dataset [Elg24] with 4000 rows, causing overplotting issues and making statistical analysis expensive.



(b) PCP of the "1500 North American Restaurants" dataset [Kum24] containing several categorial columns, leading to line bundling. Columns of unique identifiers like the *state* column are not suitable for visualization, and can be manually filtered out.

Figure 4.1: Two datasets, which were unsuitable for our application.

Eventually, four datasets were chosen for further analysis.

- The "80 Cereals" dataset [Cra17] which includes nutritional information of 80 different brands of cereals,
- The "Cars 2022" dataset [Tya22] which lists specifications of 199 different car types,
- The "Sleep Health and Lifestyle" dataset [Tha23] where 374 subjects were evaluated for various health metrics, and

- The “Life Expectancy (WHO)” dataset [Raj18] which contains development factors for all countries over several years. Since visualizing the time dimension would require more advanced techniques, a subset of this dataset was created, which only includes a single year (2000).

All four datasets were suitable to be visualized in our PCP tool; they could be rendered without causing significant overplotting issues. Each dataset contained some categorical, but predominantly numerical columns, meaning there was less line bundling. Interesting properties, such as correlations, groups, and outliers, were able to be discovered through visual inspection. We assumed that such datasets would be typical use cases for PCPs in real-world applications.

4.1.2 Statistical Analysis

We wanted to study the relationship between aspect ratio and each of the statistical properties of the line angles. For this we used an unequal variance t-test (Welch’s Test) [Alb24], which is implemented in the *TTEST* function in OpenOffice Calc [Ope24].

The four selected datasets were imported into the PCP tool, and for each, an analysis file was exported. For this analysis, we used normally distributed values of aspect ratios, as a t-test assumes a normal distribution [Alb24]. A combined dataset was prepared, which included five randomly selected columns from each of the four datasets, resulting in 20 * 16 (number of aspect ratios) = 320 different samples. The *TTEST* function was used to compare the *aspect ratio* column with the columns representing each statistical angle value (average, min, max, median, and sum). Each t-test resulted in a p-value very close to 0, shown in Table 4.1. Since each p-value is lower than 0.05, we can conclude that the differences between the means of columns are statistically significant.

Statistical property	P-value
average	1.35E-314
min	1.2E-292
max	1.63E-222
median	8.97E-290
sum	8.48E-112

Table 4.1: T-test of *aspect ratio* and different statistical properties.

We then wanted to find groups of aspect ratios that can be clustered according to their statistical values. For this, an analysis file was exported using the set of commonly used aspect ratios. The analyses of all four datasets were merged into a single file, resulting in a combined length of 8784 samples (549 axis combinations). A multidimensional k-means clustering was performed in R over the dimensions *min, max, average, median,* and *sum.*

The number of clusters was chosen using the *fviz_nbclust* function of the *factoextra* package [Dat20]. The first step was to apply the *elbow method*, this means to calculate the *total within sum of squares* for each k number of clusters and plot the result as a line chart to see if a sharp *elbow point* is present. In our case, this did not yield a clear result. Furthermore, the *average silhouette* and *Calinski-Harabasz* methods were applied, both suggesting three as the optimal number of clusters [Ozt23].

Figure 4.2 shows how the three extracted clusters are distributed over different aspect ratios. Since we calculated aspect ratio as width divided by height, smaller aspect ratios correspond to portrait orientations, while larger ratios represent landscape orientations. We can see that the larger aspect ratios are quite uniform, since only Cluster 3 is available there. For smaller aspect ratios, two clusters can be found (Cluster 1 and 2).

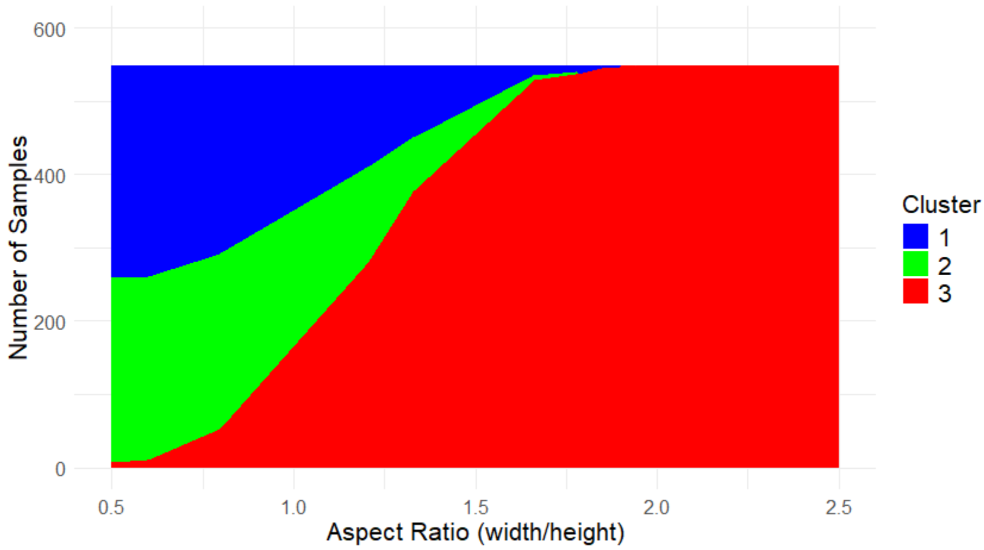


Figure 4.2: Distribution of aspect ratios among clusters. The horizontal axis shows the different aspect ratios featured in our analysis dataset, the vertical axis shows how many samples contain each of the three clusters. A "sample" is defined as a PCP, containing two axes plotted in one specific aspect ratio.

Being aware of the three clusters in the data, we further wanted to see how the individual angle parameters are distributed over the three clusters. An analysis using boxplots can be seen in Figure 4.3. Every plot represents an angle parameter, and boxplots are related to one of the three clusters, respectively.

We saw in Figure 4.2 that Cluster 3 describes almost exclusively large aspect ratios, and we can see that this cluster is the least variable one in regards to all statistical values, whereas the clusters containing smaller aspect ratios are more spread out. We also notice that the distribution of clusters behaves differently between angle parameters. The *sum* shows a distinct distribution from other parameters. Here, the three clusters are very concentrated and contain similar ranges of values, meaning that the *sum* stays the most consistent in different aspect ratios. The *median* displays a very similar result to the *average*, showing that our analysis dataset did not contain many outliers. The minimum and maximum angle also show very similar distributions of clusters. For these four parameters (*average, min, max,* and *median*), we observe that Clusters 1 and 2 are very distinct from Cluster 3, meaning that angles change significantly as the aspect ratio increases.

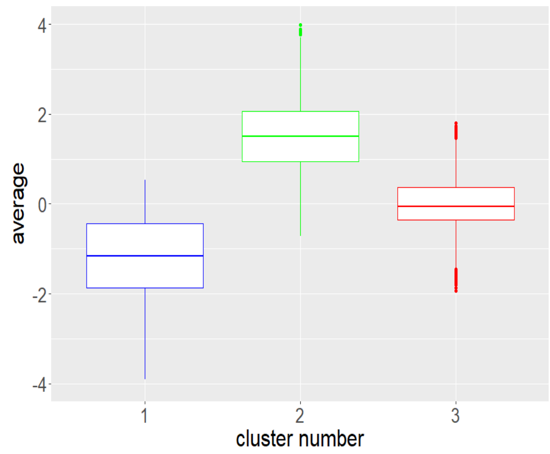
4.2 Results and interpretation

In this thesis, we did not try to measure *effectiveness* or any other aspect of the visual perception process, as this would require a user study which is outside the scope of this work. We aim to understand how much the individual elements making up the plot visually change if applying different aspect ratios, and whether there are certain groups or ranges of aspect ratios where this change is more or less impactful. This knowledge can be used to formulate certain hypotheses about the perception of parallel coordinates that may then be discussed in further works.

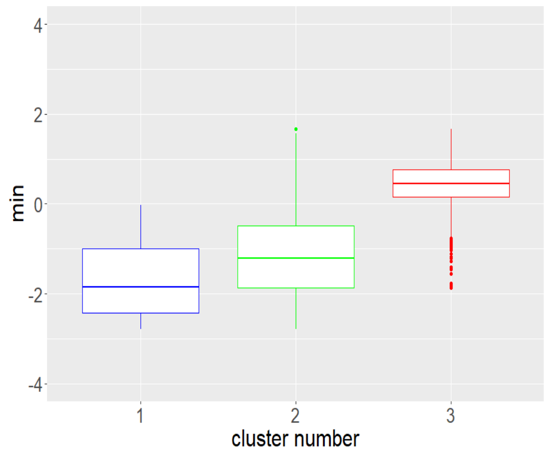
Our statistical analysis indicates that the choice of aspect ratio has a strong influence on the display of PCPs. Since the *angles* of lines between two axes are directly related to the statistical correlation of two dimensions, and judging correlations is one of the most common visual analysis tasks performed in PCPs, we hypothesize that aspect ratio has a significant effect on how PCPs are perceived.

Angles stay more consistent among larger aspect ratios. Landscape-oriented plots with similar aspect ratios will not be much different in terms of statistical distribution of the angle parameters. This is not the case for squared or portrait-oriented plots, where small changes of the aspect ratios can have less predictable effects on the distribution of the angle parameters.

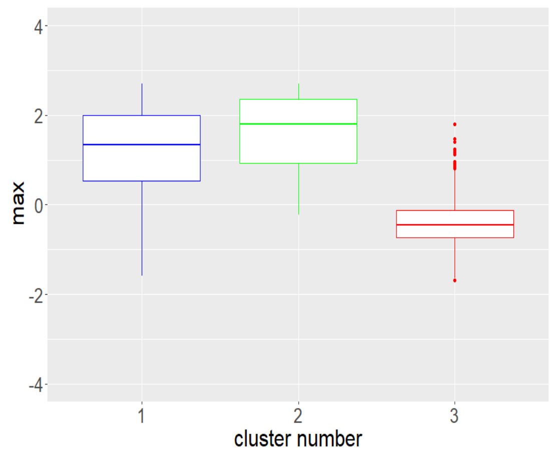
For designers and programmers who want to implement a PCP as part of an information or data visualization application, this means that they should test their layout across multiple aspect ratios to find possible unsuitable configurations. In a responsive, or dynamically resizable parallel coordinate view, it might be a good idea to only allow landscape orientations.



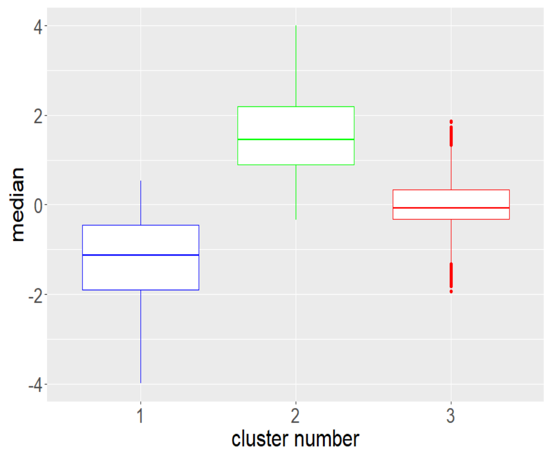
(a) Average angle



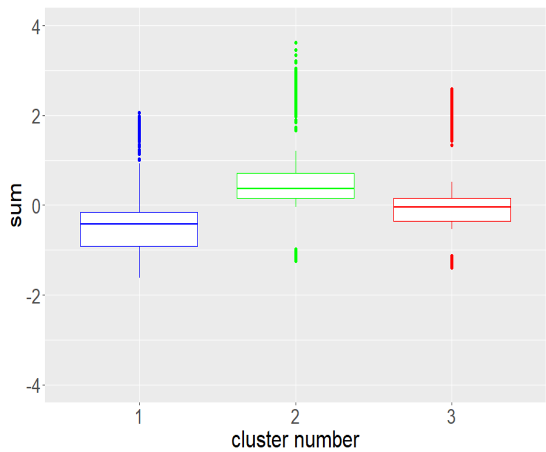
(b) Minimum angle



(c) Maximum angle



(d) Median angle



(e) Sum of angles

Figure 4.3: Clustering of different statistical properties based on our k-means analysis. Cluster 3 (red) describes the group of large aspect ratios.

CHAPTER 5

Conclusion and Future Work

The literature analysis revealed many aspects that we have to take into account for developing a visualization application. We looked at various topics - from a broad overview and history of visualization and different data types, to a comprehensive review of literature on parallel coordinates as well as aspect ratios.

As visualization has evolved, the purpose has shifted from computational applications, towards visual knowledge discovery, with an increasing influence of cognitive sciences. More recently, with the rise of large-scale data collections, a lot of research has focused on multivariate InfoVis techniques which can allow us to comprehend these large heterogeneous datasets.

Today, common uses of visualization include dashboards and web-based systems, which have to fit various screen orientations and have to be responsive, meaning, they should be adaptable to any given width and height. The aspect ratio has been a major topic in visualization research, as it can significantly influence how well information can be perceived in a graphic. Several researchers have shown how an improper choice of aspect ratio can completely hide certain properties of the data, and have proposed different methods for automatically determining an ideal value, for example, different banking techniques for bivariate line- and scatterplots. While these methods are primarily intended for static graphics, they still introduce interesting principles for graphical perception that we may be able to apply to interactive, responsive visualizations as well.

An important takeaway is that size comparisons are not perceived the same way for different types of graphical elements. For example, humans can generally judge the size difference between two lines more accurately than of two different areas. The visualization techniques we looked at were highly diverse in which graphical elements they use. An example of this can be found in the parallel coordinates plot. This display method is quite unique, as it uses its own non-orthogonal coordinate system. The graphical perception models that were formulated about position, area, and angle judgements in

Euclidean space, may not be applicable. Furthermore, in other visualization techniques, data samples are not represented as polylines, but as points, curves, areas, colors, or many other primitives. Since these elements are all perceived differently, we presume that the aspect ratio problem has no easy, universal solution - we must closely study the specific visualization technique we are working with, and understand its individual elements.

In this thesis we wanted to understand the role of aspect ratio specifically in PCPs. We implemented an interactive application that serves as a basis for exploring a PCP, with a strong focus on allowing a quick, dynamic resizing of the plot. This allows the user to experiment with different aspect ratios. Additional interactive techniques include *highlighting* of lines, which enables inspection of individual data and discovery of outliers, as well as *axis reordering*, which allows for a more thorough exploration of correlations and patterns. *Axis deletion* can be used to remove meaningless dimensions and to handle datasets that have too many columns to be displayed effectively. These interactive techniques were chosen as a minimal set of features which would allow effective data exploration without having to individually preprocess each dataset.

While we identified a significant effect of aspect ratio on the display of parallel coordinates plots, we cannot definitively say how different aspect ratios impact the *visual perception process* without conducting user studies. More advanced statistical analysis of the different graphical elements in a parallel coordinate plot could reveal correlations, trends, or clusters among aspect ratios that we might have missed in our analysis.

A user of our application may want to perform a variety of *visual tasks*, such as finding correlations, clusters, or outliers. Future user studies should take these different tasks into account, and evaluate how aspect ratio affects them individually. For example, viewing Figure 3.6, we notice that prominent angles seem to be easier to identify at a portrait orientation. A user study focusing specifically on this task could confirm or disprove this hypothesis.

Our PCP application can be extended with more advanced visualization techniques and additional interactive features. During the literature review, several common features of parallel coordinates tools were identified that may be added. *Brushing* methods seem to be standard for parallel coordinates tools. A common analysis task is to select samples that are within a given range or category in one dimension, and assess how this subset of samples behaves in other dimensions. A similar approach is *bundling*, where polylines are not just highlighted using different colors, but for example reshaped into density-based curves. Once again, a user study could help us understand what role the aspect ratio of a plot plays if working with these techniques. As we discussed earlier, visual comparisons of area are perceived differently than position or angle. This might affect, for example, the use of density-based representations.

Based on our analysis results, we made a suggestion for visualization designers to preferably use landscape-oriented displays, since these were found to be more consistent compared to square or portrait-oriented views. On the other hand, we discussed *responsive* design, especially in relation to multiple-view systems or dashboards, as well as the importance of interactivity in visual data exploration. This presents a challenge for visualization design: how can we allow a user a lot of freedom of customizing position, size, shape, and orientation of graphical elements, while also preventing them from choosing an aspect ratio that might hide important information?

Along with this, in our literature review we found different methods that have been developed for mathematically choosing an ideal aspect ratio, such as *banking* techniques for line plots. While implementation and evaluation of such aspect ratio selection methods were outside the scope of this work, it would be interesting to study to what extent these existing methods can be applied to parallel coordinates, or if new banking techniques can be developed specifically for PCPs. Finally, a remaining question is whether we can combine these selection methods with responsive web- or interface design principles. Further evaluation studies could improve our understanding of how aspect ratio impacts the visual perception process, helping us to develop a comprehensive set of guidelines for visualization designers.

List of Figures

List of Tables

Acronyms

DOM document object model. 31

IDE integrated development environment. 32

InfoVis Information Visualization. 5, 7, 15, 45

MV multiple view. 28–30

PCA principal component analysis. 9, 18

PCC Pearson's correlation coefficient. 17, 35, 36

PCP parallel coordinates plot. 1, 2, 7, 8, 13, 15–22, 31, 37, 39–43, 46, 47, 49, 50

SciVis Scientific Visualization. 5, 7

SVG scalable vector graphics. 31

Bibliography

[AA01] Gennady Andrienko and Natalia Andrienko. Constructing Parallel Coordinates Plot for Problem Solving. In *Proceedings of the 1st International Symposium on Smart Graphics*, SG '01, pages 9–14, Hawthorne, NY, USA, March 21 - 23, 2001.

[ABK98] Mihael Ankerst, Stefan Berchtold, and Daniel Keim. Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In *Proceedings of the 1998 IEEE Symposium on Information Visualization*, INFOVIS '98, pages 52–60, Research Triangle Park, NC, USA, October 19 - 20, 1998. IEEE Computer Society.

[Alb24] Elizabeth A. Albright. Two Independent Samples Unequal Variance (Welch's Test). `https://sites.nicholas.duke.edu/statsreview/means/welch/`, 2024. Accessed 8 Oct, 2024.

[Ama18] Amazon Mechanical Turk, Inc. Amazon Mechanical Turk. `https://www.mturk.com/`, 2018. Accessed 8 Oct, 2024.

[AOS15] Keith Andrews, Majda Osmić, and Gerhard Schagerl. Aggregated parallel coordinates: integrating hierarchical dimensions into parallel coordinates visualisations. In *Proceedings of the 15th International Conference on Knowledge Technologies and Data-Driven Business*, i-KNOW '15, Graz, Austria, 21 - 23 October 2015.

[APM+11] Erik W. Anderson, Kristin C. Potter, Laura E. Matzen, Jason F. Shepherd, Gilbert A. Preston, and Claudio T. Silva. A User Study of Visualization Effectiveness Using EEG and Cognitive Load. *Computer Graphics Forum*, 30(3):791–800, 2011.

[App12] Rachel Appel. Web Dev Report - Working with Graphics on the Web: Canvas vs. SVG. `https://learn.microsoft.com/en-us/archive/msdn-magazine/2012/june/web-dev-report-working-with-graphics-on-the-web-canvas-vs-svg`, 2012. Accessed 8 Oct, 2024.

[BBK+18] Michael Behrisch, Michael Blumenschein, Naam Kim, Lin Shao, Mennatallah El-Assady, Johannes Fuchs, Daniel Seebacher, Alexandra Diehl, Ulrik

Brandes, Hanspeter Pfister, Tobias Schreck, Daniel Weiskopf, and Daniel Keim. Quality Metrics for Information Visualization. *Computer Graphics Forum*, 37:625–662, 2018.

[Ber11] Jacques Bertin. *Semiology of Graphics: Diagrams, Networks, Maps.* ESRI Press, 2011.

[BFAR+22] Benjamin Bach, Euan Freeman, Alfie Abdul-Rahman, Cagatay Turkay, Saiful Khan, Yulei Fan, and Min Chen. Dashboard Design Patterns. *IEEE Transactions on Visualization and Computer Graphics*, pp:1–11, 09 2022.

[BO24a] Mike Bostock and Observable, Inc. Band scale. `https://d3js.org/`, 2024. Accessed 8 Oct, 2024.

[BO24b] Mike Bostock and Observable, Inc. D3 by Observable. `https://d3js.org/`, 2024. Accessed 8 Oct, 2024.

[Cha83] John M. Chambers. *Graphical Methods for Data Analysis.* Chapman and Hall/CRC, 1st edition, 1983.

[Cha06] Winnie Wing-Yi Chan. A Survey on Multivariate Data Visualization. Technical report, Department of Computer Science and Engineering, Hong Kong University of Science and Technology, 2006.

[Che73] Herman Chernoff. The Use of Faces to Represent Points in k-Dimensional Space Graphically. *Journal of the American Statistical Association*, 68(342):361–368, 1973.

[CHM82] William S. Cleveland, Charles S. Harris, and Robert McGill. Judgments of Circle Sizes on Statistical Maps. *Journal of the American Statistical Association*, 77(379):541–547, 1982.

[CL96] William S. Cleveland and Clive Loader. Smoothing by Local Regression: Principles and Methods. In *Statistical Theory and Computational Aspects of Smoothing*, pages 10–49. Physica-Verlag HD, 1996.

[Cle93] William S. Cleveland. *Visualizing Data.* Hobart Press, 1993.

[CM84] William S. Cleveland and Robert McGill. Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods. *Journal of the American Statistical Association*, 79(387):531–554, 1984.

[CMM88] William S. Cleveland, Marylyn E. McGill, and Robert McGill. The Shape Parameter of a Two-Variable Graph. *Journal of the American Statistical Association*, 83(402):289–300, 1988.

[Cra17] Chris Crawford. 80 Cereals - Nutrition data on 80 cereal products. `https://www.kaggle.com/datasets/crawford/80-cereals`, 2017. Accessed 8 Oct, 2024.

[CvW11] Jarry Claessen and Jarke van Wijk. Flexible Linked Axes for Multivariate Data Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17:2310–6, 2011.

[CZL+21] Xi Chen, Wei Zeng, Yanna Lin, Hayder Al-Maneea, Jonathan Roberts, and Remco Chang. Composition and Configuration Patterns in Multiple-View Visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1514–1524, 2021.

[Dat20] DataCamp. fviz_nbclust: Dertermining and Visualizing the Optimal Number of Clusters. `https://www.rdocumentation.org/packages/factoextra/versions/1.0.7/topics/fviz_nbclust`, 2020. Accessed 8 Oct, 2024.

[dB04] Selan dos Santos and Ken Brodlie. Gaining understanding of multivariate and multidimensional data through visualization. *Computers & Graphics*, 28(3):311–325, 2004.

[DCM+23] Dazhen Deng, Weiwei Cui, Xiyu Meng, Mengye Xu, Yu Liao, Haidong Zhang, and Yingcai Wu. Revisiting the Design Patterns of Composite Visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 29(12):5406–5421, 2023.

[d'O85] Maurice d'Ocagne. *Coordonnées Parallèles & Axiales: Méthode De Transformation Géométrique Et Procédé Nouveau De Calcul Graphique Déduits De La Considération Des Coordonnées Parallèles.* Gauthier-Villars, 1885.

[EDF08] 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.

[Elg24] Nidula Elgiriyewithana. Apple Quality - Explore the World of Fruits. `https://www.kaggle.com/dsv/7384155`, 2024. Accessed 8 Oct, 2024.

[Esa20] Vladimir A. Esaulov. The Art of Nomography. `https://kartsci.org/kocomu/computer-history/the-art-of-nomography/`, 2020. Accessed 8 Oct, 2024.

[FHSW13] Martin Fink, Jan-Henrik Haunert, Joachim Spoerhase, and Alexander Wolff. Selecting the Aspect Ratio of a Scatter Plot Based on Its Delaunay Triangulation. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2326–2335, 2013.

[Fir24] Firehouse. Aspect Ratio Cheat Sheet v2. `https://www.wearethefirehouse.com/aspect-ratio-cheat-sheet`, 2024. Accessed 8 Oct, 2024.

[Fri08] Michael Friendly. The Golden Age of Statistical Graphics. *Statistical Science*, 23(4):502–535, 2008.

[FWR99] Ying-huey Fua, Matthew Ward, and Elke Rundensteiner. Navigating Hierarchies with Structure-Based Brushes. In *Proceedings of the 1999 IEEE Symposium on Information Visualization*, INFOVIS '99, Washington, DC, USA, October 24 - 29, 1999.

[Git24] GitHub, Inc. GitHub Docs. `https://docs.github.com/en/enterprise-server@3.10/account-and-profile/setting-up-and-managing-your-github-profile/managing-contribution-settings-on-your-profile/showing-an-overview-of-your-activity-on-your-profile`, 2024. Accessed 8 Oct, 2024.

[HA06] Jeffrey Heer and Maneesh Agrawala. Multi-Scale Banking to 45 Degrees. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):701–708, 2006.

[HB10] Jeffrey Heer and Michael Bostock. Crowdsourcing graphical perception: Using Mechanical Turk to assess visualization design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 203–212, Atlanta, Georgia, USA, April 10 - 15, 2010.

[HK24] Dustin Hayes and Boris Kovalerchuk. *Artificial Intelligence and Visualization: Advancing Visual Knowledge Discovery*, chapter Parallel Coordinates for Discovery of Interpretable Machine Learning Models, pages 125–158. Springer Cham, 2024.

[HLL20] Jane Hoffswell, Wilmot Li, and Zhicheng Liu. Techniques for Flexible Responsive Visualization Design. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 1–13, Honolulu, HI, USA, April 25 - 30, 2020.

[HW13] Julian Heinrich and Daniel Weiskopf. State of the Art of Parallel Coordinates. In *Proceedings of the 34th Annual Conference of the European Association for Computer Graphics - State of the Art Reports*, Eurographics '13, Girona, Spain, May 6 - 10, 2013.

[IML13] Jean-Francois Im, Michael McGuffin, and Rock Leung. GPLOM: The Generalized Plot Matrix for Visualizing Multidimensional Multivariate Data. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2606–2614, 2013.

[Ins85] Alfred Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1:69–91, 1985.

[Ins09] Alfred Inselberg. *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications.* Springer New York, NY, 2009.

[JE12] Waqas Javed and Niklas Elmqvist. Exploring the design space of composite visualization. In *Proceedings of the 2012 IEEE Pacific Visualization Symposium*, PacificVis '12, pages 1–8, Songdo, Korea (South), February 28 - March 2, 2012.

[JFJW09] Sara Johansson Fernstad and Jimmy Johansson Westberg. Interactive Dimensionality Reduction Through User-defined Combinations of Quality Metrics. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):993–1000, 2009.

[JLC07] Jimmy Johansson, Patric Ljung, and Matthew Cooper. Depth cues and density in temporal parallel coordinates. In *Proceedings of the 9th Joint Eurographics / IEEE VGTC Conference on Visualization*, EUROVIS'07, page 35–42, Norrköping, Sweden, May 23 - 25, 2007.

[JWF15] Jimmy Johansson Westberg 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, 2015.

[JWFLC08] Jimmy Johansson Westberg, Camilla Forsell, Mats Lind, and Matthew Cooper. Perceiving Patterns in Parallel Coordinates: Determining Thresholds for Identification of Relationships. *Information Visualization*, 7(2):152–162, 2008.

[JWLJC06] Jimmy Johansson Westberg, Patric Ljung, Mikael Jern, and Matthew Cooper. Revealing Structure in Visualizations of Dense 2D and 3D Parallel Coordinates. *Information Visualization*, 5:125–136, 2006.

[KAF+08] Daniel Keim, Gennady Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer, and Guy Melançon. Visual Analytics: Definition, Process, and Challenges. *Information Visualization, LNCS 4950*, page 154–175, 2008.

[Kag24] Kaggle Inc. Kaggle: Your Machine Learning and Data Science Community. `https://www.kaggle.com/`, 2024. Accessed 8 Oct, 2024.

[KCS08] Aniket Kittur, Ed Chi, and Bongwon Suh. Crowdsourcing user studies with Mechanical Turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 453–456, Florence, Italy, April 5 - 10, 2008.

[Kei02] Daniel Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, 02 2002.

[KGB22] Yngve S. Kristiansen, Laura Garrison, and Stefan Bruckner. Semantic Snapping for Guided Multi-View Visualization Design. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):43–53, 2022.

[KHA10] Nicholas Kong, Jeffrey Heer, and Maneesh Agrawala. Perceptual Guidelines for Creating Rectangular Treemaps. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):990–998, 2010.

[KK96] Daniel Keim and Hans-Peter Kriegel. Visualization Techniques for Mining Large Databases: A Comparison. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):923–938, 1996.

[Kum24] Saket Kumar. 1500 North American Resturants - Explore Culinary Delights: A Curated Guide to North America's Eateries. `https://www.kaggle.com/dsv/7615817`, 2024. Accessed 8 Oct, 2024.

[LHZ16] Liang Fu Lu, Mao Lin Huang, and Jinson Zhang. Two Axes Re-ordering Methods in Parallel Coordinates Plots. *Journal of Visual Languages and Computing*, 33:3–12, 2016.

[LJWC09] Mats Lind, Jimmy Johansson Westberg, and Matthew Cooper. Many-to-Many Relational Parallel Coordinates Displays. In *Proceedings of the 13th International Conference on Information Visualisation*, IV '09, pages 25–31, Barcelona, Spain, 15 - 17 July, 2009.

[LLW+24] Yanna Lin, Haotian Li, Aoyu Wu, Yong Wang, and Huamin Qu. DMiner: Dashboard Design Mining and Recommendation. *IEEE Transactions on Visualization and Computer Graphics*, 30(7):4108–4121, 2024.

[LWW90] Jeffrey LeBlanc, Matthew O. Ward, and Norman Wittels. Exploring N-dimensional databases. In *Proceedings of the 1st IEEE Conference on Visualization*, VIS '90, pages 230–237, San Francisco, CA, USA, October 23 - 26, 1990.

[Mou11] Rida Moustafa. Parallel coordinate and parallel coordinate density plots. *Wiley Interdisciplinary Reviews: Computational Statistics*, 3(2):134–148, 2011.

[Nor06] Chris North. Toward Measuring Visualization Insight. *IEEE Computer Graphics and Applications*, 26(3):6–9, 2006.

[Ope24] Apache OpenOffice. TTEST function. `https://wiki.openoffice.org/wiki/Documentation/How_Tos/Calc:_TTEST_function`, 2024. Accessed 8 Oct, 2024.

[Ozt23] Fatih Emre Ozturk. Unsupervised Learning in R: Determination of Cluster Number. `https://medium.com/@ozturkfemre/unsupervised-`

learning-determination-of-cluster-number-be8842cdb11, 2023. Accessed 8 Oct, 2024.

[PBO+14] Gregorio Palmas, Myroslav Bachynskyi, Antti Oulasvirta, Hans Peter Seidel, and Tino Weinkauf. An Edge-Bundling Layout for Interactive Parallel Coordinates. In *Proceedings of the 2014 IEEE Pacific Visualization Symposium*, PacificVis '14, pages 57–64, Yokohama, Japan, March 4 - 7, 2014.

[PG88] Ronald Pickett and Georges Grinstein. Iconographic Displays For Visualizing Multidimensional Data. In *Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics*, pages 514–519, Beijing, China, 8 - 12 August, 1988.

[Pla17] Matthias Planitzer. Statistics.js Documentation. https://thisancog.github.io/statistics.js/, 2017. Accessed 8 Oct, 2024.

[PPLF24] Christian Partl, Patrick Plaschzug, Dieter Ladenhauf, and Gernot Fernitz. Star Plots: A Literature Survey. https://courses.isds.tugraz.at/ivis/surveys/ss2010/g4-survey-starplots.pdf, 2024. Accessed 8 Oct, 2024.

[PRS15] Jennifer Preece, Yvonne Rogers, and Helen Sharp. *Interaction Design: Beyond Human-Computer Interaction.* Wiley, 4th edition, 2015.

[PVF05] Raquel Pillat, Eliane Valiati, and Carla Freitas. Experimental Study on Evaluation of Multidimensional Information Visualization Techniques. In *Proceedings of the 2005 Latin American conference on Human-computer interaction*, CLIHC '05, pages 20–30, Cuernavaca, Mexico, October 23 - 26, 2005.

[PvW09] A. Johannes Pretorius and Jarke van Wijk. What Does the User Want to See? What do the Data Want to Be? *Information Visualization*, 8(3):153–166, 2009.

[PWR04] Wei Peng, Matthew Ward, and Elke Rundensteiner. Clutter Reduction in Multi-Dimensional Data Visualization Using Dimension Reordering. In *IEEE Symposium on Information Visualization*, pages 89–96, 2004.

[Raj18] Rajarshi, Kumar. Life Expectancy (WHO) - Statistical Analysis on Factors influencing Life Expectancy. https://www.kaggle.com/datasets/kumarajarshi/life-expectancy-who, 2018. Accessed 8 Oct, 2024.

[RTB20] Paulo Raposo, Guillaume Touya, and Pia Bereuter. A Change of Theme: The Role of Generalization in Thematic Mapping. *ISPRS International Journal of Geo-Information*, 9:371, 2020.

[RWK+06] Oliver Rübel, Gunther Weber, Soile Keränen, Charless Fowlkes, Cris Luengo Hendriks, Lisa Simirenko, Nameeta Shah, Michael Eisen, Mark Biggin, Damir Sudar, Jitendra Malik, David Knowles, and Bernd Hamann. PointCloudXplore: Visual Analysis of 3D Gene Expression Data Using Physical Views and Parallel Coordinates. In *Proceedings of the 8th Joint Eurographics / IEEE VGTC Conference on Visualization*, EUROVIS '06, pages 203–210, Lisbon, Portugal, May 8 - 10, 2006.

[SCB+19] Alper Sarikaya, Michael Correll, Lyn Bartram, Melanie Tory, and Danyel Fisher. What Do We Talk About When We Talk About Dashboards? *IEEE Transactions on Visualization and Computer Graphics*, 25(1):682–692, 2019.

[Sch22] Matthias Schonlau. Visualizing categorical data with hammock plots. German Stata Users' Group Meetings 2022 04, Stata Users Group, 2022.

[Shn96] Ben Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE Symposium on Visual Languages*, VL '96, pages 336–343, Boulder, CO, USA, September 3 - 6, 1996.

[SLHR09] Harri Siirtola, Tuuli Laivo, Tomi Heimonen, and Kari-Jouko Räihä. Visual Perception of Parallel Coordinate Visualizations. In *Proceedings of the 13th International Conference on Information Visualisation*, IV '09, pages 3–9, Barcelona, Spain, July 15 - 17, 2009.

[SMM12] Michael Sedlmair, Miriah Meyer, and Tamara Munzner. Design Study Methodology: Reflections from the Trenches and the Stacks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2431–2440, 2012.

[SR06] Harri Siirtola and Kari-Jouko Räihä. Interacting with parallel coordinates. *Interacting with Computers*, 18(6):1278–1309, 2006.

[SST+12] Chad A. Steed, Galen Shipman, Peter Thornton, Daniel Ricciuto, David Erickson, and Marcia Branstetter. Practical Application of Parallel Coordinates for Climate Model Analysis. In *Proceedings of the International Conference on Computational Science*, ICCS '12, pages 877–886, Omaha, NE, USA, June 4 - 6, 2012.

[Ste60] Stanley Smith Stevens. The Psychophysics of Sensory Function. *American Scientist*, 48:226–253, 1960.

[TGH11] Justin Talbot, John Gerth, and Pat Hanrahan. Arc Length-Based Aspect Ratio Selection. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2276–2282, 2011.

[Tha23] Laksika Tharmalingam. Sleep Health and Lifestyle Dataset - Unlock sleep insights with the Sleep Health Dataset. `https://www.kaggle.com/`

datasets/uom190346a/sleep-health-and-lifestyle-dataset, 2023. Accessed 8 Oct, 2024.

[The00] Holger Theisel. Higher order parallel coordinates. In *Proceedings of the 5th International Fall Workshop on Vision Modeling and Visualization*, VMV '00, pages 119–125, Saarbrücken, Germany, November 22 - 24, 2000.

[Tuf01] Edward Tufte. *The Visual Display of Quantitative Information.* 2nd edition, 2001.

[Tya22] Mrinal Tyagi. Cars 2022 Dataset - Complete collection of Cars available to buy in India with their specs. https://www.kaggle.com/datasets/tr1gg3rtrash/cars-2022-dataset, 2022. Accessed 8 Oct, 2024.

[vW06] Jarke van Wijk. Views on Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):421–432, 2006.

[War04] Colin Ware. *Information Visualization: Perception for Design.* Morgan Kaufmann, 2004.

[WBWK00] Michelle Q. Wang Baldonado, Allison Woodruff, and Allan Kuchinsky. Guidelines for using multiple views in information visualization. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '00, page 110–119, Palermo, Italy, May 1, 2000.

[Weg90] Edward J. Wegman. Hyperdimensional Data Analysis Using Parallel Coordinates. *Journal of the American Statistical Association*, 85(411):664–675, 1990.

[Wei19] Daniel Weidele. Conditional Parallel Coordinates. In *Proceedings of the IEEE Visualization Conference*, VIS '19, pages 221–225, Vancouver, BC, Canada, October 20 - 25, 2019.

[Wic16] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis.* Springer-Verlag New York, 2016.

[WL90] Stephen Wehrend and Clayton Lewis. A problem-oriented classification of visualization techniques. In *Proceedings of the 1st IEEE Conference on Visualization*, VIS '90, pages 139–143, San Francisco, CA, USA, October 23 - 26, 1990.

[WLG97] Rainer Wegenkittl, Helwig Löffelmann, and Eduard Gröller. Visualizing the behaviour of higher dimensional dynamical systems. In *Proceedings of the IEEE Conference on Visualization*, VIS '97, pages 119–125, Phoenix, AZ, USA, October 24, 1997.

[WWF+19] Yunhai Wang, Zeyu Wang, Chi-Wing Fu, Hansjörq Schmauder, Oliver Deussen, and Daniel Weiskopf. Image-Based Aspect Ratio Selection. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):840–849, 2019.

[WWZ+18] Yunhai Wang, Zeyu Wang, Lifeng Zhu, Jian Zhang, Chi-Wing Fu, Zhanglin Cheng, Changhe Tu, and Baoquan Chen. Is There a Robust Technique for Selecting Aspect Ratios in Line Charts? *IEEE Transactions on Visualization and Computer Graphics*, 24(12):3096–3110, 2018.

[ZCH+24] Wei Zeng, Xi Chen, Yihan Hou, Lingdan Shao, Zhe Chu, and Remco Chang. Semi-Automatic Layout Adaptation for Responsive Multiple-View Visualization Design. *IEEE Transactions on Visualization and Computer Graphics*, 30(7):3798–3812, 2024.

[ZF98] Michelle X. Zhou and Steven K. Feiner. Visual task characterization for automated visual discourse synthesis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '98, page 392–399, Los Angeles, CA, USA, April 18 - 23, 1998.

[ZS21] Boyan Zheng and Filip Sadlo. On the Visualization of Hierarchical Multivariate Data. In *Proceedings of the 14th IEEE Pacific Visualization Symposium*, PacificVis '21, pages 136–145, April 19 - 21, 2021.