UNIVERSITÉ DE FRIBOURG
UNIVERSITÄT FREIBURG

HUMAN-IST
HUMAN CENTERED INTERACTION SCIENCE & TECHNOLOGY

Enhancing the visualization of mixed multidimensional data in Parallel Coordinates

Raphaël Tuor[1]

Supervisors:
Prof. Dr. Denis Lalanne[2]
Dr. Florian Evéquoz[3]

December 14, 2016

Department of Informatics - Master Project Report

Département d'Informatique - Departement für Informatik • Université de Fribourg - Universität Freiburg • Boulevard de Pérolles 90 • 1700 Fribourg • Switzerland

phone +41 (26) 300 84 65 fax +41 (26) 300 97 31 Diuf-secr-pe@unifr.ch http://diuf.unifr.ch

Accepted for: R. Tuor, Enhancing the visualization of mixed multidimensional data in Parallel Coordinates. *Human-IST Research Center Documents Factory*, 1(1):1–2, December 2016.

[1] raphael.tuor@unifr.ch, Human-IST Research Center, University of Fribourg
[2] denis.lalanne@unifr.ch, Human-IST Research Center, University of Fribourg
[3] florian.evequoz@unifr.ch, Human-IST Research Center, University of Fribourg

Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank:

Prof. Dr. Denis Lalanne for his useful comments, his guidance, and his enthusiasm throughout the learning process of this master thesis.

Dr. Florian Evéquoz for his technical advice in our numerous meetings, and his motivation.

Pierre Vanhulst for his pertinent insights, and his technical advice.

Michael Papinutto for his great help with regard to statistical tests.

My family, my girlfriend and my close friends for their continuous support and motivation throughout the whole project.

Abstract

The goal of this master thesis is to improve the user performance for low level tasks, such as frequency- and similarity-based tasks, and a high level configuration task, on mixed multivariate datasets. We assessed several methods to enhance the visualization of mixed multidimensional data in Parallel Coordinates. This type of data is very common, and its visualization in standard Parallel Coordinates can lead to incorrect insights. Many visual enhancement techniques are available for mixed multidimensional data, but only a few user studies can confirm their efficiency. We proposed two new approaches and tested them in two controlled user experiments.

In the first study, we compared three variants of Parallel Coordinates in common data analysis tasks: standard Parallel Coordinates, *Parallel Bubbles* and Parallel Sets. We showed that adding a visual encoding of frequency improves significantly the user performance, and that *Parallel Bubbles* offer a good compromise in terms of user performance. We also showed that the perception of visual channels is a key factor to take into account when choosing a visual encoding.

The second study compared the effectiveness of a perpendicular axis layout, *Stacked Coordinates*, with Parallel Coordinates, in configuration and data analysis tasks. We showed that with Parallel Coordinates, the users find more insights, and that their performance is significantly better in a frequency task implying two dimensions and in a visual mining task. Both visualization methods are suited for frequency tasks focused on one dimension. *Stacked Coordinates* seem to offer a better performance with configuration tasks, but we should conduct a user study on a larger scale in order to get more robust results.

At the end of this thesis, we describe the ELSA tool, a case study that makes use of the *Stacked Coordinates*. ELSA is an exploration tool for sustainable architecture that was developed in collaboration with the EPFL and the EPFL+ECAL Lab.

Keywords: Master thesis report, Human-IST Research Research Center, Data visualization

Chapter 1

Introduction

Mixed multidimensional datasets are present in many scientific fields, like engineering, finance, life sciences and astronomy [23]. They mix several types of dimensions that can be categorical or continuous, non-ordered or ordinal, temporal or spatial, or logical. Nowadays, data can be gathered at low cost [27] and much higher speed than previously. Datasets combining a large variety of dimensions, along with constantly growing population sizes, have given rise to new opportunities for data analysts, and new challenges for scientific visualization.

Each type of dimension holds a different type of information. For example, continuous dimensions can define performance or distance, and categorical dimensions can define products, clients, or building references, grouping items into categories. A continuous dimension gives a sense of magnitude, allowing to quantify the degree of similarity between two data items. In contrast, a category gives a sense of identity – no standard measure of similarity exists for categorical values [14].

Despite these fundamental differences, categorical dimensions are often mistakenly presented in the same way as the continuous ones, with an ordering that does not give any information about the data [31]. Even worse: when used for categorical dimensions, visual encodings of magnitude can lead to incorrect interpretations and hypotheses, and produce visual clutter. Representing non-ordered categories with a magnitude channel creates a visual ranking, and suggests that the topmost values are more important than the ones below. One of the goals of this thesis is to compare the visual encodings suited for categorical dimensions, and to measure the effective impact of several encodings on the user performance.

When using a visualization method, the user has two distinct goals: analysis, and presentation [37, 52]. Munzner's framework [37] makes a clear distinction between these two aims: analysis refers to the discovery of new knowledge from the data, and presentation refers to the use of a visualization method to convey a message to an audience. This thesis focuses on the analysis tasks.

Simulation-generated data are an example of mixed multivariate data. They include several discrete dimensions and one, or several, continuous output dimensions that indicate the performance of a solution. Stored in the form of "flat tables" [37], mixed data are used in contexts that involve design tasks, like exploration, visual mining, and configuration tasks. Such tasks require the user to assess similarities between dimensions and items, and the distribution of data items in a continuous dimension.

The high amount of dimensions requires the use of more advanced visualization techniques than 2- or 3-dimensional datasets. Several problems arise from the visualization of this kind of data, such as clutter, overplotting and values ordering. These problems need to be addressed in order to reduce the user's visual confusion and to allow him to achieve data analysis tasks more efficiently. Many visual enhancement techniques are available for such data, but only a few user studies can confirm their efficiency [29].

In this Master thesis, we present our research about the techniques to improve the visualization of mixed multivariate data. We define the main data analysis tasks, and enumerate the best ways to reduce overplotting and clutter. We underline the importance of following the expressiveness principle [37] when choosing a channel to represent a dimension. We focus our attention on the methods aimed at enhancing the visualization of mixed multidimensional data in Parallel

Coordinates, and at improving the user performance in configuration tasks.

We developed two new visual approaches. The first, *Parallel Bubbles*, enhances visually the representation of categorical dimensions in Parallel Coordinates, and enhances the user performance in basic data analysis tasks. Our second approach, *Stacked Coordinates*, is an alternative to Parallel Coordinates meant to be used in design tasks. It should help the user to focus on a quantitative objective, and to quickly find a valid solution in an underconstrained problem. We lead two separate user experiments in order to test the potential of both visualization methods. We also demonstrate the potential of the *Stacked Coordinates* in a case study.

1.1 Goals

The goal of this master thesis is to improve the user performance for low level tasks, such as frequency- and similarity-based tasks, and a high level configuration task, on mixed multivariate datasets.

To achieve this, we have to determine and understand the problems raised by multivariate mixed data, list the most meaningful visualization methods and visual encodings, choose the best suited visual encodings, and test our hypotheses in controlled user experiments.

1. Parallel Coordinates [24] are a mature and efficient visualization method for multidimensional data, but they implement a continuous design model that does not suit categorical dimensions. Parallel Sets [31] were developed for categorical data, and are not designed to represent continuous dimensions. Categorical dimensions produce overplotting, hiding the frequency information that would help to achieve basic data analysis tasks. Therefore, our first hypothesis is the following: *"Parallel Bubbles (i.e. frequency-enhanced Parallel Coordinates) are a good compromise in terms of user performance in frequency and similarity tasks, when compared to standard Parallel Coordinates and Parallel Sets."* In order to verify this hypothesis, we conducted a study to test the effectiveness of adding a visual encoding of frequency to categorical dimensions. Three variants of Parallel Coordinates were compared: standard Parallel Coordinates, *Parallel Bubbles*, and Parallel Sets.

2. Mixed multidimensional data are used in many domains that comprise design tasks, like architecture and industrial design. The goal of the second user study is to verify if our new approach, *Stacked Coordinates*, outperforms Parallel Coordinates in a set of design tasks. The second hypothesis is the following: *"Stacked Coordinates, using a perpendicular axis layout, increase the user performance in configuration tasks when compared to Parallel Coordinates."* *Stacked Coordinates* are based on the principle of Parallel Coordinates, but represent each data item as a dot instead of a polyline. The position of dots on the X axis is determined by their performance. The use of a perpendicular axis layout is meant to give the user a clear view on the output range of each category. The representation of each item with a dot instead of a polyline is meant to reduce the visual clutter, increase the overview and reduce the need for interaction. With a different perspective on the data, using this method should also give other insights than Parallel Coordinates.

This Master project should help towards improving the knowledge about mixed data visualization enhancement methods. The two user experiments give interesting outcomes about the effectiveness of visual encodings in data analysis tasks performed on mixed multivariate data.

To reach these goals, we had to understand and classify existing visualization methods for mixed multidimensional data, and to determine the best visual encodings to improve the user performance in a cluttered view. We also had to determine a new layout to improve the user performance in configuration tasks. The visualization should give a quick overview on the range of outputs for each dimension value, and reduce overplotting problems.

In the next section we provide a description of the general structure of this thesis.

1.2 Structure

The **first part** of this thesis (chapters 2, 3, 4.1 and 4.2) consists in our literature review. We structure it using the three-part analysis framework defined by Munzner [37]:

- *what* is the type of data to visualize.
- *why* is the task being performed.
- *how* is the vis idiom constructed in terms of design choices.

For the *what*, we define the characteristics of mixed multivariate data. For the *why*, we define a taxonomy of the tasks that the user performs on the data, and we explain the most common tasks. For the *how*, we list and compare the visualization methods and visual encodings that are available for mixed multivariate data.

In the **second part** (chapter 5), we summarize the problematic and explain the design choices that lead to the two visualization prototypes. We also explain the technological choices.

The **third part** (chapter 6.3 and 7) describes our two controlled user experiments:

1. The first controlled experiment consists in testing our first hypothesis. We compare three variations of Parallel Coordinates in similarity- and frequency-based tasks – the result is that *Parallel Bubbles* offer a good trade-off in performance between the two other existing methods.
2. In the second controlled experiment, we attempt to verify if the users of the *Stacked Coordinates* achieve a better performance in configuration tasks than the users of jitter- and color-enhanced Parallel Coordinates, in an underconstrained design space. We also compare the type of insights provided with the *Stacked Coordinates* and the Parallel Coordinates approach.

The **fourth part** (chapter 8) is a case study of the ELSA project. ELSA is a real-world application of the *Stacked Coordinates* method, intended to be used in the context of architectural design.

We conclude with the **fifth part** (chapter 9): we discuss the results and outcomes of this Master project, and suggest the next steps for future developments.

Chapter 2

Mixed multidimensional data

Today, scientific data is gathered at much higher speeds and lower costs [27], through computer simulations and sensors of various kinds. These datasets usually consist in a mix of several types of dimensions. Mixed datasets combining categorical and continuous dimensions are common in diverse fields such as astronomy, finance, physics, engineering and medicine, for example. These datasets are usually stored in flat tables: each row represents one data item and each column is a dimension of the dataset [37]. In the next sections we describe categorical and ordered dimensions, as well as mixed data.

2.1 Categorical dimensions

Categorical (or nominal) dimensions only have a discrete amount of values. They convey a sense of *identity* [37], allowing to distinguish if two values are the same or different (e.g. apples vs oranges). They play a very important role in the analysis of many real-world data sets: "users working with data often examine different classes before even looking at numbers." [31] Nominal dimensions do not hold any underlying order: they can be ordered, but only according to an auxiliary (quantifiable) information, such as price, or degree of preference. Examples for categories are race, genre, shape or color. In the remainder of this thesis, we refer to unique values of each categorical dimension as *levels* or *categories*.

2.2 Ordered dimensions

Ordered dimensions do have an implicit ranking (e.g. "shirt size"). They are of two sub-types: *ordinal*, and *continuous*.

- *Ordinal* dimensions do only have a discrete amount of values. They share many common points with categorical dimensions, except that they have an implicit order. For example, the "shirt size" dimension is ordinal: its values are S, M, L, XL and have an logical order. Ordinal dimensions do not allow to measure the magnitude between values: it is not possible to subtract two shirt sizes.
- *Continuous* (also called *quantitative*) dimensions are a subset of ordered dimensions [37]. They have a potentially infinite number of values. Continuous dimensions imply a measurement of magnitude "that supports arithmetic comparisons". For example we can compare and subtract two heights, lengths or two distances.

2.3 Mixed data

Mixed datasets are defined as containing at least one categorical dimension and one continuous dimension. As we already mentioned, this type of data is present in many fields.

For example, mixed datasets are used in architecture, in the early design stages of a building [30]. The environmental impact of the building can be simulated using a set of features (parameters) representing the building, and a Building Performance Simulation (BPS) software computing the resulting energy performance. Most parameters defining the building consist of categorical dimensions (e.g. window-to-wall ratio, type of glazing, HVAC system), and the remaining consists of continuous dimensions (environmental indicators) that define the performance of the building over its life cycle (LCA). Such datasets comprise a large set of design alternatives (a few thousands), with each alternative representing a simulated building. When the designer explores the database, he can only choose from a discrete set of predefined values for each parameter. The figure 7.1 shows such a database represented with the Parallel Coordinates method. Each axis is a dimension, and only the rightmost axis represents a continuous dimension (the performance indicator).

The visualization of mixed datasets with regular techniques like Parallel Coordinates and Scatterplot matrices implies several discrepancies in the user's mental model. Such datasets require special care in order not to convey false information: the expressiveness principle is often violated by using the same encoding for two different types of dimensions. Each dimension type requires a specific visual encoding in order to convey the information it holds.

Data analysts are often interested in information that is not directly expressed by the data itself, such as the similarity between two categorical dimensions. This additional information requires the use of specific visual encodings. As we explain in chapter 4.2, these encodings are not all equally effective for all the types of dimensions, and should be chosen carefully.

2.4 Conclusion

Categorical and ordered dimensions convey completely different information. Categorical data "should not be shown in a way that perceptually implies an ordering that does not exist" [37], and ordered data "[...] should be shown in a way that our perceptual system intrinsically senses as ordered." [37] Mixed datasets are common, and their characteristics require to choose the visual encodings carefully, depending on the type of information (discrete, continuous) and its importance for the user. The importance of the information is determined by the tasks that the user wants to achieve. Therefore, it is crucial to define the tasks that the user wants to perform. In the next chapter, we explain the main tasks that data analysts perform on mixed datasets.

Chapter 3

A taxonomy of user tasks

In this chapter we define a taxonomy of user tasks: we break down the tasks performed by data analysts into the smallest possible entities. Identifying the smallest level of granularity in these task is a crucial step in order to select the best visual encodings.

3.1 Actions

We break down the tasks performed by users with the taxonomy established by Munzner [37]. She describes three levels of actions that characterize goals.

3.1.1 High-level choices

High-level choices are the overall actions performed by the user on the data. They consist in *consuming* data in order to discover patterns, *presenting* the data to an audience, or simply *enjoying* the visualization.

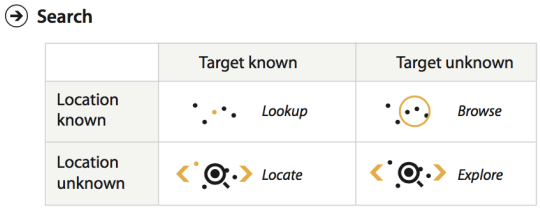


Figure 3.1: The mid-level actions. From [37]

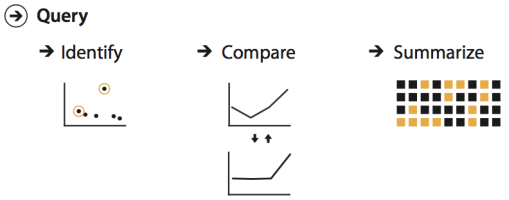


Figure 3.2: The low-level actions. From [37]

In data analysis tasks, the first step is to *consume* existing data in order to *present* his insights thereafter. The need to *produce* something (i.e. annotate, record, or derive) with the visualization is secondary.

3.1.2 Mid-level choices

The kind of search that the user wants to perform defines mid-level choices. These depend on whether the user knows the identity and location of the target (i.e. the data item) or not.

These choices are:

- *lookup*: both location and identity of the target are known. Example: identify a car in the visualization, knowing all its characteristics.
- *browse*: the location of the target is known, but not its identity. Example: find an eco-friendly car model.
- *locate*: the identity of the target is known, but not its location. Example: find a car model by its ID number.
- *explore*: the identity and location of the potential target are unknown. Example: explore the cars dataset to gain new knowledge.

In similarity, frequency and configuration tasks, the *target* is typically unknown – the goal of the user is to find a target, i.e. a pattern, or a set of dimensions that fits some constraints. The user does not already know the exact characteristics, i.e. the identity, of the solution. Thus, the user wants to *explore* and to *browse* the data, rather than to *look up* something specific whose location is already known, or to *locate* a known target at an unknown location.

Here is a more detailed description of what *browsing* and *exploring* the data mean:

- ***Browsing*** means that the user is searching for "one or more items that fit some kind of specification, such as matching up with a particular range of attributes" [37]. Configuration tasks are an example of browsing: for example, the user assesses the output value of all the cars that have between 6 and 8 cylinders, in order to find cars that are above the minimum threshold.
- ***Exploring*** means that the user starts with an overview of the whole dataset and dimensions, and then searches for interesting characteristics. As Munzner [37] states, several examples include searching for outliers in a scatterplot, or "for spikes or periodic patterns in a line graph of time-series data". This task is related to Shneiderman's Visual Information Seeking Mantra [48] – he describes the visual interaction process as : Overview first, zoom and filter, then details-on-demand.

3.1.3 Low-level choices

Low-level choices occur when one or several interesting targets have been found in the data. They represent what kind of *query* the user needs to perform on data items.

A query consists in:

- *identifying* one target.
- *comparing* several targets.
- *summarizing* a full set of targets.

For example, in a Parallel Coordinates plot representing the cars dataset, the user may want to *identify* a single car model. He may also want to *compare* two cars regarding their MPG performance. He also might want to *summarize* a subset of cars (e.g. from one brand) to determine what is the average MPG performance across the subset – a synonym for this task is *overview* [37].

3.2 Targets

Targets can be defined as any finding of interest that the user makes in the data. The goal of the user is to find targets that are categorized into three groups (see figure 3.3) [37]:

Figure 3.3: Illustration of the three targets in data analysis tasks. From [37]

- *Trends*: characterization of patterns in the data. Examples are increases, decreases, peaks, troughs and plateaus.
- *Outliers*: items that do not fit well with patterns.
- *Features*: any structure of interest, depending on the context.

Dimensions are specific properties that are visually encoded. A target can include one or many dimensions.

Targets involving one dimension

Finding an individual value for a given dimension represents the lowest-level target. For a given dimension, finding the extremes, i.e. the minimum and maximum values, as well as the distribution of values (frequency-based tasks), are frequent targets of interest [37].

Targets involving several dimensions

When several dimensions are involved in the target, the target falls into one of these three types:

- *Dependencies*: the values of the first dimension directly depend on the values of the second.
- *Correlations*: the values of the first dimension tend to be tied to the values of the second one.
- *Similarity*: a quantitative measurement calculated on all the values of pairs of dimensions, and ranking the dimensions regarding their similarity from each other.

3.3 Frequency and similarity tasks

Fernstad and Johansson [14] state that the main tasks performed by data analysts are *frequency* and *similarity* tasks.

1. *Frequency* tasks involve **one or more dimensions**. The uses assesses the distribution of values for a specific dimension. They correspond to the lowest-level target that we describe in section 3.2. The *frequency* tasks consist in finding "[...] the relative number of items belonging to specific categories or combinations of categories". They convey an information of frequency, which is of great interest: frequency information is "[...] the main property of focus in categorical data visualization." [14] An example of a frequency task is to count the number of data items belonging to a given category.
2. *Similarity* tasks imply **two or more dimensions**. They correspond to the *correlation* and *similarity* targets that we described above. They consist in assessing the degree of correlation between a pair of dimensions, or at identifying patterns or clusters among the data defined in terms of similarity. Similarity is the strength of a correlation between two dimensions: examples of similarity measures are the string edit distance or the Manhattan distance [7].

3.4 Conclusion

Starting with the highest-level choices, we defined the tasks of interest for this thesis: we focus on the *consumption* of data, in order to *browse* items (i.e. that match with some criteria) and *explore* items and find interesting patterns. Depending on their number, these are then *identified*, *compared* and/or *summarized*. The findings based on these tasks are categorized into *trends*, *outliers*, or more generally *features*. Depending on the amount of dimensions they include, they consist in *distributions*, *similarities* or *correlations*.

In summary, we defined the two most common tasks that data analysts perform as frequency- and similarity-based tasks.

In the two next chapters, we conduct a literature review about the visualization methods for multidimensional mixed datasets. We also define the current state of the art of visual enhancement methods for mixed data.

Chapter 4

State of the Art

4.1 Literature review: visualization methods for multidimensional data

In this literature review we establish a research framework about the visualization of mixed multidimensional data. In this part, we give an overview on the current state of the art of visualization methods for multivariate mixed data in the form of flat tables. Based on this knowledge, we will be able to develop our hypotheses.

Whereas basic visualization methods such as dot and line charts, stacked bar charts, pie charts and scatterplots are appropriate for data with up to 3 dimensions, more advanced techniques are required in order to visualize multivariate data. Unfortunately, human vision "is not multidimensional" [25]. Therefore, to be understandable, the visualization of multivariate data requires the use of techniques such as projection, dimensional reduction, or linking [52]. Interaction techniques like data filtering, or dimensional reordering, help the user to reduce his cognitive load and to identify patterns. We are limited to 2D displays in order to present the information. Small multiples, that is, several instances of the same graphic representing subsets of the data, are often used to represent high dimensional data. But as we will see, visualization methods are limited in the size of the population and in the number of dimensions that they can display.

In this chapter, we give an overview on the strengths and limitations of existing techniques. We start by defining visual encodings, the basic building blocks of graphical displays.

In the next chapter, we address the main methods to reduce problems arising from the visualization of multivariate mixed datasets.

Visual encodings

Munzner [37] summarizes visual encodings (even the most complex ones) in two concepts: **marks** and **channels**. **Marks** are geometric elements that represent data items or links between them. Some examples of marks are: dots, lines and squares. **Channels** are visual functions that control the appearance of marks, like the size of the dots, their position, or the color of a link.

She underlines the importance to use the right visual encoding for the right data. "The same data attribute encoded with two different visual channels will result in different information content in our heads after it has passed through the perceptual and cognitive processing pathways of the human visual system." [37] Using the right visual encoding allows the data analyst to execute his tasks faster and to gather meaningful insights.

Two principles, *expressiveness* and *effectiveness*, guide the choice of a channel for a given type of data:

1. *Expressiveness* principle: the visual encoding should express "all of, and only, the information contained in the dataset attributes" [37].

2. *Effectiveness* principle: "the importance of the dimension should match the salience of the channel". That is, the most important dimensions should be encoded with the most effective

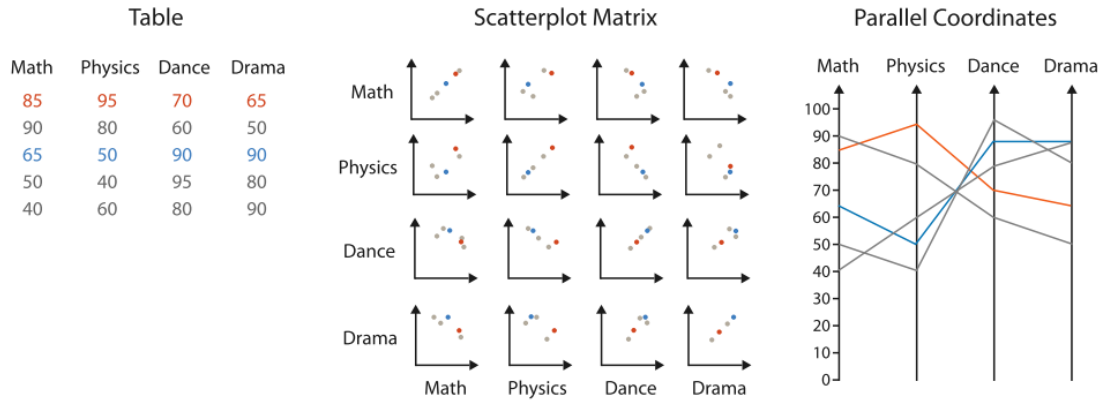


Figure 4.1: Visualization of a small multidimensional dataset in a scatterplot matrix and Parallel Coordinates. From Munzner [37]

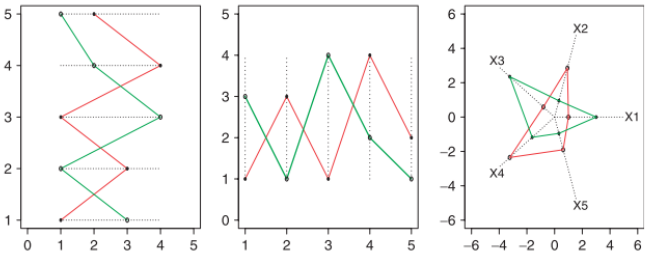


Figure 4.2: Three layouts of Parallel Coordinates, with different spatial axis orientations. The color hue channel is used to distinguish between the two observations. From [36]

channels, and decreasingly important dimensions can then be matched with the less effective ones.

Following the expressiveness principle, the channel that fits best categorical dimensions is the *identity channel*: it does not represent categories in a way that perceptually induces an ordering. The identity channel gives information about *what* something is or *where* it is.

A continuous ordered dimension should be represented with a *magnitude channel*. It convey the information that our perceptual system should sense: an ordering, and an idea of *how much* there is of something.

In the next section we review the main visualization methods for multivariate data.

4.1.1 Parallel Coordinates

Parallel Coordinates [24, 25] are the most popular visualization method used to represent multivariate mixed data. They are applied to a large set of multidimensional problems [42] in fields like life sciences, engineering [23] and finance. In practice, such datasets contain up to about 10-15 dimensions [31]. Moustafa [36] states that Parallel Coordinates are becoming an "essential tool for visualizing hyperdimensional numerical data from almost all real life applications [...]".

Standard Parallel Coordinates use a parallel layout for axes: the horizontal spatial position is used to separate axes, and the vertical spatial position is used to express the value along each aligned axis. Each multidimensional data item is displayed as a polyline intersecting all the axes. Three possible layouts of Parallel Coordinates (vertical, horizontal, and star) are visible in figure 4.2 – they are displaying a 5-dimensional dataset composed of two items. The star layout is the best in case there are inliers in the data: "homogenous records [...] appear as distinct star shape" [36].

A connecting line between two entities shows the relationship between two axes in an explicit way. This is especially pertinent to spot trends [37] and can be used to identify an individual data item. The lines connecting the dimensions allow to identify correlated dimensions: a high positive correlation is characterized by a set of parallel segments, while a negative correlation is characterized by a set of segments crossing over at a single point. However, this only allows pairwise comparison between neighboring axes. Visible patterns depend on the ordering of axes; this requires the user to test all possible configurations of axes, which gets time consuming as the number of dimension increases.

In small-sized populations, outliers can be easily identified. Parallel Coordinates can also assist in the identification of highly discriminant dimensions [36], and are helpful in data mining tasks such as the identification of clusters and the exploration of class properties.

Parallel Coordinates are highly scalable in terms of the number of quantitative values that can be represented on a single axis [37]. In contrast, the number of items is limited to a few hundreds, without the use of any enhancement method. The number of dimensions is limited to a dozen.

Discrepancies in the user's mental model appear when the dimensions include several types (categorical, ordinal or continuous) and the cognitive load increases when the amount of dimensions gets too large. We address these problems in the next chapter.

Expressiveness and effectiveness

Regarding the **expressiveness** principle, using the position on the vertical axis (a magnitude channel) for continuous and ordered dimensions is pertinent. Each axis represents the values in an ascending order. This allows to assess the distance between values, and select a range of values.

For categorical dimensions, however, using the magnitude channel violates the expressiveness principle. The position of values on the axis implies an order which is not directly tied to the data – usually, categories are ordered alphabetically. The use of an identity channel (like color hue or shape) would be more adapted.

Regarding **effectiveness**, dimensions are represented from the left to the right. The position on the X axis, a magnitude channel, is highly effective and ensures a good discriminability [37]: many dimensions can be displayed and the user will still be available to distinguish among them.

The position on the X axis can encode the importance of the dimensions regarding the output value [36]. Thus, the function that defines the ordering of dimensions should be explicit to the user and adapted according to the task.

The color hue, an identity channel, is also very effective [37]. It should encode an information depending on the goal of the user. If the goal of the user is to locate a target, the color hue should be used to distinguish each data item, as it is the case in figure 4.2. One can colour the polylines in two ways, with reference to a continuous dimension: one can either use a color gradient to encode the value of the continuous dimension (magnitude channel) or use a color hue to encode the class of each polyline regarding the threshold value (identity channel). The color coding of polylines helps the user to assess the output performance of each item or its class (valid or non-valid), respectively.

Variants of Parallel Coordinates

Variants of Parallel Coordinates solve some of the problems we mentioned. Fua et al. [15] propose the **hierarchical Parallel Coordinates** method (see figure 4.3). It uses "interactively controlled aggregation" [37], increasing the scalability to hundreds of thousands of items. A hierarchical clustering of the items is performed, and a band of varying width and opacity represents each cluster. The colors of clusters represent their proximity in the cluster hierarchy. The level of detail can be chosen interactively with a slider.

Kosara et al. [31] proposed the Parallel Sets (see figure 4.4), an approach that uses frequency-based techniques to represent categories. This removes the discrepancy between the user's mental model and the visualization of the data. The levels are represented by boxes that are scaled according to their corresponding frequency.

Schonlau [44] developed a similar approach named Hammock plots (see figure 4.5). It is also adapted to categorical and continuous data, representing them as interval data. The lines are replaced by "rectangles that are proportional to the number of observations they represent".

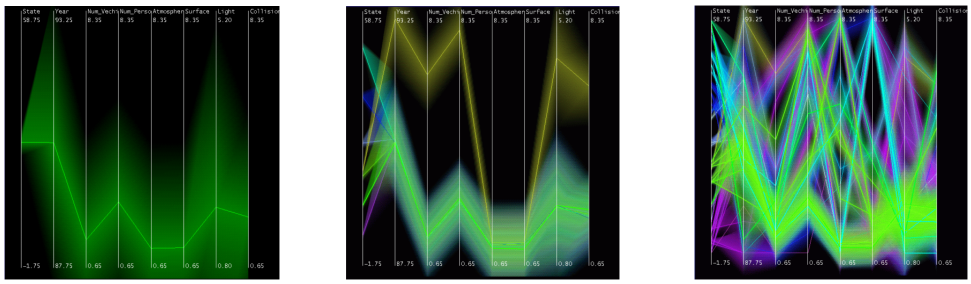


Figure 4.3: Hierarchical Parallel Coordinates displaying the Fatal Accident data set of 230,000 data elements at different level of details. The first plot shows a cut across the root node. The last image shows the cut chaining near the leaf nodes. From [15].

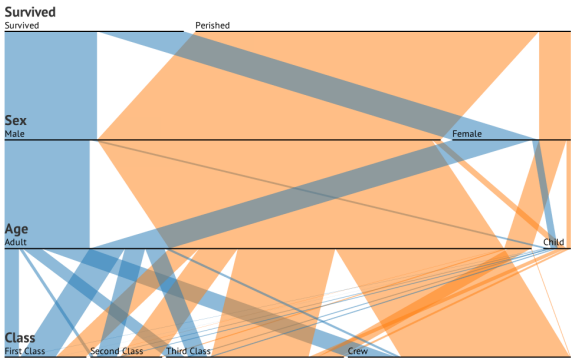


Figure 4.4: Parallel Sets, an example of CatViz. The Titanic dataset is represented. From [2].

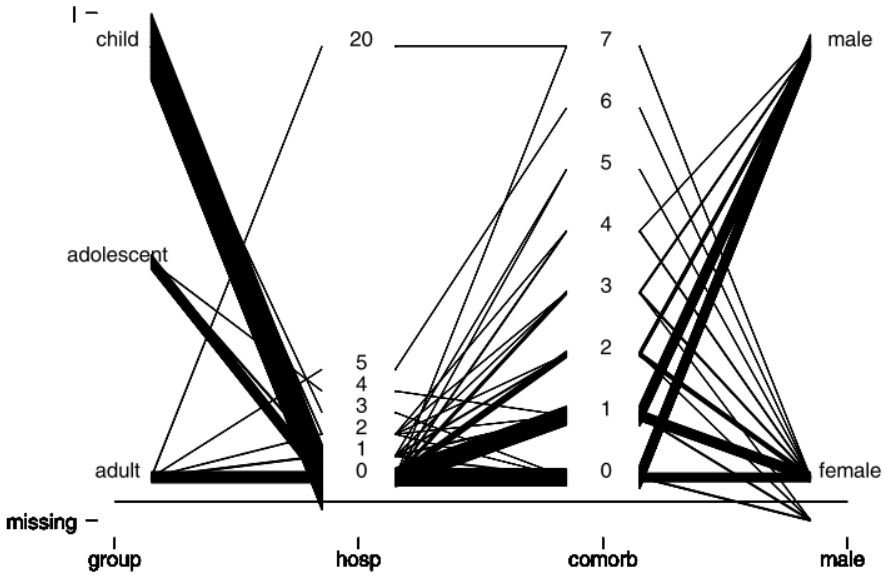


Figure 4.5: Hammock plot representing a dataset of 4 dimensions. The width of a rectangle is proportional to the number of observations it represents. From [44].

4.1.2 Trellis plot

The trellis plot consists in a rectangular array of two-dimensional plots, also called small multiples. The general idea is to represent subsets of the data based on a condition (or filter) for one or more dimensions. For a categorical dimension, one plot per category (level) is created. For example, in figure 4.6, one plot is created for each level of the categorical dimension "Site". In the case of a continuous dimension, subsets will be defined in function of intervals of that dimension (e.g. the dimension "Yield" in figure 4.6). As with Parallel Coordinates, a categorical dimension can be encoded with the color hue (e.g. the dimension "Year" in figure 4.6). Several displays can be produced by a trellis plot: scatterplots, scatterplot matrices, bar charts and histograms.

By showing several subsets of the data, visual clutter is reduced and the structure of the data is revealed. For example, figure 4.7 shows the cars dataset, filtering the data according to the car type.

Its main limitation lies in the number of attributes it can display: up to 3 categorical attributes, and 1 quantitative attribute [37]. The use of linking between several trellis plots could help circumvent the problem.

Expressiveness and effectiveness

The use of magnitude channels for categories does not respect the **expressiveness** principle. Moreover, it can generate overplotting with larger datasets. Adding a color coding to the dots for one categorical dimension (identity channel) removes the need for an additional split. The values of the said dimension are combined into one single plot, enhancing the expressiveness of the visualization. This also helps distinguish patterns and outliers, as it is the case in figure 4.6: the level "University Farm" of the dimension "Site" appears to be an anomaly. In this example, the ordering has a specific meaning that helps detect patterns: levels of the categorical dimensions "Site" and "Variety" are ordered by their median regarding the "Yield" dimension. For the continuous dimension, expressiveness is respected with the use of the magnitude channel.

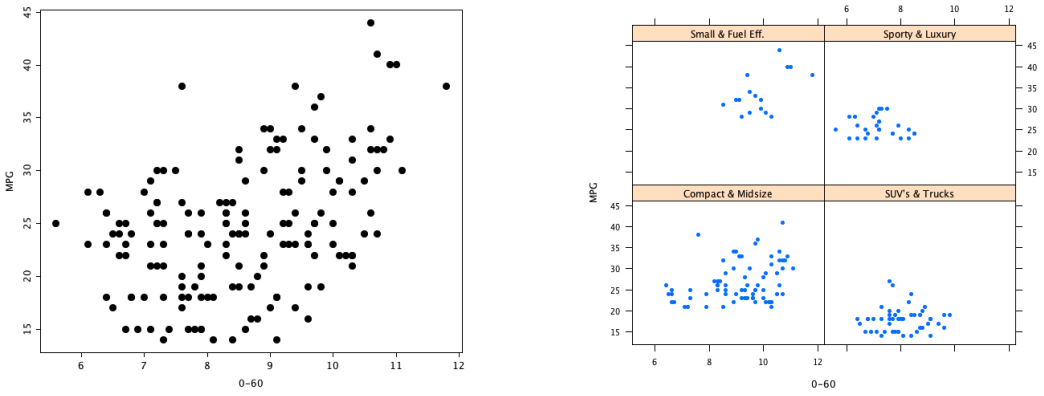


Figure 4.7: Scatterplot (left) showing the cars dataset, with two continuous dimensions: the Y axis shows the fuel efficiency (MPG), and the X axis shows the 0-60 mph acceleration. Trellis plot splits this data according to the car type. From [3]

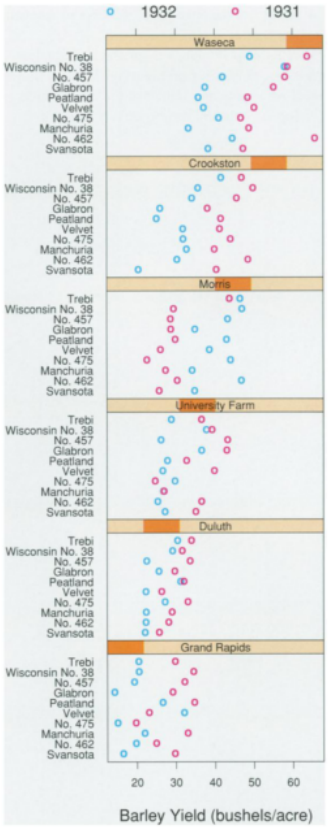


Figure 4.6: Trellis plot using color hue (an identity channel) for the two levels of the categorical dimension "Year" [6]. Several views are created by partitioning the categorical dimension "Site". "Variety" uses the Y axis, a magnitude channel. The X axis encodes the continuous dimension "Yield".

4.1.3 Scatterplot matrix

Scatterplots [20], are basic building blocks in statistical graphics and data visualization [42]. The scatterplot represents each record by a point on a cartesian plane. The scatterplot matrix produces an array of scatterplots, each of which showing a pair of dimensions [12]. It uses the approach of

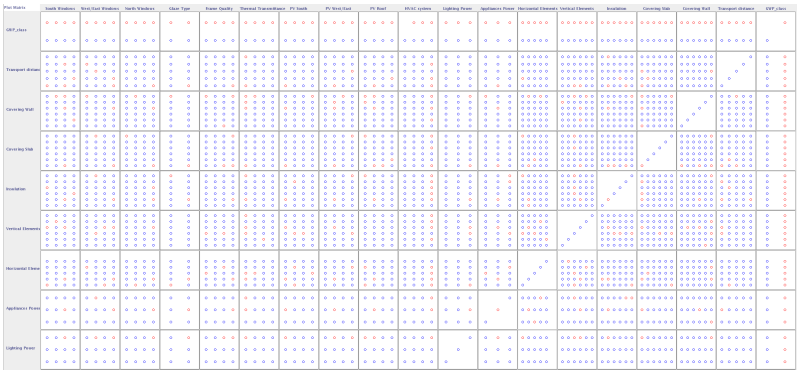


Figure 4.8: Fragment of a scatterplot matrix representing a 19-dimensional simulated building dataset of 9690 items. All dimensions are categorical, and encoded with a magnitude channel. The identity channel encodes the two levels of an additional categorical dimension (i.e. valid and non-valid design alternatives). Overplotting occurs: many dots are plotted on top of each other, hiding the frequency information.

small multiples, i.e. plotting multiple versions of subsets of the data. It is useful to find correlations, trends, clusters and outliers [37].

Scatterplot matrices are designed for continuous dimensions – with categorical data, major overplotting occurs (see figure 4.8). Adding jittering (i.e. spherical noise) to an observation can alleviate the overplotting problem at some extent [45] – the amount of jittering is limited by the screen size.

Even for small amounts of dimensions, scatterplot matrices require a large grid of cells: "With k variables, there are k(k - 1)/2 pairs [...]" [42]. Considering a 10-dimensional dataset, the amount of cells is thus 45 – and implying a high mental load for the user. As stated by Chen et al. [12], the multiplicity problem becomes critical for more than 25 variables: due to the high amount of dimensions, the identification of patterns in scatterplots becomes really impractical. Figure 4.1 illustrates this problem and shows the advantage of Parallel Coordinates even for low-dimensional data.

This problem might be circumvented by using features selection algorithms in order to select a minimum subset of dimensions (subset selection) and filter out less relevant dimensions, thus reducing the amount of scatterplot pairs to show. Such approaches can use sensitivity analysis, or information gain ratio methods.

Harrison et al. [19] recently expanded on previous work by studying the perception of correlations in Parallel Coordinates compared with eight other visualization techniques: scatterplots, stacked areas, stacked lines, stacked bars, donuts, radar charts, line plots, and ordered line plots. 1687 participants took part in the test, using a crowdsourcing platform. The task was to judge the strengths of different correlations. Their results are in agreement with the work from Li et al. [29], in that scatterplots depict correlations better overall than Parallel Coordinates.

Expressiveness and effectiveness

Scatterplot matrices work well with quantitative dimensions. The magnitude channel encodes the values with the spatial position.

The color channel can be used for one categorical dimension, and the size channel can be used to encode an additional quantitative dimension.

4.1.4 Conclusion

In the current state of the art, it is impossible to determine which visualization method is optimal for multivariate mixed data. The visualization methods that we reviewed all present limitations, either in the level of dimensionality that they can handle, in the data size, or in the variety of dimensions. As we explain at the end of the chapter 4.2, the overall understanding of when to use

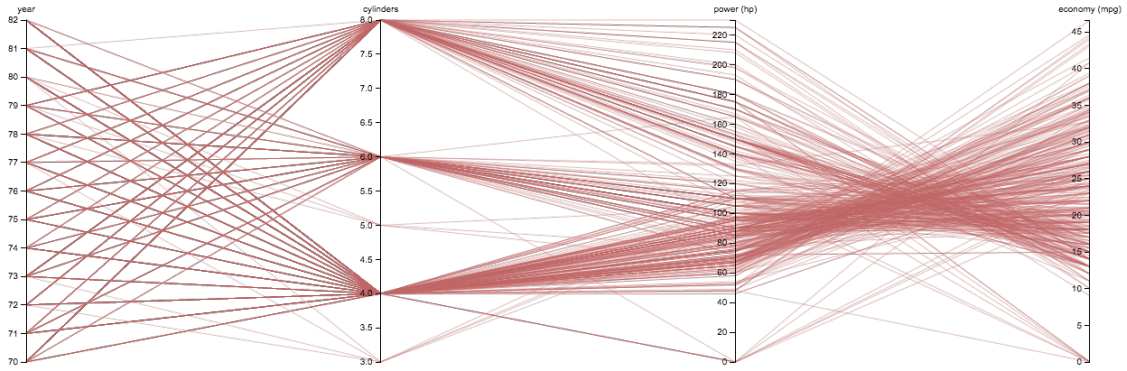


Figure 4.9: Standard Parallel Coordinates plot representing the cars dataset. The presented data consists of 2 discrete, ordered dimensions (*year* and *cylinders*), and 2 continuous dimensions (*power (hp)* and *economy (mpg)*). No visual enhancement is added.

a specific visualization technique to detect a particular pattern is still low, and further studies are required in this area.

Over the visualization methods that we listed, Parallel Coordinates look promising to represent mixed multidimensional data. With the add of visual encodings, it can handle hundreds of thousands of items, and a dozen of dimensions. For this reason, in the remainder of this thesis, we focus on the ways to enhance visualization of mixed multivariate data in Parallel Coordinates.

In the next chapter we address the visual building blocks of Parallel Coordinates more in depth. We list the best ways to encode the relevant aspects of mixed multivariate data, and the techniques to circumvent visual clutter.

4.2 Visual encoding and enhancement methods

The visual confusion that arises from Parallel Coordinates can be reduced with the use of adequate visual encoding and interaction methods. This chapter addresses two important issues of Parallel Coordinates: the visual clutter and the representation of categorical dimensions.

The first section focuses on visual clutter reduction methods. As we already mentioned, Parallel Coordinates do not scale well: even a few thousand of records (see figure 4.10) can make the visualization heavily cluttered. Categorical dimensions are a source of clutter too: they produce overplotting due to the limited amount of values that each dimension can take, masking the frequency information.

The second section focuses on reducing the visual incoherence arising from non-ordered categorical dimensions. The visual incoherence occurs because the non-ordered categorical dimensions are represented using the exact same visual channel as continuous (and ordered) dimensions. There is a discrepancy between the user's mental model of categorical dimensions and their visual representation. It is a source of visual confusion, possibly leading to wrong insights: [...] a ranking is imposed on the visual mapping transformation, influencing perception of the data". [31] Figure 4.9 illustrates this confusion, with a Parallel Coordinates plot displaying a mixed multivariate dataset consisting of 2 categorical (ordered) dimensions, and 2 continuous dimensions: there is no clear way to distinguish whether or not the categorical dimensions are ordered.

4.2.1 Visual clutter

Parallel Coordinates have a low visual scalability [36] and clutter problems can arise depending on the characteristics of the dataset.

Clutter happens when the amount of polylines is so large that it makes it difficult to identify a single polyline and to find any cluster or patterns. Figure 4.10 illustrates an effective use of Parallel Coordinates, allowing to see patterns, and a non effective use of Parallel Coordinates, cluttered by a large dataset. The clutter problem arises from the one-to-one mapping strategy used by

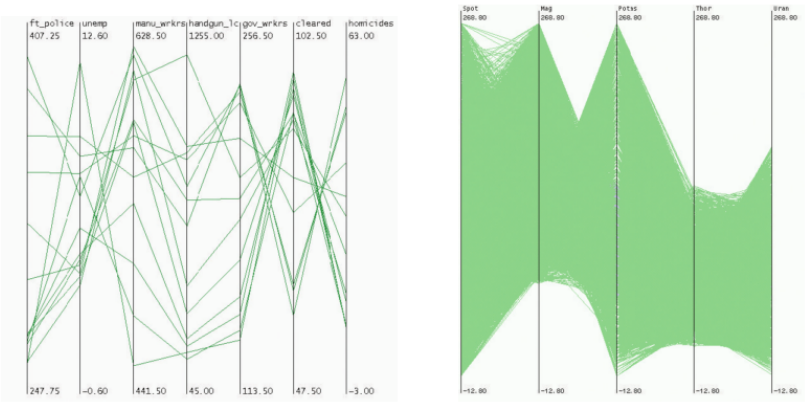


Figure 4.10: At the left, effective use of Parallel Coordinates (13 items, 7 dimensions), where correlations are visible. At the right (over 16'000 items and 5 dimensions), heavy clutter precludes perception of any data trends or correlations. From [15].

Parallel Coordinates: each single record in the original data space is mapped to the visualization space. This results in "[...] a severely cluttered plot when the number of data points exceeds few thousand [...] which is a very small size dataset by today's data standards" [36]. In Parallel Coordinates, "lines need more ink than points such that the total mass of data appears larger in Parallel Coordinates than in scatterplots" [23]. The main factors that influence clutter are:

- the screen resolution
- the dataset size
- the number of values in each dimension

Clutter affects negatively the detection of useful insights and patterns [36]. It can also hide correlations between variables or make identification of an individual polyline impossible.

Dealing with large number of points in Parallel Coordinates is a challenge. Moustafa [36] states that "exceeding the resolution limits results in heavy overplotting and the plot becomes ineffective".

4.2.2 Clutter reduction methods

According to [23], clutter reduction approaches can be grouped in filtering, aggregation, spatial distortion and dimensional reordering. One should also mention the use of the α channel and an opacity function to highlight pixels with high frequency. A suggested approach for categorical dimensions is the use of the color channel. We describe these techniques in the next sections.

Research on Parallel Coordinates "has primarily focused on making the technique less sensitive to visual clutter by reducing the number of polylines or by reducing, or reordering, the parallel axes." [29] Supporting direct user interaction, for example with brushing (a filtering method) or dimensions reordering, can reduce the amount of information shown and user's cognitive load.

Data-driven and screen-based approaches

Data-driven approaches refer to algorithms "that operate on the data before mapping- and rendering in terms of the visualization pipeline and do not affect the visualization." [23] **Screen-based** approaches modify parameters of those two stages.

Clustering the data and visualizing only the centroids in traditional Parallel Coordinates is a **data-driven** approach [23]. Hierarchical Parallel Coordinates [15] are another example of data-driven approach (figure 4.3).

Among **screen-based** approaches, researchers propose frequency encodings like the α channel to influence the opacity of lines, zooming into the image [23] or Parallel Sets [31].

Direct and indirect approaches

Moustafa [36] defined two types of solutions in order to solve overplotting and categorical data representation problems : the **indirect** and the **direct** approach.

The **indirect** approach aims at visualizing the **density** of the mapped information in the Parallel Coordinates. Such a method "[...] assists in identifying dense regions among the axes when dealing with large datasets." An example of this approach is the use of α-blending (a magnitude channel) in order to make polylines semi-opaque, thus revealing the most numerous line segments (see figure 7.1).

The **direct approach** does not focus on the visualization of all the data items but rather at visualizing **summary statistics** of the data. It facilitates the identification of class properties. An example of this approach is the aggregation method, used in Parallel Sets [31], that we define in section 4.2.3.

Screen resolution

A higher screen resolution allows to represent more information. It is the simplest approach to help reduce visual clutter. More pixels are available to represent visual information, making more polylines distinguishable and allowing the visualization of larger datasets.

Filtering

Exploratory data analysis techniques "have limitations on the amount and dimensionality of the data they can process effectively." [11] Filtering consists in removing signals from the input. It can be done at two levels : the first one consists in removing dimensions, i.e. axes, in the Parallel Coordinates (manually or with an algorithm). The second level consists in removing dimension values, with the interaction method known as *brushing*.

Brushing [47] consists in specifying ranges of values and making use of logical operators [33]. It allows to render only a portion of the polylines, reducing the visual clutter. Brushing is relevant for continuous and ordered axes, but reveals to be of poor help with non-ordered categorical axes: if the order of categories does not convey any significant meaning, selecting a range of nominal values does not hold any signification. A good way to implement filtering for non-ordered dimensions is to allow the user to select a discrete amount of categories instead of a range. Therefore, brushing does not solve overplotting problems with mixed data.

The idea of dimension reduction techniques is "to visualize only few [dimensions] that carry out most of the information content in the data" [36]. Cantú-Paz et al. [11] mention several feature selection algorithms in order to remove redundant or irrelevant dimensions that may affect negatively analysis tasks. They mention filters that estimate how well each attribute splits the data between classes using Kullback-Leibler (KL) distance, and algorithms that rank the dimensions in descending order of Chi-square statistics computed from their contingency tables, with the use of decision tree criteria such as Information Gain Ratio (C4.5 algorithm), or with Principal Component Analysis (PCA).

With multivariate datasets, removing irrelevant parameters decreases visual clutter, and thus the user's cognitive load. This "dimensional" approach is good to minimize the amount of dimensions, but does not do anything in terms of reducing the amount of data items to be displayed.

Spatial distortion

Spatial distortion techniques consist in scaling the distance between the ticks on an individual axis, or in modifying the distance between two axes. Two examples of spatial distortion are the fisheye view and the linear zoom [23].

Spatial distortion can help in several ways: it reduces clutter, clarifies dense areas and facilitates the brushing of individual lines with a pointing device [23]. However, like filtering methods, it does not restore the frequency information that is lost: overplotting is still present.

Rosario et al. [43] present a way to give a meaning to the spacing between the levels of a categorical dimension: similar levels are closer to each other. Their approach transforms levels

into numbers with techniques similar to Multiple Correspondence Analysis. By doing so, the spacing among levels conveys semantic relationships. Their method consists in three steps:

1. The **Distance** step consists in identifying a set of independent dimensions that allow to calculate the distance between their nominal levels.
2. The **Quantification** step uses the distance information to assign order and spacing among nominal levels.
3. The **Classing** step uses results from the previous step to determine which levels within a dimension are similar to each other, grouping them together.

An example of the use of this method can be seen in figure 4.12a, where the values are set arbitrarily. The FCA-based, in figure 4.12b, gives a much clearer overview of the datasest. This method can also be used at the dimensions scale, to assign spacing among the axes given a similarity measure. Heinrich et al. [23] advocate precaution regarding the use of this method at the dimensions scale. They state that "horizontal distortion affects angles and slopes of lines, which can have an impact on the accuracy of judging angles", and hence the correlation levels.

Dimensional reordering

Moustafa [36] stated that when comparing the values on one axis with those on a nonadjacent axis, the ordering, rotation and permutation of axes has a significant importance. Placed next to each other from the left to the right, dimensions are not represented independently and the overview differs depending on the order of axes. This imposed order defines which patterns emerge between adjacent axes [23]. For example, two highly correlated dimensions can be placed next to each other in order for the user to detect this insight. It is a good way to reduce clutter, "[...] revealing patterns (e.g. of correlation) that might have been hidden before". [23].

Ordering the axes in a meaningful way is task- and data-dependent: if classes are known a priori, a ranking of axes can be done, "[...] based on principal component or based on their discriminant power" [36]. The axes can be positioned by quantifying the relative influence that each dimension has on the class value. This can be achieved by performing a sensitivity analysis or by using a metric such as entropy: classification algorithms such as C4.5 [40] allow to rank dimensions based on their purity, i.e. their ability to clearly divide the data the between classes. This allows the user to quickly see the most relevant dimensions (visual mining).

For similarity tasks, an ordering based on correlation or on distance measures makes more sense [36]. These ranking methods can also be used to set the distance between axes, positioning similar axes closer to each other as we mentioned in the section 4.2.2.

4.2.3 Overplotting

Overplotting is a particular type of clutter. It occurs on categorical dimensions, when the discrete number of levels is significantly smaller than the size of the dataset, with many samples sharing a given level – it leads to having multiples lines passing through the same points on axes [21]. Figure 4.11 shows an example of overplotting: if several data points have identical values along neighboring axes, a portion of their polylines will overlap exactly. These overlaps hide the frequency information and potential patterns, such as correlations and tendencies.

As we mentioned in section 4.1.1, Parallel Coordinates do not handle well categorical dimensions [23]: "Most existing work has focused on the visualization of numerical data, treating categories as a special case with only a few values." [31] In fact, overplotting results from using a visual channel that does not respect the principle of expressiveness described by Munzner [37]: "the visual encoding should express all of, and only, the information in the dataset attributes". Using a magnitude channel for categorical dimensions violates this principle. With overplotting, the frequency information is not visible.

Standard clutter reduction techniques such as filtering or dimensions reordering are inefficient for categorical dimensions: neither do they remove overplotting, nor do they restore the frequency information that is lost, nor do they remove the visual discrepancy due to the ranking of categories.

Therefore, specific approaches must be used for visual analysis of categorical data. Fernstad et al. [14] group them into two subsets:

- **QuantViz**, visualization techniques that allow to "[...] represent each category with a numerical value (quantification) and then analyze the data using visualization methods commonly employed for numerical data [...]" (see figure 4.13).
- **CatViz**, that is "[...] to employ visualization methods specifically designed for the characteristics of categorical data [...]" (see figure 4.4). This approach respects Munzner's expressiveness principle.

An example of QuantViz can be found in [43], where nominal variables are preprocessed using the *Distance-Quantification-Classing* approach before being imported into the visual exploration tool.

An example of CatViz is the use of a frequency encoding, like Parallel Sets [31] do. Frequency encoding restores the frequency information, but induce another loss of information: only aggregates of information are visualized, and looking up a single data item along the axes becomes impossible.

Aggregation

Aggregation techniques are based on the principle of grouping data, and representing aggregate items instead of individual samples [23]. Such aggregates include mean, median or cluster centroid of a subset of samples.

Frequency-based representations show the distribution of the data items in each level of a categorical dimension. Geng et al. [16] propose angular histograms and frequency curves, two frequency-based approaches to large and high-dimensional data visualization. Another approach is suggested by Teoh and Ma [51]: "For each category, an interval is constructed on the continuous axes to make more polylines visible. The frequency of each category is thus visible to the user." The Information Mural is also proposed [26] but reveals to be hard to read and imprecise for truly categorical data.

Kosara et al. [31] developed the Parallel Sets to represent each category by a visual entity scaled according to its corresponding frequency. They "allow the user to interactively remap the data to new categorizations and, thus, to consider more data dimensions during exploration and analysis than usually possible." This method is best suited for categorical-only data: it does not provide an ideal way to integrate continuous data since each continuous dimension axis is divided into bins, "and thus, transformed into a categorical dimension." With this method, outliers are impossible to detect. The authors note that "showing continuous axes as true Parallel Coordinates dimensions would of course be the most useful display of this data".

Curves

Graham and Kennedy[17] proposed the use of curves in order to help differentiate lines that cross at axes, "an occurrence that increases dramatically when using axes with a few discrete values"[42]. Thought it revealed that "with many curves it became difficult to differentiate them if they were bunched close together along their paths."[42]

4.2.4 Mixed data

Integrating continuous dimensions along with non-ordered categorical dimensions in a unified graphical representation poses the following challenge: user's mental model of these two types of dimensions differs widely. In standard Parallel Coordinates, the continuous design model that is implemented for non-ordered categorical dimensions does not match the discrete user model of the data [31]. Since the visual encoding is the same for two types of dimensions (continuous and nominal), it violates the "expressiveness principle" that we mentioned earlier. Standard Parallel Coordinates induce an ordering of nominal values, while nominal dimensions do not hold any order in their values. They are purely qualitative (e.g.: shirt colors) and representing an order between them can lead to wrong conclusions from the data analyst.

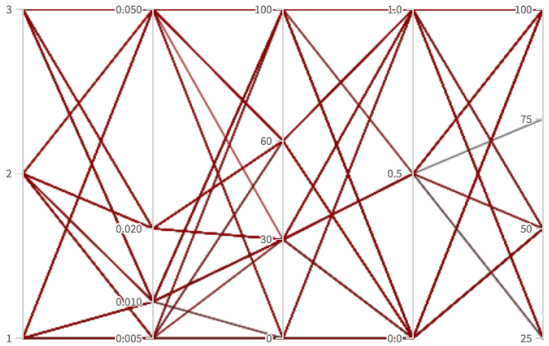


Figure 4.11: Example of overplotting in Parallel Coordinates, with a dataset consisting of 5 categorical dimensions. The distribution of each dimension is hidden, hindering insights with regard to the distribution.

Nominal dimensions require different approaches in order to keep the visualization understandable. Kosara et al. [31] state: "This discrepancy between the user's mental model and the presented image is eliminated by the use of frequency-based techniques: categories are represented by visual entities that are scaled according to their corresponding frequency." Munzner [37] notes that "the identity channels are the correct match for the categorical attributes that have no intrinsic order". Identity channels include the spatial region, color hue, motion and shape.

Values ranking

As opposed to continuous dimensions, discrete dimensions consist of a limited set of values. Each data item is part of only one group (i.e. category) for each dimension, and can not take a value in between. In such datasets we find groups of records that share the same category for a given dimension.

On a nominal axis, values do not have a natural ordering. These values have to be mapped to a metric scale before being visualized. The ranking is usually alphabetic, giving no relevant information to the user and biasing his overview.

Rosario et al. [43] proposed the *Distance-Quantification-Classing* algorithm, that assigns order and spacing among nominal values based on a distance measure. An example of use of this algorithm is shown in figure 4.12. The identification of similar levels inside a dimension becomes easier than with an arbitrary ordering.

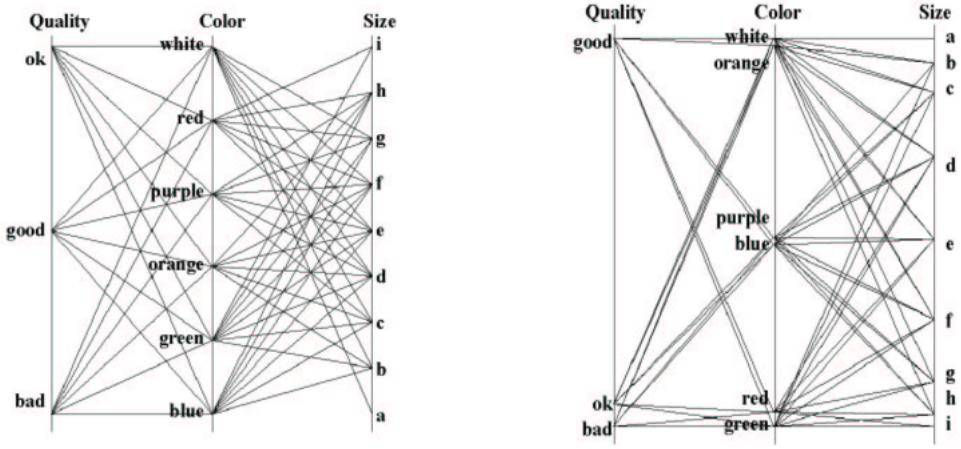
Identity channel

The identity channel includes visual encodings like colour hue, shapes and spatial region. Colour hue is a simple and effective method to distinguish a small set of categories [23, 37]. "Techniques based on colour, blending and curved lines are commonly seen in the literature as suggestions for improving the visual quality." [29] If the color channel is not available, the shape can be used instead, with a slightly reduced efficiency [37].

The spatial region channel can also encode identity. In Parallel Coordinates, removing the line connecting the categorical levels prevents the user to assess it as a continuous scale.

Interaction

Supporting direct user interaction helps the user to acquire useful insights into the information embedded in the underlying data [13, 9]. The usage of selection and highlighting reduce the user's cognitive load and has a crucial influence on the ability to extract patterns. Examples of such interactions are brushing (i.e. interactive filtering) and dimensions reordering.



(a) Parallel Coordinates with arbitrary quantification.

(b) Parallel Coordinates with FCA-based quantification.

Figure 4.12: Focused Correspondence Analysis [43] (at the right) helps the user identify the most similar categories.

4.2.5 Conclusion

Parallel Coordinates do not scale well in terms of dataset size and dimensionality. Furthermore, they implement a continuous design model that does not suit categorical dimensions. We reviewed the visual enhancement methods that aim to reduce *clutter* (due to large datasets), *overplotting* (due to discrete values), and to integrate categorical dimensions in a meaningful way along with continuous dimensions (mixed data).

In the next section, we explain the design choices that led to the two controlled user experiments. We base our choices on the literature review, and on the research guidelines provided by Johansson et al. [29].

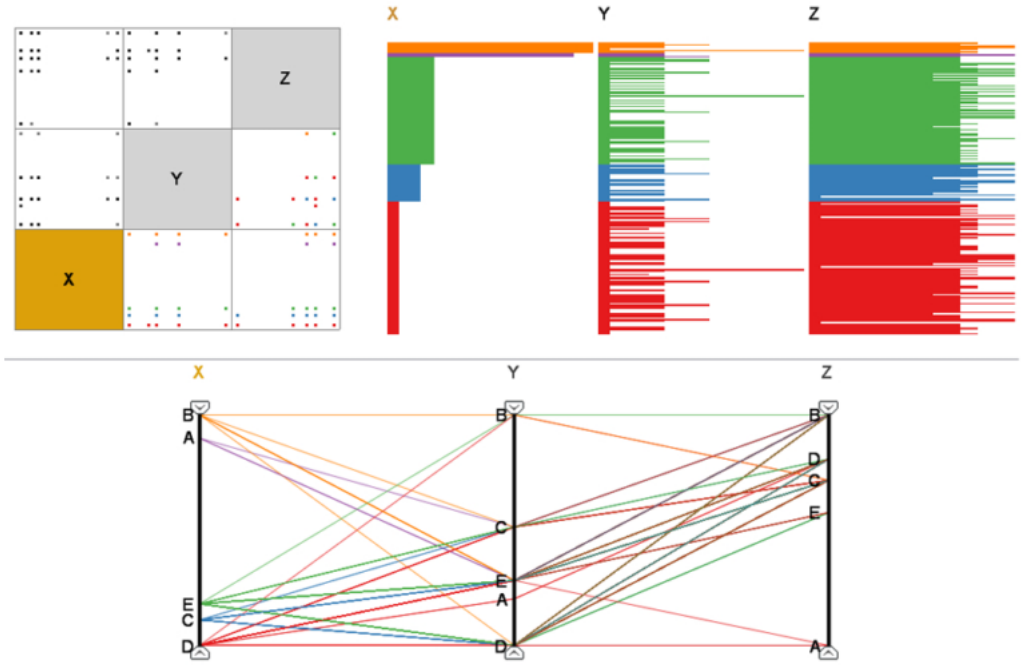


Figure 4.13: QuantViz: a set of visualization methods meant for continuous data is used to represent a categorical dataset. From Kosara et al. [31]

Chapter 5

Design

In this section we summarize the problematic and explain the design choices that lead to the two visualization prototypes. We also explain the technological choices.

In the previous chapters, we reviewed the main visualization techniques and enhancement methods for mixed multivariate datasets. We identified the most common data analysis tasks: frequency- and similarity-based tasks. We also underlined the importance of configuration tasks in the domains that make use of mixed data. Taking into account the number of dimensions and the size of datasets, the most promising methods are Parallel Coordinates and scatterplot matrices.

Regarding similarity and frequency tasks, these two visualizations are relevant: scatterplot matrices can show the relationships between all variables simultaneously, when Parallel Coordinates technique only allows the identification of relationships between adjacent axes. "The visual resolution of scatterplots is superior to Parallel Coordinates and users can distinguish twice as many correlation levels when using scatterplots as compared with Parallel Coordinates." [32] Parallel Coordinates are also more subject to make the user overestimate negative correlations compared to scatterplot matrices [29]. However, in chapter 4.1 we noted that scatterplot matrices do not scale well with many dimensions. Parallel Coordinates handle better high amounts of dimensions, but still require visual improvements in order to reduce clutter, overplotting, axes ordering/spacing and mental discrepancy problems.

We reviewed techniques to reduce clutter in Parallel Coordinates and improve the visualization of categorical dimensions. In order to respect the expressiveness principle, the visual encoding for categorical dimensions has to be chosen carefully, using the identity channel.

5.1 Guidelines for research

In a recent paper, Johansson et al. [29] performed a survey on 23 existing papers that present user-centred evaluations of standard 2D Parallel Coordinates techniques and its variations. They highlight the fact that despite the large number of publications proposing variations of Parallel Coordinates, only a limited number present results from user-centred evaluations. Figure 5.1 gives an overview on the performance of 26 techniques in relation to standard Parallel Coordinates (2DPC), for 7 tasks. We can conclude that up to now, many tasks have not been assessed (configuration, visual mining) and some visualization methods are missing (Trellis Plot, Scatterplot matrix). Moreover, the comparison lacks information about the nature of the data (mixed, continuous, categorical) that was visualized.

They categorize the evaluations as follows :

1. evaluating axis layouts of Parallel Coordinates
2. comparing clutter reduction methods
3. showing practical applicability of Parallel Coordinates
4. comparing Parallel Coordinates with other data analysis techniques

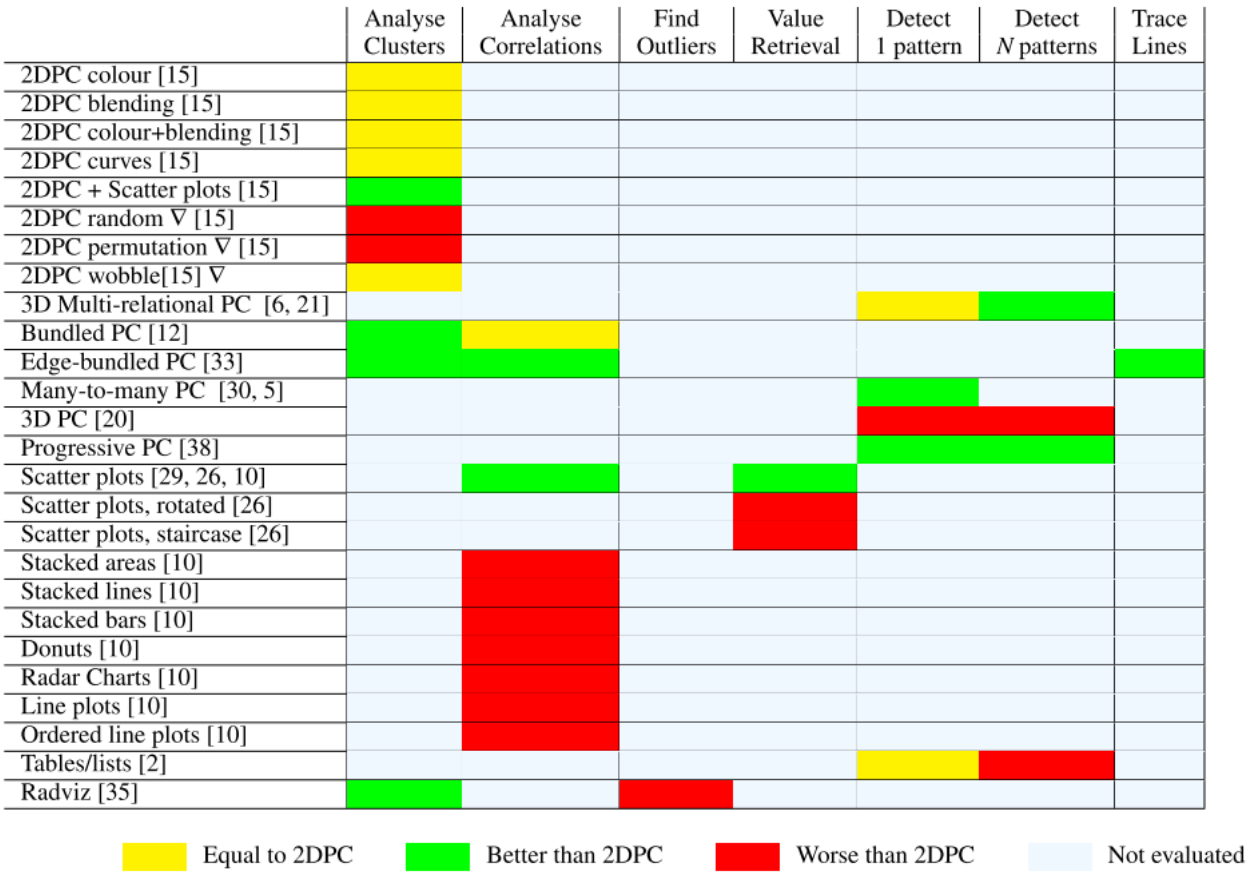


Figure 5.1: Comparison of evaluations of 26 techniques in relation to standard Parallel Coordinates (2DPC). "A yellow colour indicates no significant difference in performance. A green colour means that the technique outperforms 2DPC for the specific task. A red colour means that the technique performs worse than 2DPC. A light blue colour shows that no evaluation has been found in the literature. ∇ denotes that the technique is based on animation" [29]. Source: Johansson et al. [29].

The enhancement of the visualization of mixed multivariate data falls into the categories 1 and 2. Evaluating axis layouts of Parallel Coordinates includes "techniques for arranging axes in 2D Parallel Coordinates in order to highlight specific types of relationships, or for reducing clutter".

They discuss 7 studies that evaluated the axis layouts of Parallel Coordinates. They note that "the 2D Parallel Coordinates axis layout is both effective and efficient for tasks involving comparing relationships between variables". This layout is qualified as intuitive "and novice users learn it without effort". They underline the need to investigate and study systematically the differences between axis layouts with different tasks and users.

To sum up, there still lacks an evidence of measurable benefits that would encourage the use of a variation of Parallel Coordinates over standard ones [29]. It is unclear whether frequency-based techniques positively affect the ability of users to quickly and reliably identify patterns in the data: do they correctly interpret categorical dimensions? Do these methods allow for a better performance in standard data analysis tasks?

Therefore, we focus our research on two aspects influenced by the characteristics of mixed multivariate data. First, we want to test the effect of adding a frequency encoding for categorical dimensions on basic data analysis tasks. Secondly, mixed multivariate data are present in fields in which the configuration tasks are central. Thus, we want to develop a visualization method that allows the user to accomplish configuration tasks faster than with Parallel Coordinates. For this, we make use of the X axis (a magnitude channel) to represent the "performance" dimension, a continuous dimension. The next two chapters describe the protocol of our two experiments.

The first study focuses on "comparing clutter reduction methods" [29]. We compare the user performance in three variants of Parallel Coordinates, with basic data analysis tasks. The results of this first experiment are described in details in section 6.3.

The second study focuses on "comparing Parallel Coordinates with other data analysis techniques" [29]. More precisely, we want to compare the user performance between standard Parallel Coordinates and *Stacked Coordinates*, a new visualization method, in five tasks focusing on several aspects of data analysis: frequency, visual mining, configuration and exploration.

5.2 Frequency encoding: *Parallel Bubbles*

The first problematic that we want to address is the representation of the categorical dimensions in Parallel Coordinates, when they are represented together with continuous dimensions (mixed data). As we noted, Parallel Coordinates present a continuous design model, and are not well adapted to categorical dimensions.

We established that the data analysis tasks are built on two types of tasks: frequency and similarity tasks (i.e. targets). These targets involve one or more dimensions. In order to assess the degree of similarity, or the distribution on a given dimension, the user needs to estimate the number of items that belong to a given level. This capacity is reduced or can disappear completely due to overplotting, that is, the loss of a frequency information occurring due to categorical dimensions. This visual clutter is called overplotting.

In order to compensate this overplotting, we have chosen to add a visual encoding of frequency (a mark) that fulfills two basic objectives:

1. Restore the frequency information that is lost. In order to achieve this, a magnitude channel is used: for a categorical dimension, the size of the mark varies according to the amount of items that belong to each level.

2. Give a visual distinction between the continuous and categorical dimensions, using an identity channel. The shapes of marks representing continuous and categorical are different, mitigating user's mental discrepancy. The user can visually differentiate the type of dimension.

The mark that we have chosen is a circle (a "bubble") that is added to each categorical value, and our prototype is called *Parallel Bubbles*.

We have discovered that no user study compared the performance of frequency encodings in Parallel Coordinates, with similarity- and frequency- based tasks dealing with mixed multivariate data [29]. This is thus a field to investigate, and our first user study addresses it. The goal of the

first study is thus to compare *Parallel Bubbles* to two alternatives: Parallel Coordinates, designed for continuous dimensions, and Parallel Sets, designed for categorical dimensions.

5.3 Configuration tool: *Stacked Coordinates*

Configuration and mining tasks are common in many domains. They imply the design of valid solutions (architecture, industrial design) on multidimensional datasets that mix categorical and continuous dimensions. Thus, a challenge is to improve the user performance in configuration and mining tasks on mixed multivariate data.

The problem of Parallel Coordinates for a configuration task is that they require the user to brush every single value on each axis, in order to compare the corresponding performance on the continuous axis. This requires time, and increases mental load: the user has to remember the output ranges of each category in order to compare them.

Therefore, the goal is to give the user a quick and clear overview on the continuous output of each categorical level, in order for him to find a valid solution faster. We describe below the most pertinent visual enhancements for Parallel Coordinates, as well as the design steps that led to our visualization prototype, *Stacked Coordinates*.

5.3.1 Enhancing Parallel Coordinates

The first idea of improvement is to make use of an identity channel, in order to make a clear distinction between valid and non-valid design alternatives. Each polyline represents an item. The color hue of each polyline is chosen according to a condition based on its performance: red if it is non-valid according to the performance threshold, green if it is valid.

The second idea of improvement is to add jitter (circular noise) to categorical dimensions. This enhances the visual perception of frequency on each categorical axis.

Despite these visual improvements, the user still needs to interact with the visualization in order to compare the performance associated to each category. Moreover, clutter problems still occur due to a high amount of polylines. In previous chapters we noted that only a few user studies evaluate the axis layouts of Parallel Coordinates. Parallel Coordinates are "effective and efficient for tasks involving comparing relationships between variables", and is qualified as intuitive [29]. But as we noted, Parallel Coordinates are not optimal for multidimensional datasets since they only allow comparison between adjacent dimensions, and require many interactions from the user in order to get an overview on the output value. This is why we developed a new visualization method that would overcome these problems by giving a direct overview on the output range of each category. We detail our approach in the next section.

5.3.2 *Stacked Coordinates*

In this section we describe the process that led us to the design of the *Stacked Coordinates* visualization method. The three main steps are illustrated in figure 5.5.

We took as a starting point the Parallel Coordinates layout. The goal was to design a visualization method that would improve the performance of the user in configuration tasks, by giving a clear overview on the output values of each category . The development of this new method was done in the context of the ELSA project. The ELSA tool aims at helping architects in the early design process. We describe this project more in details in chapter 8.

We organized working sessions with members of the EPFL+ECAL team, in order to develop an alternative to Parallel Coordinates. The first thing that we noticed was that in Parallel Coordinates, the user can only assess the performance distribution of the axis that is adjacent to the output dimension (see figure 5.2). In order to get an overview on the distribution of each categorical level without the need to interact, we decided to break down each categorical dimension into one single visual entity. Figure 5.3 shows a way to do this by splitting Parallel Coordinates by categorical dimension. This allows to assess directly the output value for a each dimension.

The line segments between the dimensions still create visual clutter that needs to be reduced if we want to represent large datasets. We noted that representing each design alternative as a dot instead of a polyline requires less pixels, thus reducing the visual clutter.

We also noted that the standard Parallel Coordinates do not make use of the X axis in a meaningful way: the space between axes does not hold any signification, and the X axis is available to encode an additional information. Using the X axis as a magnitude channel seemed to make sense. Therefor, we decided to use it as a way to encode the output value of each categorical value. The figure 5.4 shows the concept of *stacked coordinates*: each dot represents a data item, and the position of a dot on the Y axis depends on the category to which it belongs. Each dot is positioned on the X axis according to its output. A dot having a high output value is positioned to the right of the X axis. A red vertical line is placed on the X value that corresponds to the output threshold. This allows to easily assess the distribution of categories regarding the output value. A category can have its dots grouped on one side of the threshold or the other (e.g. Param2, category A), or distributed equally on both sides (e.g. Param1, category C).

The use of an identity channel to represent the categorical levels respects the expressiveness principle: in figure 5.4 we can see that each level (A, B, C, D) has a defined position, and is not linked visually with the others by a line. Each of these visual entities is displayed simultaneously, and acts like a filter on the data: only the design alternatives that have the specific categorical value are displayed.

With these visual improvements, the configuration tasks should be faster:

- The overview on the distribution of each category regarding the output dimension reduces the user's mental load. When comparing the range of output values of several categories, he does not need to memorize them as he would in Parallel Coordinates.
- The use of an identity channel for each categorical dimension respects the expressiveness principle: the user is unlikely to think that the distance between two levels has a meaning.
- The use of the X axis as a magnitude channel for a continuous dimension respects the expressiveness principle. The effectiveness principle is respected too: the attention of the user is focused on the threshold bar, helping him keep in mind his objective of finding a valid solution.

Removing the need to interact in order to compare the output range of each category should improve the user performance in configuration tasks. To test our assumptions, we compared the performance of jittering- and color-enhanced Parallel Coordinates to the *Stacked Coordinates* in a second controlled experiment (chapter 7).

5.4 Technology

In this section we present the technology that was used in order to implement the visualization tools.

5.4.1 Overview

We implemented the visualization prototypes using the client side web technology stack: HTML, CSS, SVG and Javascript. Since we planned to test the visualizations with several datasets, we needed a flexible approach. Code reusability was also a criterion, because we needed to create several versions of Parallel Coordinates in the first study.

Here is a summary of our requirements:

- **flexibility**: be able to switch between several datasets.
- **code reusability**: be able to make several versions of a visualization method.
- **performance**: be able to handle large datasets without compromising the user experience.
- **interaction**: implement several interactions methods such as clicks and brushing.

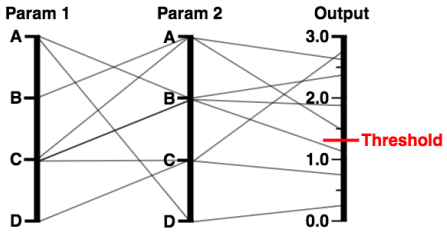


Figure 5.2: Step 1 – Parallel Coordinates with two categorical dimensions (Param1 and Param2), and one continuous dimension (Output).

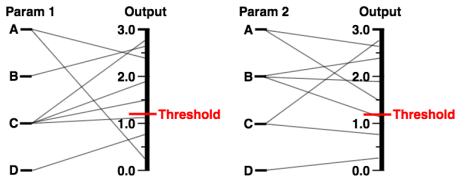


Figure 5.3: Step 2 – Splitting the view into two Parallel Coordinates allows to get an overview on the output value for each categorical dimension.

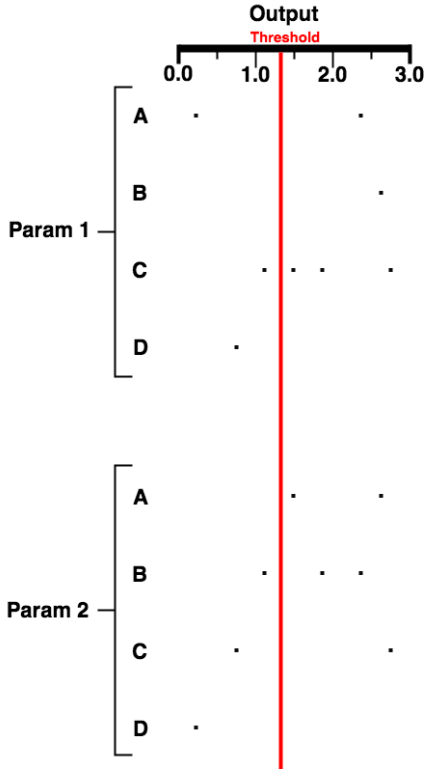


Figure 5.4: Step 3 – *Stacked Coordinates* removes the clutter due to polylines. Assessing the performance distribution of each level is straightforward.

Figure 5.5: The 3-step design process that led to *Stacked Coordinates*.

5.4.2 Implementation

To address these needs, we chose the D3.js JavaScript library[1]. D3.js allows to create and manipulate the Document Object Model (DOM) elements in accordance with the data. D3.js uses a declarative approach, rather than iterating over data structures and creating or updating visual elements in the DOM (imperative approach).

The declarative approach of D3.js consists in these steps:

1. we select a set of DOM elements (that may not exist in the document yet) with the `.selectAll()` operator. Example: to select all the paragraphs, we use `d3.selectAll(p)`.

2. we join data items and existing DOM elements with each other, using the `.data()` operator to specify the list of data items. Each data item can then be used to modify the properties of the DOM element that it is attached to. Example: to add the `var numbers = [15, 8, 42, 4];` array, we write `.data(numbers)`.

3. if no DOM element exists yet (empty selection), we need to create new DOM elements for these items by adding the `.enter()` selection. The `.append(p)` operator specifies which DOM elements needs to be added for each data item.

4. the `.exit()` selection allows to remove the data items that are not in the data anymore, and remove the DOM elements accordingly.

5. the visual aspect of the DOM element can be modified according to the datum that is attached to it. For example, to set the height of a SVG rectangle according to the datum, we can write: `.style(“height”, function(d){return d + "px";})`

A strength of the D3.js library is its seamless integration with current web technologies, giving a maximal compatibility with all current web browsers, making it simple to share the prototypes with collaborators to gather feedback. Secondly it works with the client side web technology stack (HTML, CSS, SVG), and makes the development easier by taking advantage of embedded browser's functionality, like the mouse interactions. The rather steep learning curve of D3.js is compensated by a large documentation made available on the web.

D3.js uses a CSS-style selector to select a set of nodes in the DOM. Next, operators are used in order to manipulate their characteristics (e.g. color, size, position) according to the data. For example, in order to create the axes of a Parallel Coordinates plot, we have to

In addition to D3.js, we used Browserify[2], a Javascript tool that allows to maximize code reusability. We can define dependencies and Browswerify bundles them all up into a single JavaScript file. Segments of code can then be used in several visualization prototypes.

5.4.3 Database

The datasets were stored as CSV files. D3.js allows to get the data from such files with the command `d3.csv(file_url)`.

For the first user study, we used Python and the `numpy` in order to generate the three datasets required. The function `random.multivariate_normal(mean,cov,size)` allowed to generate random samples from a multivariate normal distribution.

For the second user study, we used the cars dataset[3], reduced to four dimensions.

[1] D3.js: https://d3js.org/
[2] Browserify: http://browserify.org/
[3] Cars dataset: https://archive.ics.uci.edu/ml/datasets/Auto+MPG

Chapter 6

User study 1: *Parallel Bubbles*

This first user experiment compares the user performance in similarity- and frequency-based tasks, among three variations of Parallel Coordinates: standard Parallel Coordinates [24], *Parallel Bubbles* and Parallel Sets [31].

6.1 Context

From the literature review, we learned that many publications propose variations of Parallel Coordinates and methods to face the challenges arising from multidimensional mixed data, but only a limited number presents results from user-centered evaluations [29].

Another reason to focus on Parallel Coordinates is that they have "often been found to be advantageous to stateof-the-art techniques when introduced in a new application area." [29] The same authors add that "although users tend to be confused at the beginning they quickly learn how to use Parallel Coordinates and tend to appreciate the way they can interact with their data." The understanding is still limited on "[...] when to use a specific visualization technique for a specific pattern [...]". Therefore, it is important to conduct further studies in this area.

6.1.1 Tasks

For the sake of consistency with other studies [31], we focused our search on similarity- and frequency-based tasks. As we noted in the literature review, these two types of tasks represent the main tasks performed by data analysts [14]. The exploration of a multidimensional dataset consists in a sequence of these two types of tasks, until an hypothesis or a pattern is found. Similarity tasks are about comparing two dimensions at a time, allowing to detect patterns. Frequency tasks are useful to assess the distribution of categorical dimensions, allowing to detect the most numerous category.

6.1.2 Clutter reduction with a frequency encoding

With the two types of tasks described above, we focus on the smallest level of granularity of Parallel Coordinates, that is the comparison of similarity between two axes, and the distribution on one axis. Therefore, we chose to use a dataset consisting of two dimensions. Since the order of two axes does not have a strong meaning, this study does not address the techniques to arrange axes of Parallel Coordinates.

The focus was set on visual clutter, which is "[...] a major limitation of Parallel Coordinates, regardless of whether the technique is implemented as a 2D or 3D display." [29] From the same authors, we learn that "the evaluations performed present no improved general performance of the different variations. In order to advance this area of research, existing approaches need to be further evaluated in order to find out which techniques have potential and which should be avoided. The knowledge gained from these studies should then be fed into the development of new approaches for clutter reduction."

6.2 Visualization method: *Parallel Bubbles*

The approach that we propose, *Parallel Bubbles*, should enhance the visual perception of categorical dimensions by adding a visual encoding of frequency. The cornerstone of *Parallel Bubbles* is a "bubble" (a circle) of variable radius that represents a categorical level. Its area is linearly proportional to the amount of items that belong to the said value: the radius of each bubble is defined by computing the square root of the number of items, and multiplying it by two. Sets of vertically aligned bubbles are arranged on each categorical dimension axis. Continuous dimensions are represented as regular Parallel Coordinates axes, and are easily distinguishable from the categorical axes. Each data item is represented by a polyline passing through every continuous and categorical axes. This approach gives an idea of the distribution of categories inside the data and gives a visual help : it restores the frequency information lost because of overplotting

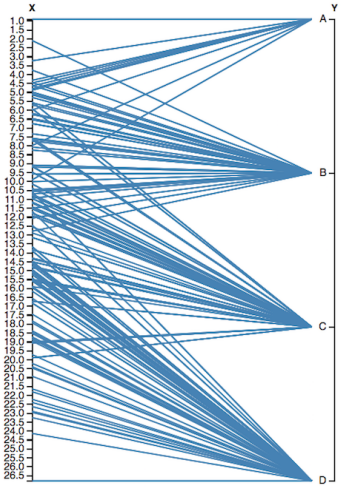


Figure 6.1: Parallel Coordinates

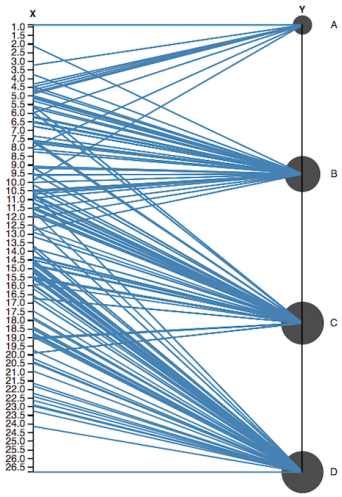


Figure 6.2: Parallel Bubbles

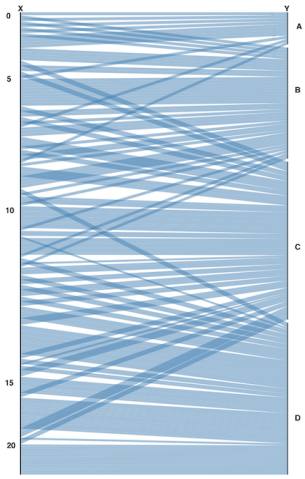


Figure 6.3: Parallel Sets

Figure 6.4: The three variants of Parallel Coordinates that we tested. The left axis is continuous, the right axis is categorical. Here, the dataset 2 (mild correlation) is represented.

6.3 The study

This first user study consisted in adding a frequency encoding to the categorical dimensions of Parallel Coordinates. We call the resulting visualization method *Parallel Bubbles*. As we saw in section 4.2.1, overplotting is a recurring problem with categorical dimensions, and frequency-based approaches seem promising in order to restore the lost frequency information. In this experiment we propose a new method, *Parallel Bubbles*, a frequency-based method meant to enhance the perception of categorical dimensions in Parallel Coordinates. We compared the user performance with three variations of Parallel Coordinates (see figure 6.4): standard Parallel Coordinates plots, *Parallel Bubbles* and Parallel Sets[31]. As measurements, we recorded time and computed a score on similarity and frequency tasks. The results gave us new knowledge about the impact of several encodings of frequency on the user performance.

6.3.1 Hypothesis

The main hypothesis that we tested was that *the three types of visualizations induce a significant difference of performance in frequency and similarity tasks, no matter how strongly correlated the data is.* We also stated the following hypotheses : *Parallel Coordinates offer the best performances for similarity tasks, and Parallel Sets offer the best performances for frequency tasks. Parallel*

Bubbles should be a good compromise in terms of performance for both types of tasks, with a significantly better performance than Parallel Coordinates in frequency tasks, and significantly better performance than Parallel Sets in similarity tasks. We justify our hypotheses below, with a short description of the three visualization methods:

- **ParaCoord** – standard Parallel Coordinates (figure 6.1). This method is sensible to clutter and overplotting problems, making the frequency tasks harder to complete. The similarity tasks should still be easy to complete, because overplotting does not hinder the similarity information.
- **ParaBub** – Parallel Coordinates, with the add of a visual encoding of frequency for categorical values ("Y" axis) in the form of bubbles of variable size (figure 6.2). The frequency of categorical values is easier to estimate, and the similarity between dimensions is still easy to assess, thanks to the polylines.
- **ParaSet** – a variation of Parallel Coordinates, encoding frequencies according to the height of the segment on the axis (figure 6.3). Continuous axis, "X", is transformed into a categorical dimension. "X" axis is harder to read than in other visualizations, only a few values being displayed on the graduation. The similarity tasks should be harder to complete, because the polylines are replaced by colored areas. This visualization method should be the best for frequency tasks.

We established three tasks in order to evaluate the performance of participants on the three visualizations. We describe them in the next section.

6.3.2 Tasks

The tasks that we submitted to the participants only represent a subset of tasks commonly carried out by data analysts and that comprise, among others, clusters and outliers identification, classification, or selection of single data points. According to Fernstad and Johansson [14], "the overall task of data analysis is to identify structures and patterns within data. Most patterns, such as correlation and clusters, can be defined in terms of similarity. Hence, the most relevant general task to focus on are, in our opinion, the identification of relationships in terms of similarity." They also stated that "[...] when it comes to analysis of categorical data the frequency of categories, i.e. the relative number of items belonging to specific categories or combinations of categories, is often of major interest and is, as mentioned previously, the main property of focus in categorical data visualization." With this in mind we defined frequency and similarity tasks. Here is a brief description of the two types of tasks:

1. **Similarity** – identify structures in data. To do so, it is important to be able to estimate the strength of the correlation. Polylines are a good way to achieve these tasks (Parallel Coordinates, *Parallel Bubbles*).
2. **Frequency** – estimate the amount of items belonging to a given categorical value. Being able to identify the most represented category is important too. A visual encoding of frequency helps to achieve these tasks (Parallel Sets, *Parallel Bubbles*).

On this basis, we defined the three following tasks:

- **(T1) Similarity task** : *Are the X and Y axes correlated?* Possible answers: Strongly, mildly, not at all.
- **(T2) Frequency task** : *What proportion of lines have the value B for the Y axis?* Possible answers: a range of values from 0% to 100%, by step of 10%.
- **(T3) Frequency task** : *What is the most represented value on Y axis?* Possible answers : A, B, C, D.

We reduced interaction capabilities as much as possible in order to minimize the amount of dependent variables and to obtain more robust results. Thus, we did not allow the user to manually reorder the axes, to change the position of categorical values on the axis and to delete, add or group dimensions together.

6.3.3 Study material

We submitted the tasks in the form of an online survey[1] and participants were recruited via Prolific[2], a crowdsourcing platform dedicated to research surveys. The survey was filled by 367 participants in total. Each of them was paid 0.70£. Since a tutorial was proposed in the beginning of the survey and guaranteed a good comprehension of the visualization methods, we did not put in place any selection criterion.

In order to get more robust results, we tested the visualization methods on various datasets with three levels of correlation. In order to fully control the correlation we generated the data ourselves: we generated three datasets with a Python script, with a size (465 to 480 data items) large enough to generate overplotting. We chose to use two-dimensional datasets because the tasks we defined are comparison tasks. This type of task focuses the user's attention on the smallest level of granularity offered by Parallel Coordinates. As explained above, multidimensional data exploration is based upon a subset of tasks of this type. For our datasets we thus chose to use one continuous dimension X and one categorical dimension Y. The function `random.multivariate_normal(mean,cov,size)` from the `numpy` library allowed us to generate random samples from a multivariate normal distribution. For each dataset we ran this function three times using overlapping input ranges (e.g. for one dataset: $[0, 20]$, $[10, 30]$ $[20, 40]$) in order to make the distribution more even. The `cov` argument contained the covariance matrix which allowed to control the variation of the variable X in regard to the variable Y, by multiplying the elements $C_{x,y}$ and $C_{y,x}$ by an index taking the values 0.0 (no correlation), 0.8 (mild correlation) and 1.0 (strong correlation) for each dataset.

6.3.4 Experimental design

We conceived the study as a "between-group" with as independent variables the type of visualization (ParaCoord, ParaBub or ParaSet) and the type of data (no correlation, mild correlation, strong correlation). The presentation order of the three datasets was counterbalanced using the Latin square procedure [18], giving 6 data order variants for each visualization, for a total of 18 questionnaires. Each participant was assigned to one visualization type.

To avoid any learning bias, each of the three datasets was shown only once to each participant, in the order defined by the Latin square method. Each participant had to follow a tutorial explaining the visualization method that was assigned to him, and then had to perform 9 tasks: 3 tasks on each of the 3 datasets.

6.3.5 Procedure

In order to level the visual analysis capabilities of participants on each visualization method, we placed a written tutorial in the beginning of the survey. At the end of the tutorial, participants had to fill "control questions". We used the results of these control questions in order to exclude random answers and keep the remaining for further analysis. Participants were informed in the beginning of the survey that we would evaluate the performance of visualization techniques rather than their individual performance.

Once the tutorial was finished, the main part of the survey started for the participant. Using the selected visualization method, he had to do the three tasks that we described above, for each dataset.

6.4 Results

We deleted the results of 5 participants who had completed the first task only (1 for ParaCoord and 4 for ParaSet). We analyzed the results of 122 participants for Parallel Coordinates, 124 for *Parallel Bubbles* and 121 for Parallel Sets.

For the task T1 (similarity), whose answer is dichotomous (true or false), we computed the error rate (see figure 6.15). For task T2 (frequency), the answer was given in %, by step of 10%.

[1] SurveyMonkey: https://surveymonkey.com
[2] Prolific: https://prolific.ac

For consistency with other studies [22, 49], we computed the log absolute error of accuracy in this way: $log_2(|judgedvalue - truevalue| + \frac{1}{8})$. For task T3 (similarity), whose answer was dichotomous too, we computed the error rate (see figure 6.15).

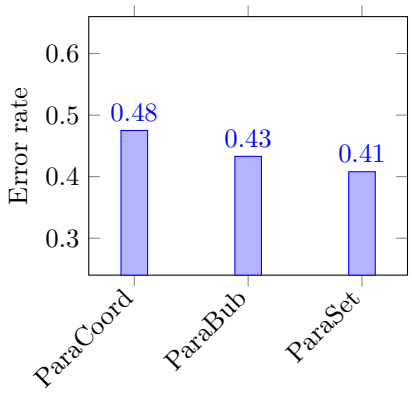


Figure 6.5: Error rate for T1 (similarity) on all datasets.

6.4.1 Overall performance

We performed an analysis of variance (ANOVA) [18] on the results of the three tasks. The results of the analysis rejected the null hypothesis with $p < 0.01$, confirming the hypothesis that the type of visualization induces a significant difference in terms of performance performance.

6.4.2 T1 – Similarity task

The error rates of the three visualization methods are visible in figure 6.5. On the three separate datasets, the ANOVA revealed a significant difference only on dataset 2 (mildly correlated) and with $p < 0.5$ and $F(3.0205) = 5.7048$.

We performed a T-test on the scores of each pair of visualizations: there is a significant difference ($p < 0.5$) between the scores of Parallel Coordinates and Parallel Sets, and between the scores of *Parallel Bubbles* and Parallel Sets. The difference is not significant between Parallel Coordinates and *Parallel Bubbles*. We can thus conclude that Parallel Sets are the best suited in similarity tasks.

6.4.3 T2 – Frequency task

For T2, we computed the mean of quartiles 1 and 3 for the log absolute error, and the 95% confidence interval. Both are visible in table 6.1. A plot representing these data is visible in figure 6.6. A decomposition by type of data (1 = non correlated, 2 = mildly correlated, 3 = very correlated) confirms the global results – without a surprise, very correlated data lead to the smallest error rate (see Figure 6.10). We note that Parallel Coordinates are systematically outperformed by the two other methods.

We conducted a Two-Way ANOVA in order to see if these performance differences for T2 were significant. As we can see in the table 6.2, the mean responses for the three levels of the factor `visu` are significantly different: F(9.44, 1092) = 0.0001, P < .05. This means that the use of a visual encoding of frequency (sets or bubbles) leads to a significant difference in terms of performance. The mean responses for the three levels of the factor `data` are significantly different: F(59.11, 1092) = 0, P < .05. Combining this with the results of figure 6.10, we can tell that the most strongly correlated data leads to lower error rates. Performed on the three separate datasets, the ANOVA revealed a significant difference on dataset 1 (not correlated) with $p < 0.5$ and $F(3.0205) = 3.2678$ and dataset 3 (strongly correlated) with $p < 0.5$ and $F(3.0205) = 4.7971$.

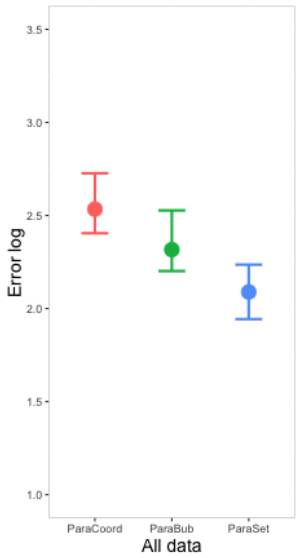


Figure 6.6: T2: Average of quartiles Q1 and Q3, for the log absolute error, for all datasets. 95% confidence intervals.

Type	Log error	CI 95%
ParaCoord	2.534	±0.165
ParaBub	2.316	±0.159
ParaSet	2.089	±0.142

Table 6.1: T2: Average of quartiles Q1 and Q3, for the log absolute error. 95% confidence intervals. Split by visualization type (ANOVA: $F(6.9514) = 8.5469, p < 0.01$)

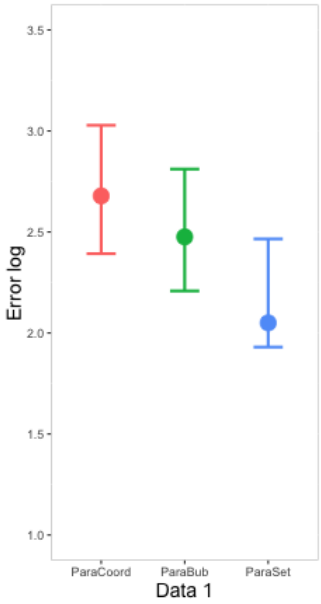


Figure 6.7: Data 1: Non correlated data.

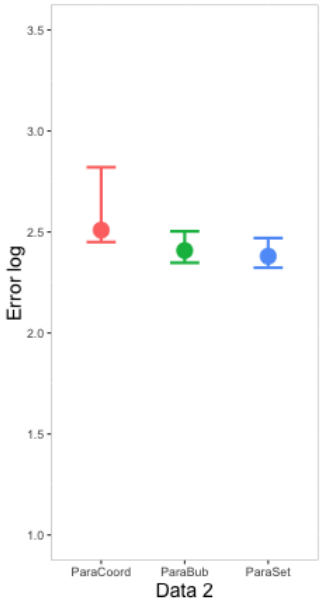


Figure 6.8: Data 2: Mildly correlated data.

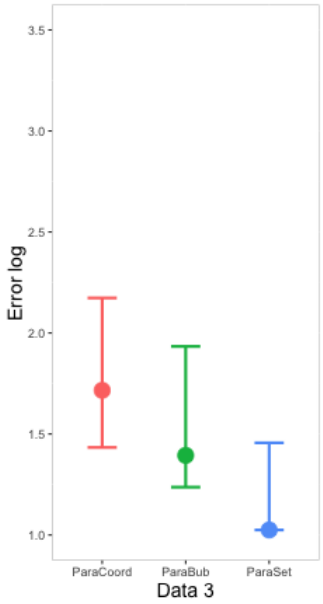


Figure 6.9: Data 3: Very correlated data.

Figure 6.10: Average of the 20th and 80th percentiles, and 95% confidence interval for the log absolute error for T2. Split by data type.

Source	**d.o.f.**	**F**	**Prob>F**
visu	2	9.44	0.0001
data	2	59.11	0
visu*data	4	0.71	0.587
Error	1092		
Total	1100		

Table 6.2: Two-Way ANOVA performed on the scores of T2 (frequency).

The mean responses for the interaction `visu` * `data` are not significantly different; this means that the significant difference between the three types of visualization is valid for various levels of correlation. The ranking of performance among visualization methods stays the same, no matter how strongly correlated the data is.

We performed a post hoc test (multiple comparisons), in order to discover which means are significantly different from each other. For the visualization type, the data type and the interaction between visualization and data. By looking at the MultCompare for the interaction (figure 6.13), we notice that no matter how strongly correlated the data is, the ranking of visualizations stay the same.

Regarding the MultCompare for visualization types (figure 6.11), we can see that the performance of Parallel Sets is the best for this task, and significantly different from the performance of Parallel Coordinates and *Parallel Bubbles*. The performance between Parallel Coordinates and *Parallel Bubbles* is not significantly different. The Parallel Sets deliver the best performance for this task.

Regarding the MultCompare for data types 6.12, we can see that the least correlated datasets (1 and 2) are part of the same population, while the dataset 3, with the weakest correlation, leads to the poorest performance.

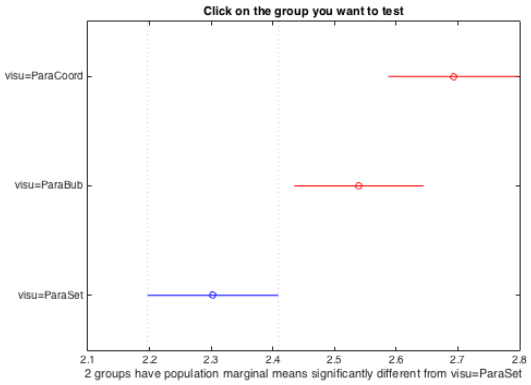


Figure 6.11: Multcompare for visualizations.

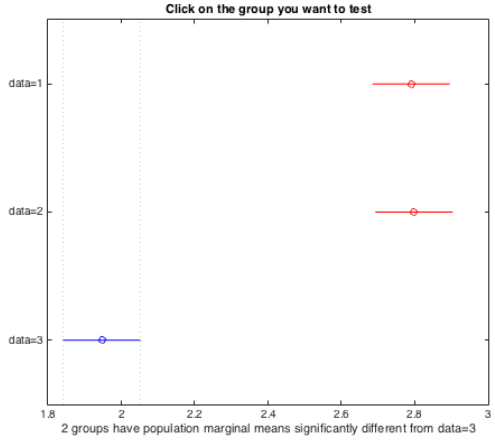


Figure 6.12: MultCompare for data.

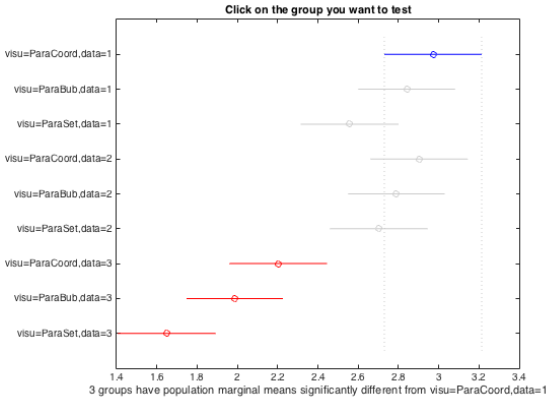


Figure 6.13: MultCompare for the interaction visu*data

Figure 6.14: MultCompare for T2. All visualizations and data.

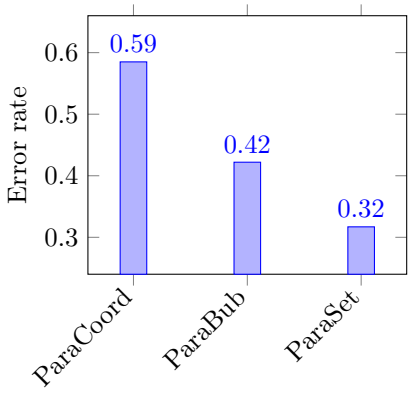


Figure 6.15: Average error rate for T3 (frequency) on all datasets.

These results show that adding a frequency encoding has a significant influence on performance for similarity and frequency tasks, no matter how correlated the data is.

6.4.4 T3 – Frequency task

The error rates of the task T3 are visible in figure 6.15. The ANOVA revealed a significant difference for all three datasets: for dataset 1 (not correlated) with $p < 0.5$ and $F(3.0205) = 5.029$, for dataset 2 (mildly correlated) with $p < 0.001$ and $F(7.0405) = 46.1558$ and for dataset 3 (strongly correlated) with $p < 0.001$ and $F(7.0405) = 47.7846$. The T-Test ($p < 0.05$) performed between each pair of visualizations revealed a significant difference between all visualization methods. The Parallel Sets are always better than the two other methods for this frequency task.

6.5 Conclusion

Our main hypothesis was not completely verified: *"Parallel Bubbles are a good compromise in terms of user performance in frequency and similarity tasks, when compared to standard Parallel Coordinates and Parallel Sets."* The difference of performance was significant between the *Parallel Bubbles* and Parallel Sets, for all tasks. When comparing Parallel Coordinates and *Parallel Bubbles*, the difference was significant only for the task 3 (Frequency). Contrary to what we supposed, Parallel Coordinates delivered systematically worse performances in similarity tasks. This can be due to the fact that the tasks were focused on the categorical axis. The visual encoding of frequency of Parallel Sets seems to be more effective than the encoding of *Parallel Bubbles*.

The better performance of Parallel Sets in comparison with *Parallel Bubbles* might be caused by the difference of accuracy with which we perceive the channels used. For *Parallel Bubbles*, we used an area encoding: the area of bubbles is linearly proportional to the number of items, the radius of a bubble being equal to the square root of the frequency, multiplied by two. For Parallel Sets, we used a length encoding: the height of the boxes was defined as linearly proportional to the frequency.

From Stevens [50], we know that the apparent magnitude of an area is perceptually compressed (by a factor of 0.7), while the perception of length is very close to the true value (see figure 6.16). Based on these results, we propose ways to improve the *Parallel Bubbles* in the next section.

6.5.1 Future improvements

One way to improve the *Parallel Bubbles* would be to define the radius of circles as linearly proportional to the frequency, instead of the square root that we used here. This would counterbalance the psychophysical effects related to area representation defined by Stevens [50]. The performance of *Parallel Bubbles* would probably increase.

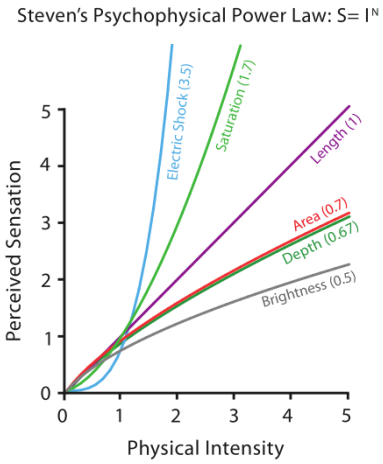


Figure 6.16: The psychophysical power law of Stevens [50]. "The apparent magnitude of all sensory channels follows a power function based on the stimulus intensity." From [37]

We could also consider an alternative of *Parallel Bubbles* composed of "sets" for the categorical axes, and regular axes for continuous axes.

As future works, it would be interesting to further extend the *Parallel Bubbles* method, implement it in a functional system and use it in a real context. It would be pertinent to test it in a use case.

Future studies should focus on tasks centered on the continuous axis; performances of Parallel Coordinates would probably be better due to the continuous design model they implement. Conversely, Parallel Sets would give the worst performance.

Chapter 7

User study 2: *Stacked Coordinates*

In this chapter we describe the second user study that we conducted. It focuses on the configuration tasks.

The aim of this study is to evaluate the effectiveness and efficiency of the new visual approach that we developed, *Stacked Coordinates*, in common data analysis and configuration tasks. More specifically, we want to compare the user performance when carrying out design tasks using Parallel Coordinates and *Stacked Coordinates*.

7.1 Context

When compared with other visualization techniques, Parallel Coordinates "[...] give a good overview of the data." [29]. As we concluded in our first study, this technique is well suited for mixed multivariate datasets when a visual encoding is added to it. However, several problems remain: the order of axes heavily influences the patterns that can be detected and the mental load gets high due to the large amount of information displayed. Moreover, the one-to-one mapping between the data and the visualization involves a large amount of polylines: this results in overplotting and clutter problems [36].

Parallel Coordinates are a common tool for user-involved tradeoff analysis. In such tasks, the user has to choose a set of values defining a solution, given one or more constraints. The goal of the user is to explore the design alternatives, to choose the dimensions values that fit some criteria, and to find one (or many) valid solution(s) defined using the dimensions.

As we explain in chapter 5, we designed *Stacked Coordinates* in order to give an overview on the output dimension for each separate categorical dimension.

In their literature survey of user-centred evaluation of Parallel Coordinates, Johansson et al. [29] state that "the knowledge of when to use a specific technique is still low and substantial research is needed to fill the gaps [...]". "It will also be necessary to go beyond linear relationships and study more complex, non-linear relationships in order to fully utilize Parallel Coordinates."

7.1.1 Underconstrained configuration tasks

Underconstrained configuration tasks are central in many fields, such as industrial and building design. No optimal solution exists, and the final choice depends on both qualitative (preference, comfort) and quantitative criteria (performance threshold). Such tasks involve both human and machine: the user is actively involved in the decision making process and needs to choose among a set of acceptable solutions. A constraint is a condition that has to be met in order for the solution to be valid. It is usually a numerical value representing cost, or performance. We call this numerical value *output* in the reminder of this chapter.

Underconstrained configuration tasks are often performed on mixed multivariate datasets. Such datasets can be the result of simulations that estimate the output value for any given combination of dimensions. As an example, building performance simulations (design alternatives) include a

dozen of dimensions and several output values (i.e. environmental indicators) representing the amount of CO2 emitted by each design alternative.

As we described in earlier chapters, several visual encodings can enhance the visualization of mixed multivariate datasets in Parallel Coordinates. However, unlike frequency and similarity tasks, configuration tasks require the user to focus mainly on the output value.

Given a mixed dataset containing several categorical dimensions and one continuous dimension, our idea is to test an alternative of Parallel Coordinates by representing the distribution of each categorical value on an individual axis. Each reference in the dataset is displayed as a dot on the corresponding axis. The position of each dot on the continuous axis is given by the value it belongs to. Making use of the X axis in a meaningful way, the user gets an overview on the output values for each category. This should improve the performance of the user in configuration tasks.

7.1.2 Qualitative feedback

The other aim of this study is to get an appreciation of the users regarding the two types of visualizations. Johansson et al. [29] underline the importance of evaluating the aesthetic aspect of Parallel Coordinates, and the lack of attention in this area: "the design and aesthetics of Parallel Coordinates have not received much attention in previous work." They recommend to take the direction of "[...] more qualitative studies and studies executed in the field rather than in controlled lab settings." Traditional metrics such as error rates and response times are important but narrow.

"It can be expected that a Parallel Coordinates display that is visually appealing with intuitive interactions would attract the attention of more users and stimulate uptake and usage. To reach such results would require qualitative evaluation that investigates users' subjective actions, opinions and attitudes in depth." [29]

For these reasons, we asked the users to compare the aspect and ease of use of the two visualizations, and to give their appreciation.

7.2 Visualization method: *Stacked Coordinates*

Taking as inspiration the Parallel Coordinates, we developed a new way to visualize the data. The aim of this visualization method is to guide the user towards an specific "output" goal. Parallel Coordinates with jittering, alpha blending and color encoding allow to assess the distribution of all dimensions, regarding one continuous axis; however, interaction is required in order to get the output range of a given categorical value. It is not possible to compare the output range for each categorical value. The idea of this new visualization is to visualize in an easy way the distribution of the data points on a continuous X axis, for each categorical dimension. Knowing this would help in configuration tasks, since Parallel Coordinates do not seem ideal in order to help the user achieve configuration tasks quickly.

7.2.1 The visual metaphor

Stacked Coordinates are particularly suited for categorical multivariate data visualization. They allow to get a quick overview of the range of outputs that a given parameter value leads to. This should give the data analyst a simple way to identify the best and worst parameters for his design. Like Parallel Coordinates, this visualization reads in the vertical and horizontal directions. Reading in the vertical direction gives an overview of the available values for each parameters, while reading in the horizontal direction gives an idea about the output range for each parameter value. A frequency information is also given in the form of semi-opaque dots representing the data points. We describe these three visual encodings in the following.

The basic building block of *Stacked Coordinates* consists in a group of dots representing the distribution of a categorical dimension value on the continuous axis. Each dot represents a single data item. Its position on the X axis is determined by the output value of the reference. The groups of dots are stacked on top of each other.

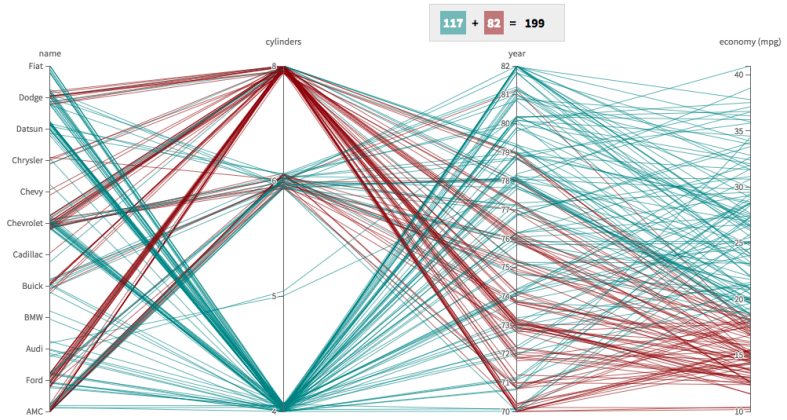


Figure 7.1: Parallel Coordinates displaying the subset 1 of the cars dataset.

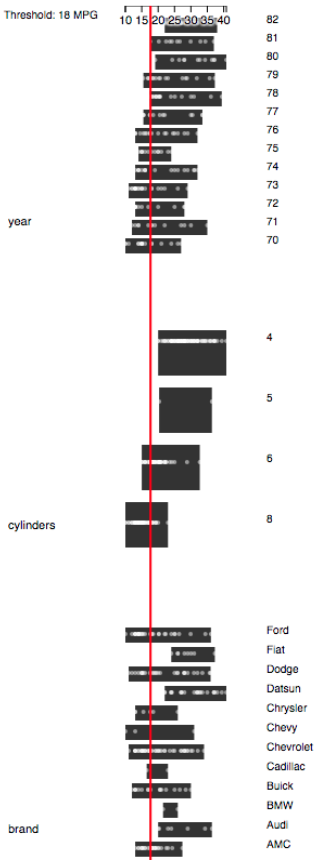


Figure 7.2: *Stacked Coordinates* displaying the subset 1 of the cars dataset.

Figure 7.3: The two visualization methods that we compared. The first subset of the cars dataset is displayed.

In order to represent the output range of a given dimension value in a more easy way, a rectangle is positioned behind each group of dots. The left-hand side of each rectangle is positioned on the X axis according to the minimum value of the set of dots, and the position of the right-hand side is determined by its maximum value. Doing this for every parameter value, all rectangles have different starting points and widths.

The whole visualization looks like a tower composed of groups of rectangles, themselves containing white dots that represent the distribution of references on the continuous axis. The representation of dots on the X axis acts at two levels:

1. each dimension is represented by a set of rectangles. The user can easily assess the extreme output values for each category.

2. in a given category (a rectangle), the position of the dots on the X axis gives the user a way to assess the distribution of the data items. He can assess whether the given dimension is mostly associated to valid or invalid solutions. He can quickly identify which category is the most represented in the database.

7.2.2 Interaction method

By default, the whole database is shown. All categorical values are available to the user. The user can do two main actions: select a parameter value, or deselect it. Similarly to brushing in parallel coordinate, selecting a dimension value (i.e. clicking on a rectangle) acts like a filter on the dataset and only the subset of the data is shown in the visualization.

1. for the selected dimension, all other values (i.e. sets of dots) are hidden.

2. the other dimensions are also filtered. A verification is made for each dimension: if a value is not available in the database in combination with the actual selection, its corresponding set of dots is hidden.

3. the range of the output is updated, and the width and position of rectangles is updated accordingly. The width narrows down more and more at each selection.

Once the user has selected a value for each dimension, the solution is considered valid if its position on the X axis is located on the good side of the threshold. The good side of the threshold is defined by the goal (stay above or stay below).

7.3 The study

In order to compare the Parallel Coordinates with the *Stacked Coordinates* we have set up a controlled user experiment. In the following we define the hypotheses that we wanted to verify, the tasks that we submitted to participants, and the experimental design.

7.3.1 Hypotheses

Our hypothesis is that the *Stacked Coordinates* allow the user to perform configuration tasks faster than with Parallel Coordinates. The visual representation of the distribution of each category should guide the user towards valid choices more efficiently than Parallel Coordinates do. The user should be able to identify the best and worst values at a glance.

The representation of the distribution for the value of each dimension should facilitate the identification of the best and worst values with the use of the X axis threshold. A secondary hypothesis is that the *Stacked Coordinates* can outperform Parallel Coordinates in frequency-based tasks.

The polylines that are used in Parallel Coordinates help to assess the similarity and correlation level between dimensions. Data exploration involves the visual detection of such patterns. Since the *Stacked Coordinates* do not display links between dimensions, we think that *Stacked Coordinates* are less suitable for data exploration tasks.

7.3.2 Tasks

In order to test these assumptions, we defined a set of 5 tasks, focusing on both data analysis and design approaches. For each task, we recorded the performance in terms of accuracy and response time. At the end of each session, a qualitative feedback was gathered. Participants had to give their opinion regarding the difficulty of achieving tasks with both visualizations, and their personal preference.

Here is a description of the types of tasks :

- **Frequency** – estimate the amount of items belonging to a given category, and identify the most represented category. Estimate the best and worst category regarding the output threshold.
- **Configuration** – given a constraint on one dimension, give a set of combinations of dimensions values that describe valid solutions.
- **Visual mining** – find the most relevant parameter, regarding the output values, based on the concept of entropy. It corresponds to the very first step of the C4.5 algorithm, used to build decision trees. This algorithm ranks the dimensions according to their purity, i.e. information gain, in order to position the purest parameters closer to the root of the tree.
- **Exploration** – extract knowledge from the data. The user has to detect as many patterns as possible, in a limited amount of time. Such patterns include clusters, tendencies, outliers and comparisons.

On this basis, we asked the following five tasks:

- **(T1) Frequency task**: *How many cars have 6 cylinders and have been built between 1975 and 1980?* Possible answers: a number.
- **(T2) Frequency task**: *Which cylinders values are associated with the most cars below the threshold?* Possible answers: the number of cylinders (3, 4, 5, 6 or 8).
- **(T3) Configuration task**: *Give 3 different parameters combinations defining some cars that are above the minimum MPG* Answer : three sets of three parameters that define each one car.
- **(T4) Visual mining task**: *Between "brand", "cylinders" and "year", which parameter seems to be the most decisive to identify if a car is above or under the threshold ?* Possible answers: the name of the most relevant dimension (year, cylinders or brand).
- **(T5) Exploration task** *Explore the data for 5 minutes. What findings do you make? Do you have any insights to give about the data? Do you notice patterns?* Answer: the insights given by the user.

We ended the questionnaire with a qualitative feedback from the participant. We asked him which visualization method he preferred, and the reasons for his choice.

7.3.3 Study materials

All the tests were carried out on a 2011 Apple MacBook Pro with a 2.4 GHz Intel Core i5 CPU and a screen resolution of 1280 by 800 pixels.

The participants were presented with the two visualization methods together with the five tasks to perform.

We implemented both visualization methods were with the D3.js javascript library [1], and made them available in the form of a web tool running on a local server.

Since we needed a mixed multivariate dataset, we chose to use a dimension-reduced version of the cars dataset. We reduced it to 3 categorical dimensions and one continuous dimension. The three categorical dimensions are: brand, cylinders, and year. The continuous dimension is MPG

[1] D3.js: https://d3js.org/

Vis method	Frequency 1	Frequency 2	Configuration	Mining
Parallel Coordinates	21.57	9.25	74.5	3.14
Stacked Coordinates	55.43	10.25	43.38	16.00

Table 7.1: Average completion times for all tasks, after removing the outliers.

(Miles per Gallon). We also defined a threshold for the output: cars that are above 18 MPG are considered valid (i.e. eco-friendly), and cars that are below this threshold are considered non-valid. We divided this dataset into two parts of 199 items, in order to be able to verify the influence of the data on the performance. During each session, we recorded the answers and the time for each task.

7.3.4 Experimental design

The experiment was designed as a 3-factor "within-group" design. The two factors were the type of visualization (ParaCoord vs. StackCoord), the dataset (D1 vs D2), and the order in which the visualization was shown (O1 vs O2). Hence, we got $2 * 2 * 2 = 8$ separate experimental phases.

The order of the five task was always the same, because we did not expect any significant effect due to the task order. Thus, we did not consider them as factors within the experimental design.We counterbalanced the presentation of the eight phases with the Latin-square procedure, resulting in eight different phase orders. Each participant got assigned to two of the eight phases orders, for a total of two participants for each phase order. Each participant performed 10 tasks in total (5 per phase).

The study was conducted with 8 participants (7 students, and 1 university professor) of various expertise levels in visual analytics. The supervisor of the study stayed next to the participant during each experiment, in order to answer the questions he might have. Since the tasks needed a certain knowledge about visual analytics, we made sure to level the abilities of participants. Therefore, a short explanation was given to each participant before each task, and the supervisor verified that he had understood the task before starting to record the time.

7.3.5 Procedure

First, we had to level the degree of knowledge of each participant and their visual analytics abilities. Therefore, we gave each participant a verbal introduction and tutorial about the visualization and interaction methods. The concepts of frequency, configuration, entropy and exploration in the context of the study were also described. We explained the four dimensions defining the cars dataset. We told the participants that their response times would be measured and anonymised.

The test consisted in three periods: training, experiment and discussion. The training period allowed the participant to familiarize himself with the visualization and the interaction methods. The experiment period consisted in a sequence of 5 tasks. Once the supervisor was certain that the participant had understood the task, he started the timer and the participant had to answer as quickly as possible. The answers were given orally by the participant, and the supervisor wrote them. Once the two first periods were finished, the participant was asked questions as to which visualization type he preferred using. The questions were: which visualization method did you prefer, and why? Which functionalities were missing?

7.4 Results

In this section we present the results of the study in details. For the tasks T1-T4, we gathered the absolute error and completion time. For the task T5, we compared the amount of insights and the completion time.

The number of participants is pretty low – therefore, in order to get statistically reliable results, we made sure that the measured data met the assumptions required by the T-test. We followed this procedure for the measured data of each task:

1. **Symmetry** – We verified the level of non-symmetry of the data. For this we computed the skewness score [46] and its standard error. If the result of the division of the skewness score by its standard error does not fall within the interval $[-1.96, 1.96]$, the data is not considered as normally distributed with respect to that statistic.

2. **Tailedness** – We verified the level of tailedness by computing the kurtosis score [53] and its standard error. If the result of the division of the kurtosis score by its standard error does not fall within the interval $[-1.96, 1.96]$, the data is not considered as normally distribute with respect to that statistic.

3. **Remove outliers** – If, in one of the two previous steps, the data appeared not to be normally distributed, we proceeded to a visual inspection of the distribution using a histogram. We removed any outlier, then repeated steps 1 and 2.

4. **F-test** – F-test is used to test the null hypothesis that the variances of two populations are equal. Its result allowed to determine if the T-Test was done on groups of equal variance or not.

5. **T-Test** – We performed an unpaired Student's T-test on the data to test the null hypothesis that the population means are equal. We used an α value with a significance of $p < 0.5$.

6. **ANOVA** – We also carried out an n-way analysis of variance with the `anovan` function of MATLAB [34]. This method measures the effects of N factors on a response variable y. Our response variables were the completion time of each task and the absolute error. The factors were the visualization method (ParaCoord vs. StackCoord), the visualization order (1 or 2), and the dataset used (D1 or D2). This method returned the exact p-values for all factors, allowing to measure their respective effect on the mean of the vector y.

The average completion times for each task are visible in the table 7.1. We removed 1 outlier for the task T1 and 1 outlier for the task T4.

7.4.1 T1 – Frequency task

The participant had to answer this question: *How many cars have 6 cylinders and have been built between 1975 and 1980?* The answer given was the number of cars complying with these constraints.

The completion times for the Parallel Coordinates were not normally distributed. Due to this we removed one outlier prior to statistical testing. The population size for each visualization method was thus 7.

The average completion time was nearly three times faster with Parallel Coordinates: 21.57 seconds with the Parallel Coordinates, and 55.43 seconds with the *Stacked Coordinates*. The T-test for the completion time of this task revealed a strongly significant effect of the visualization method, for $p < 0.0001$. The mean and 99% confidence interval for this task are visible in Figure 7.4.

The Two-way analysis of variance (ANOVA) confirmed the T-test: there is a significant effect of the two levels visualization type on the completion time. Mean responses of the vector `completion time` are significantly different for the levels 'ParaCoord' and 'StackCoord' of the factor `visualization`. The p-value 0.9392 indicates that the mean responses for the two levels of the factor `data` are not significantly different.

Surprisingly, the average time to complete the task was longer for the second try: 34.29 sec for the first time it was executed, and 42.71 sec the second time. Similarly to the results for factor `data`, the p-value 0.4187 indicates that the mean responses for the two levels of the factor `task order` are not significantly different.

We compared the error rate between the two visualizations. The error rate was 1/7 with the Parallel Coordinates, and 2/7 with the *Stacked Coordinates*.

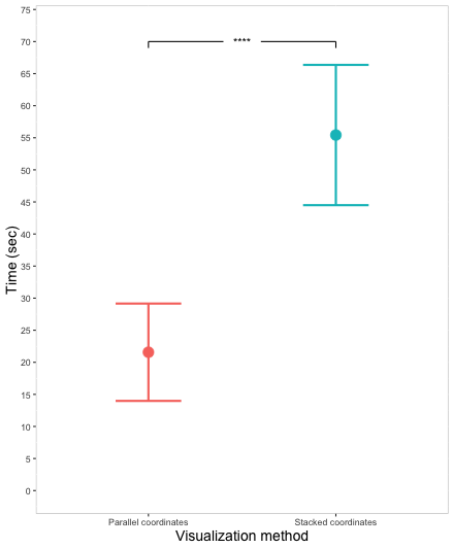


Figure 7.4: Plot displaying the average completion time and 99.99% confidence intervals for task T1.

7.4.2 T2 – Frequency task

The participant had to answer this question: *Which cylinders values are associated with the most cars below the threshold?* The answer given was the number of cylinders.

For this task, the measured time was normally distributed. All participants answered correctly to the question.

The average completion time was 9.25 seconds for Parallel Coordinates, and 10.25 seconds for *Stacked Coordinates* (see figure 7.5). Despite the fact that this task was also frequency-based, the T-test did not show any significant effect of the visualization method on the completion time. The p-value 0.7871 indicates that the mean responses for the two levels of the factor `visualization` are not significantly different.

The 2-way ANOVA showed that the mean responses for the levels `1` and `2` of the factor `data` are significantly different, with a p-value of 0.0356. The average completion time was 5.5 seconds for data 1, and 14 seconds for data 2. This difference can be explained by the fact that this task required to focus on only one dimension, i.e. the cylinders, and that the distribution of classes in this specific dimension is all the more important. The user had to assess the frequency on one axis: evaluate the amount of red lines in parallel coordinate, and evaluate the amount of dots at the left of the threshold bar in the *Stacked Coordinates*. Both datasets have the same population size (199), but the dataset 1 has nearly twice the amount of cars below the threshold (82 for dataset 1 vs 42 for dataset 2). This difference can explain why the factor `data` was more important than the factor `visualization`. Participants executed this task nearly three times faster with the data 1 than with the data 2.

Like for T1, the average time to complete this task is longer the second time it is executed: 8.5 sec for the first time, and 11 sec the second time. The factor `task order` does not induce any significant difference on the time performance, with a p-value of 0.554.

7.4.3 T3 – Configuration task

The participant had to answer this question: *Give 3 different parameters combinations defining some cars that are above the minimum MPG.* He had to answer by giving three sets of three parameters that define the cars.

The time performance was better with the *Stacked Coordinates*, with an average of 47.38 sec for *Stacked Coordinates*, and an average of 74.50 sec for Parallel Coordinates (see table 7.1 and figure 7.6). However, the T-test and ANOVA did not show any significant effect of the factor

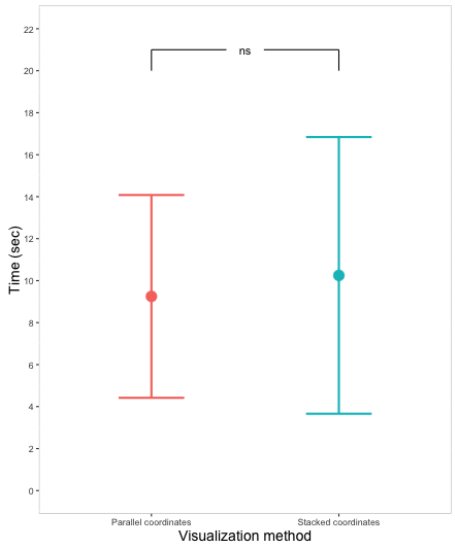


Figure 7.5: Plot displaying the average completion time and 95% confidence intervals for task T2 (frequency). The factor `data` had a significant effect on the mean responses of the completion time.

`visualization` or `data` on the completion time. Our hypothesis is not validated: despite a better performance with the *Stacked Coordinates*, its use does not provide a significantly better time completion for the configuration task. Repeating this task on a larger amount of participants would give more data and thus give less variation: precision increases as the sample size increases [5].

The average time to complete the task was 73.63 sec for the first time it was asked, and clearly shorter the second time, with 48.25 sec. However, the p-value 0.1179 did not show any significant difference of the time completion with the factor `task order`.

We also compared the average MPG performance of given by participants: 31.72 MPG with Parallel Coordinates, and 30.49 with *Stacked Coordinates*. The difference is not significant.

7.4.4 T4 – Visual mining task

The participant had to answer this question: *Between "brand", "cylinders" and "year", which parameter seems like the most decisive to identify if a car is above or under the threshold ?* For this task, the skewness and kurtosis scores showed that the measured completion times were not normally distributed. To get a normal distribution we had to delete one outlier (participant #8).

For Parallel Coordinates the average completion time was 3.14 seconds, and 16 seconds for *Stacked Coordinates* (see figure 7.7).

The T-test showed a significant effect of visualization type on the completion time with a p-value of 0.05. The ANOVA gave the same results, with significantly different mean responses for the factor `visualization` (p-value: 0.0179). The factor `data` does not have any significant effect on the completion time.

The average time to complete the task was 9.86 seconds the first time it was asked, and slightly faster the second time, with 9.29 sec. The ANOVA confirmed that the order in which the tasks are asked does not induce any significant difference on the time performance.

7.4.5 T5 – Exploration task

The participant had to answer this question: *Explore the data for 5 minutes. What findings do you make? Do you have any insights to give about the data? Do you notice patterns?* They answered by giving the patterns they found using the visualization method.

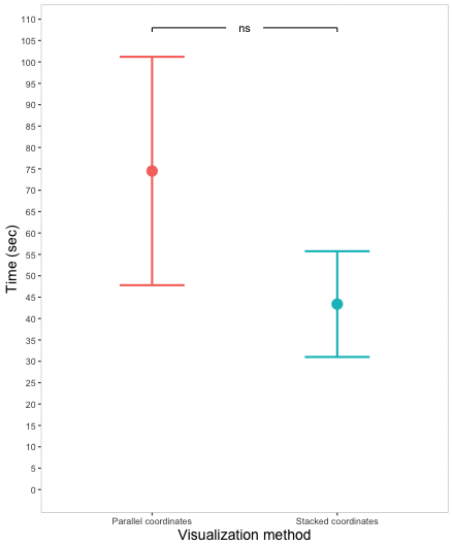


Figure 7.6: Plot displaying the average completion time and 95% confidence intervals for task T3 (configuration).

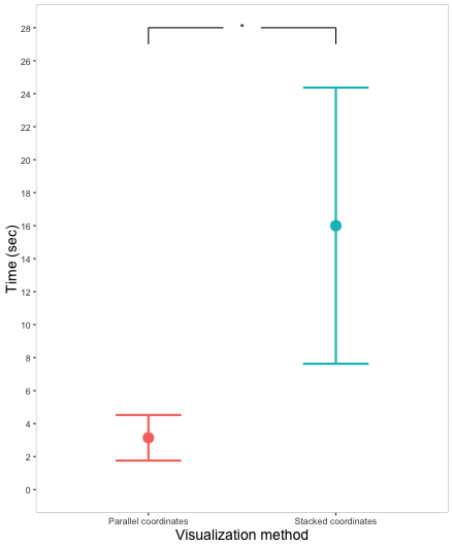


Figure 7.7: Plot displaying the average completion time and 95% confidence intervals for task T4 (visual mining).

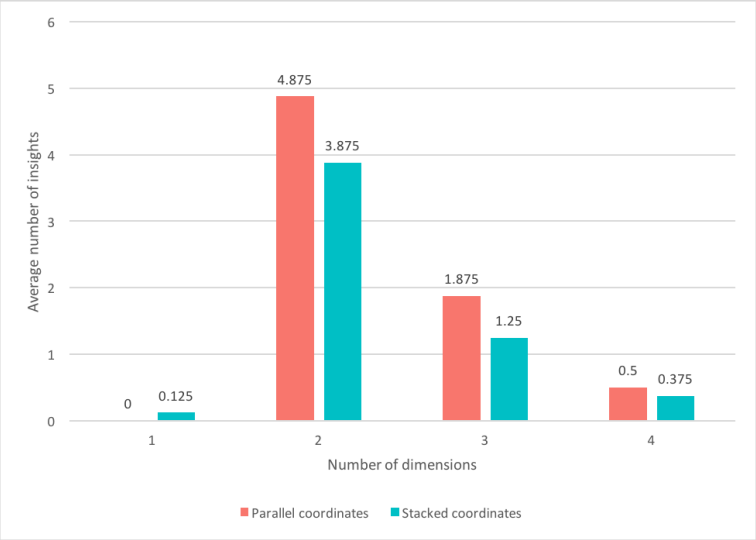


Figure 7.8: Plot displaying the average insights given by participants on each visualization method.

Participants had the goal to discover, i.e. find new knowledge [37] using both visualization methods.

We tried to classify the insights given by participants, using the taxonomy of targets defined by [37] that we explained earlier in section 3.2. We did not succeed: we realized that the classification of an insight depends on its interpretation, and only the data analyst can classify it accurately. Asking participants to classify their insights would have been pertinent.

In average, participants using the Parallel Coordinates gave more insights than participants using *Stacked Coordinates*: 7.38 insights with Parallel Coordinates, and 5.63 insights with the *Stacked Coordinates*. We classified insights using as criterion the number of dimensions they encompass. The average by participant and by number of dimensions is displayed in figure 7.8. Most insights encompass two dimensions, confirming that users principally focus on two dimensions at a time in order to find knowledge. The only insight regarding one dimension was given with *Stacked Coordinates*

The ANOVA did not show any significantly different mean responses for the factors `visualization`, `data` and `task order`. We can thus conclude that both visualizations are well suited for data exploration.

7.5 Qualitative feedback

During the last part of the experiment, all the 8 participants expressed an clear preference for Parallel Coordinates. Tables A.1, A.2,A.3 and A.4 provide the participants' opinions regarding the visualization methods and data analysis tasks.

7.5.1 Usefulness and usability

Regarding the overview, one recurring comment about *Stacked Coordinates* was that they allow a good overview on the distribution and on the range of values, without requiring any interaction. A participant mentioned that the quality of the overview decreased when values were selected. This seems to be valid: values that are not included in the selection are completely hidden. Two participants told that in *Stacked Coordinates*, the threshold is easy to assess thank to the threshold line. With Parallel Coordinates, the following comment was mentioned: unlike *Stacked Coordinates*, the overview stays good, even while selecting parameters. The polylines that are outside of the selection are grayed out and still visible in the background. In Parallel Coordinates, the

use of colors was mentioned by three participants as a strength to assess the distribution of values regarding the threshold.

Regarding configuration tasks, one participant noticed the usefulness of the Stacked Coordinates to confirm an existing hypothesis, with a goal in mind. A total of 6 comments were made about the usefulness of *Stacked Coordinates* to assess the output values of one category. This can explain the better time performance with this visualization method in the configuration task (T3). In Parallel Coordinates, some participants reported the color coding of the output class made it easier for them to identify the distribution of categories.

Two participants reported that Parallel Coordinates seem better to identify an individual line.

Three participants mentioned that Parallel Coordinates are also better to detect tendencies (correlations, similarities).

For Parallel Coordinates, a participant found that the jittering (circular noise), changing randomly at every brushing, could be misleading. One participant also forgot that some dimensions were already brushed. Resetting the selection after a while, or making the selection more obvious could help correct this problem.

7.6 Conclusion

Our main hypothesis was that the *Stacked Coordinates* should provide a significantly better time performance than Parallel Coordinates in configuration (T3) tasks. The statistical analysis tends to indicate that *Stacked Coordinates* allow the user to finish the given configuration task faster than with Parallel Coordinates, but that the difference is not significant. Thus, the study should be repeated on a larger number of participants in order to get more precise results.

Our second hypothesis was that the *Stacked Coordinates* allow a better time performance in frequency (T1 and T2) and visual mining (T4) tasks.

For the first frequency task (T1), that requires the assessment of two dimensions, the statistical analysis showed that the performance is significantly different between the two visualization methods, with a clear advantage for the Parallel Coordinates. *Stacked Coordinates* do not require any interaction to get an overview of frequencies, because the visual encoding of the output value on the X axis (magnitude channel) gives this information. However, the difference can be explained by the fact that users had to interact a lot, because it was not possible to select a range of years: the user had to click on each year value in order to find out how many cars belong to this value. With Parallel Coordinates, it was possible to select a range of years, and to get the count of items in the selection, at the top of the display.

For the second frequency task (T2), focused on one dimension, the completion times were shown to be more influenced by the factor `data` (1 or 2) than by the factor `visualization`. This can be explained by the difference in the two levels of the factor `data`: first, the frequency of classes in both datasets is different, with twice the amount of non-valid items in D1 than in D2. Secondly, this question is centered on only one dimension. Both visualization methods are good to assess one dimension, with the distribution being encoded on the X axis in *Stacked Coordinates*, and with a color channel in Parallel Coordinates. The completion of the task was therefore only favored by the type of data used.

The statistical analysis has shown that Parallel Coordinates are better than *Stacked Coordinates* for the visual mining task (T4), i.e. the identification of the most relevant parameter regarding the identification of the output value. The color encoding used in Parallel Coordinates, combined with the frequency encoding (i.e. jittering) seems to be more efficient than the visual representation of the distribution on the X axis (magnitude channel) in *Stacked Coordinates*, for this type of task.

For task T5 (exploration), the average number of insights was larger with Parallel Coordinates than with *Stacked Coordinates*. During their search for insights, participants interacted with the visualization method. As we reported, with *Stacked Coordinates* the overview decreased as the participant reduced his selection. The interaction problems that we already mentioned can be the cause of less insights found in the data with *Stacked Coordinates*. In the next section, we provide a list of future developments for *Stacked Coordinates* in order to solve this problem.

Despite the preference of participants for the Parallel Coordinates, our analysis of the completion times using both visualization methods showed that the difference of performance regarding

completion times was not significantly different for two tasks: the tasks T2 (frequency) and T3 (configuration).

Overall, *Stacked Coordinates* felt more difficult to use, induced more cognitive load, and looked less appealing than Parallel Coordinates. Parallel Coordinates were qualified as more playful, and looking fancier. The interaction with Parallel Coordinates requires more steps, but feels more intuitive. The horizontal layout is also a strength when compared to *Stacked Coordinates*. Three participants mentioned that the need to scroll vertically in *Stacked Coordinates* hindered the overview.

Finally, the *Stacked Coordinates* and Parallel Coordinates seem equally good for frequency tasks focused on one dimension. Parallel Coordinates are significantly better than *Stacked Coordinates* in frequency tasks implying two dimensions, and in the visual mining task focused on the search of the purest dimension. The *Stacked Coordinates* seem to be better for a configuration task, but only a larger-scale study could give more precise results to validate or reject this hypothesis.

7.6.1 Future improvements

Unlike Parallel Coordinates, *Stacked Coordinates* is a new method, with the teething problems it involves. In particular, these problems were mostly linked to the interaction (see table A.4). They can be easily corrected and do not call into question the usefulness of *Stacked Coordinates* for configuration tasks.

The following is a list of the main areas of improvement for the *Stacked Coordinates*:

- First, the need to reload the page in order to reset the system is the most cited problem (3 over 8 mentioned it). This problem can be easily overcome by adding a "reset" button to the interface.
- Next, the need to scroll is a problem to get a good overview. A few solutions exist. The simplest is to use a horizontal layout ; on a computer display, the amount of pixels is larger in the width than in the height, allowing to display more dimensions The next idea is to keep the vertical layout, but reduce the height of each dimension. Another solution is to use a dimensional reduction algorithm (PCA, C4.5, sensitivity) in order to display only the most relevant parameters.
- The third problem is that selecting values hinders the user's ability to get a good overview on the data. A solution for this would be to gray out unselected values instead of hiding them. In Parallel Coordinates, the unselected polylines are grayed out and still visible in the background.

Ranking the dimensions according to their entropy (e.g. C4.5 algorithm) in both visualization methods could also improve the performance of the user in the configuration and visual mining tasks that we defined.

Chapter 8

ELSA: a tool for architects

ELSA[1] is an exploration tool for architects. Its goal is to guide the user in the early design stages of building projects, by offering different design alternatives and their resulting environmental impact [30]. It demonstrates the usefulness of the *Stacked Coordinates* visualization approach in a real-world application. This project was conducted in the Smart Living Lab project, and is the result of a collaboration between the University of Fribourg (Human-IST Research Center[2]), the EPFL (Building 2050 Research Group[3]) and the EPFL+ECAL Lab[4]. The Building 2050 team was composed of Stefano Cozza and Thomas Jusselme (head of the project). They were in charge of generating the database that is used in the tool, and to find the optimal statistical method to cover the design space. The Human-IST Research Center team was composed of Florian Evéquoz, Denis Lalanne, Julien Nembrini, and Raphaël Tuor. Their contribution to the project consisted in developing visualization prototypes in order to find the most meaningful ways to represent the BPS data. The EPFL+ECAL Lab team was composed of Jasmine Florentine, Nicolas Henchoz and Andreas Koller. Their work consisted in finding the best ways to communicate the knowledge of the tool to the end users (the architects), to propose the best user experience possible, and to conduct surveys with architects. The final implementation of ELSA was done by tokiwi-services[5], a web and mobile development agency based in Switzerland.

In order to test the usefulness of ELSA in the early architectural design stage, a user study is currently being conducted. The results will be published in two articles, early 2017.

Integrating Life Cycle Analysis (LCA) in the early design stages is a challenge. LCA allows to to qualify the environmental performance of a building. This information determines if a building complies with environmental targets or not. The objective of the ELSA tool is to help architects gather knowledge about the impact of parameters on environmental performance indicators, widen their sphere of possibilities, give them a mean to assess the feasibility of their ideas, and to better communicate with engineers. To achieve this, the tool has to give to the architect insights about the underlying rules of a simulated building dataset. It has to provide decision support and design space exploration capabilities in order to help both architects and engineers design energy efficient buildings together. The data generated by Building Performance Simulation (BPS) are multivariate and include categorical (ordered and non-ordered) and continuous dimensions.

This section defines the context in which the project was conducted. We describe the building performance simulation database that was used, and the project requirements.

8.1 General context

Over the last decades, meeting sustainable energy consumption goals has become an essential step in the design of new buildings. The 2000 W society [28] has set specific requirements in

[1]ELSA: http://elsa.epfl.ch/
[2]Human-IST: http://human-ist.unifr.ch
[3]Building 2050 Research Group: ttp://building2050.epfl.ch/home
[4]EPFL+ECAL Lab: http://www.epfl-ecal-lab.ch
[5]tokiwi-services: http://tokiwi-services.ch

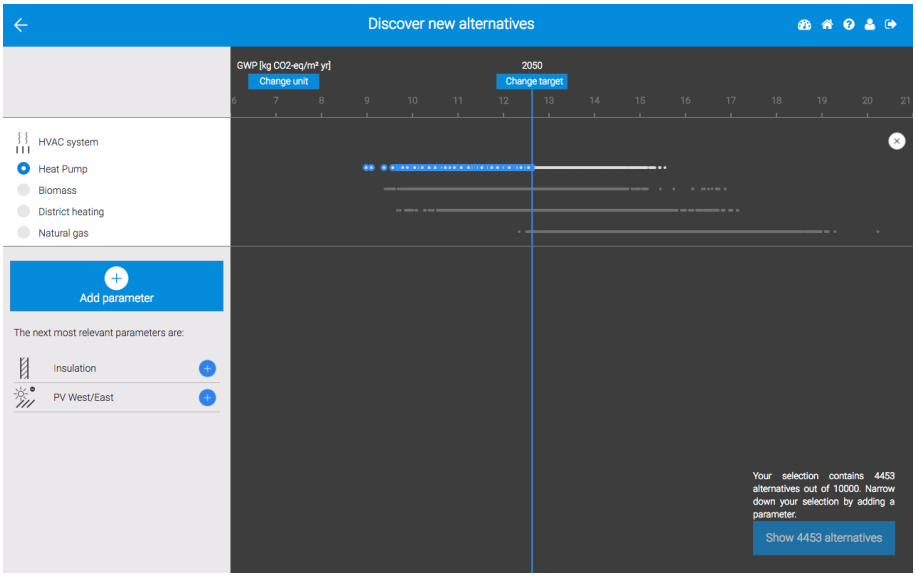


Figure 8.1: The main interface of ELSA. Each design alternative is represented by a dot. One parameter value is selected. Design: EPFL+ECAL Lab. Visualization method: Human-IST Research Center. Database: Building 2050 Research Group. From [1].

several dimensions such as Life Cycle Analysis (LCA), CO2 emissions and heating demand. Many parameters define the building, and they all influence these energy performance indicators to different extents. BPS software allows to get precise information about the energy performance indicators of a given set of parameters. The challenge is to find a meaningful way to present this essential information to the architects in charge of taking design decisions. The arrival of sustainable energy consumption goals has brought a new level of complexity to the building design. Thus, getting a clear overview on the implications of any choice during the design process has got a very tedious task – not to say impossible – for architects.

The main issue in using BPS software is that its inherent complexity is incompatible with the architect's creative process. First, it requires the user to define the building by writing a script, whereas architects mainly rely on sketches and previous references [39] to develop their ideas. Moreover, comparing a set of solutions often requires the use of batch processing that can take weeks to compute. Secondly, BPS output values require expert-level knowledge to be understood: they are hard to interpret, preventing the architects to get a clear overview on the design space. Therefore, the seamless integration of BPS software in the architect's design process is of major interest.

8.2 Goal

The goal of this project was to develop a visualization tool to give architects an overview of the simulation-generated building data. It should also give them insights about possible rules and models, while leaving room for their creativity and original ideas. It should also allow them to store their favorite design alternatives in a dedicated space. With Parallel Coordinates as a starting point, we developed a new visualization method, *Stacked Coordinates*, meant to give an overview on the range of outputs induced by every parameter value. The constructivist approach should guide the architect efficiently towards the goal of developing sustainable buildings.

Here are the requirements for the end users, architects. The tool should:

- give an overview on the data.
- improve the knowledge of the architect regarding sustainability.
- foster the creativity of the architect.

8.2.1 Energy constraints

Building's life cycle needs to be analyzed early in the design stage in order to lower the CO2 emissions and comply with standards such as SIA 380/1 [41]. These emissions can be grouped into two categories: the building's Embodied Impact (EI) (the construction stage) and Operational Impact (OI) (the use period). Alongside energy performance, other criteria like comfort, costs and construction times must be taken into account at the early design stage.

As an example of application of such constraints, the "2000-watt society" is a vision promoted by the Board of the Swiss Federal Institute of Technology in 1998 [28]. It defines energy goals that will have to be reached by 2150 in Switzerland. Intermediate targets have been derived from it for 2050 and projects like the Smart Living Lab Project [30] set the goal to respond to these. Three indicators must be improved and stay under a pre-determined threshold in order to assess the compliance with these targets.

8.2.2 Early design

Early design stages are strongly affecting final building energy performance and are thus a critical step in the building design process, as illustrated in figure 8.3. Miyamoto et al. [35] state that "in architectural design, visual tools are considered as more important than verbal ones". The architects, involved in the early building design, are at the core of this problematic: the amount of parameters and energetic constraints, and the lack of required tools integrating BPS in their workflow, slow down their creative process and force them to rely on the engineers to adapt their design choices afterwards – energy efficient buildings require a whole set of expert-level engineering skills. Architects do not have the expert knowledge in BPS but would highly benefit from a good understanding of the parameters involved in the energy performance.

The challenge of this project is to integrate BPS into architects' early design stages, in the form of a visual exploratory tool, without limiting their creativity. Thus, involving architects early in the design process should foster this energetic goal by helping them take informed decisions. As we already mentioned, and at the best of our knowledge, not a single BPS tool empowers the architect with visual parameters exploration capabilities yet, nor does stimulate his creativity.

In the design process, architects mainly rely on previous experiences and some specific references: "Many studies have, for example, demonstrated the mechanising effect of experience. Quite simply, once we have seen something done in a certain way, or done it ourselves, this experience tends to reinforce the idea in our minds and may block out other alternatives" [39]. Thus, one of the goals of the tool is to extend the vision of architects on the possible solutions, and give them new references to start their design process with. More specifically, the tool should provide architects with an overview on the design space, a ranking of parameters according to their effect on the energy performance indicators (e.g. GWP) and an understanding of the underlying rules and models influencing the building energy performance. In the end this interactive visualization would remain a creative tool: it should not limit the architect to the materials and options that were computed ahead of its use. Since not all possible parameter values are computed in the simulation, the tool should allow the user to give energy targets to the engineers at the component level. This will be done in the next steps of the development of this project.

8.2.3 Database

The BPS data to present to the architects inside the tool consists in a large set (several tens of thousands) of possible parameters combinations defining possible design alternatives, that we call references in the following. Each reference is associated to three output values representing its resulting ecological impact regarding three environmental indicators. A set of categorical dimensions defines each reference, such as the glazing ratio for a given wall or the material chosen for floors.

The building database was generated with EnergyPlus[6], a BPS software that allows to model the energy consumption for HVAC and many other building components. The final version of the database consists in 25'000 design alternatives, above and under the threshold.

[6]EnergyPlus: https://energyplus.net/

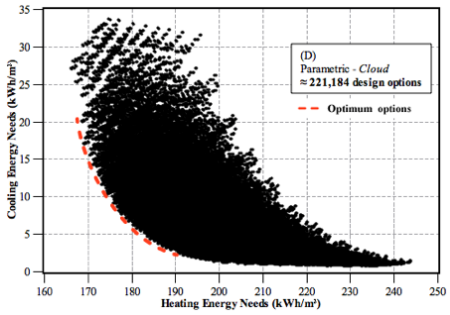


Figure 8.2: Pareto space for a set of design alternatives computed with BPS software. Valid solutions are located above the frontier. The leftmost solutions have the highest energy performance. From [38]

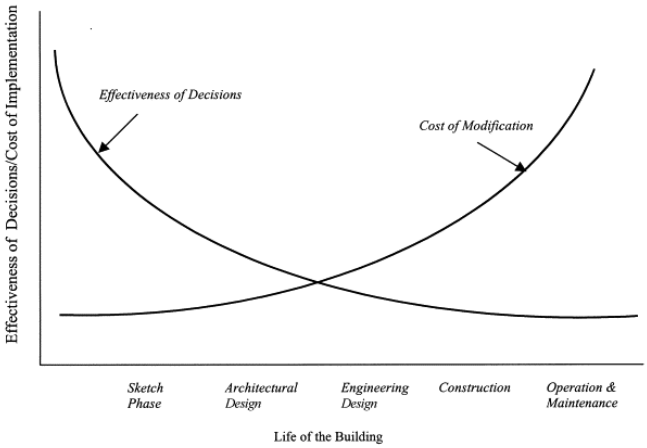


Figure 8.3: Plot showing decision costs and their impact on the performance of a building through its life cycle. From [4].

For the generation of the database, one single building shape was used, with 365.04 meter square of South facade surface and 486.72 meter square for the East and West facades.

18 categorical dimensions were chosen to define the building: South windows, North Windows, West/east Windows, Glaze Type, Frame Quality, Thermal Transmittance, PV South, PV West/east, PV roof, HVAC System, Lighting power, Appliances power, Horizontal elements, Vertical Elements, Insulation, Covering Slab, Covering External Wall and Transport Distance.

Members of the EPFL team were in charge of generating the database through a BPS software. As we explained in the first chapters of this thesis, simulated datasets induce several problems, such as overplotting and clutter [23], preventing the user to get a clear overview of the data (cf. figure 4.11). These problems had to be taken into consideration when choosing the most suited visualization method for the tool; we reviewed the techniques available to overcome them in the literature review in section 4.1.

8.3 Requirements specification

The difficulty in using these simulated datasets at the early design stage resides in the large amount of output values that BPS provide, and the technical terms used to describe the repercussions that parameters choices have on performance.

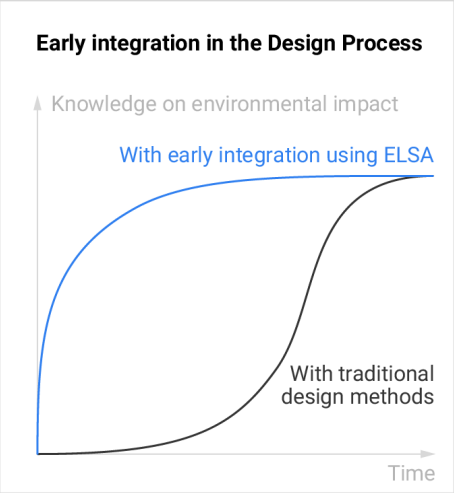


Figure 8.4: The ELSA tool allows the user to gain knowledge about the environmental impact of parameters faster than with traditional design methods. Plot made by Andreas Koller. [1]

Explore the database

The tool acts like an "experience accelerator", giving the opportunity to architects to compare many design alternatives, and learn from parameters combinations.

This is why the architects need to be able to explore the data, select parameters combinations and look at the environmental output values.

This way, in a short amount of time they will gain experience that would have required years to acquire otherwise (by completing full projects).

Communication tool

Understand the impact of components

A ranking of the parameters is required in order to give the architect the optimal sequence of selection of parameters. The relevance of parameters is computed, using the C4.5 algorithm.

Understand the limits of the database

Due to performance and time issues, the database only contains a subset of all possible design alternatives. Thus, an important requirement is to show the user that the database does only cover a subset of the possible parameter combinations. When exploring the database, architects may think that since a parameter combination is not possible in the tool, it is not feasible in reality. The technique used in ELSA consists in showing the total number of design alternatives that are available in the tool, and the number of alternatives that are currently in the selection.

Save design alternatives to favorites

The tool should enhance the communication between the members of the team: the architect needs to be able to save his favorite design alternatives, and show them to the stakeholders (clients, engineers, project managers).

As stated by Kosara et al. [31], users do not perform data analysis in one uninterrupted session: "[...] the analysis of a typical real-world data set requires many sessions, potentially spread out over a long time period. The user needs to be able to save results to continue where he or she left off as seamlessly as possible." This leverages the need for the tool to implement a "save to favorites" functionality, in order for the user to be able to save his references and to come back to them later.

Visualization challenges

As we already mentioned in the first chapters, several issues arise from the visualization of mixed multidimensional data. The *Stacked Coordinates* method solves most of them: the overplotting is reduced with the use of dots instead of lines and with the decomposition by categorical value, an identity channel is used for each categorical dimension, and the overview of the output value of each categorical value is made easier with the perpendicular axis layout.

8.4 Conclusion

We demonstrated the use of the *Stacked Coordinates* approach in a real-world application. This data visualization method can help the designer understand the impact of design choices and make informed parameters changes in order to reach the energy targets. The architect can input input parameters defining his project, assess its performance, identify those having the most impact and adapt his design accordingly. He can also save references and compare them easily.

In order to test the real usefulness of this tool, a controlled user experiment will be conducted with architecture students. The ELSA tool will be compared to the Parallel Coordinates approach, in a set of tasks (configuration, visual mining, knowledge about LCA) and questionnaires (Self-Assessment Manikin [8], System Usability Scale [10]). Two articles will explain the results of the study. Here is a set of hypotheses that we will try to verify:

- Using a tool to visualize BPS data increases significantly the user performance in the knowledge test about LCA.
- There is a significant difference in time performance between the ELSA tool and Parallel Coordinates in the design tasks.
- Both tools increase the feeling of control and/or happiness of the architect (regarding the Self-Assessment Manikin results).

Chapter 9

Conclusion

9.1 Wrap up

The goal of this master thesis was to improve the user performance for low level tasks, such as frequency- and similarity-based tasks, and a high level configuration task, on mixed multivariate datasets.

We proposed and tested two new visualization prototypes meant to correct some of the problems arising from mixed multidimensional data: *Parallel Bubbles*, and *Stacked Coordinates*.

After giving a definition of mixed multidimensional data, and underlining their importance in the fields involving design tasks, we noticed that categorical and continuous dimensions convey completely different information, and that visual encodings have to be chosen carefully and separately for both types of dimension. To respect the *expressiveness* principle means to choose a visual encoding that conveys only the information that is in the data. For example, using a magnitude channel for a categorical dimensions is a typical example of the violation of this principle, since it suggests a ranking of the values by the user. To respect the *effectiveness* principle means to encode the most important dimensions with the most effective visual channels. The importance of dimensions is defined by the tasks that the user wants to do.

We defined the tasks that data analysts and designers usually perform on this kind of data: frequency- and similarity-based tasks, and configuration tasks. We then reviewed the main visualization methods for multidimensional data, and listed their strengths and weaknesses regarding the number of dimensions and data items they can display, and addressing their expressiveness and effectiveness. We gave an overview of the problems raised by mixed multidimensional datasets. The primary sources of problems are: the use of the same encoding for a mix of categorical and continuous dimensions inside the same visualization, the high number of dimensions, and the large number of data items. We listed the available visual enhancement methods and assessed their ability to solve the problems mentioned above.

We also learned the importance of following a strict protocol during controlled experiments in order to get robust results. The next section describes the outcomes of the two user studies that we conducted.

9.2 Contributions

Our two user studies follow the guidelines for future research stated by Johansson et al. [29]: "existing approaches need to be further evaluated in order to find out which techniques have potential and which should be avoided."

9.2.1 *Parallel Bubbles*

Our first study focused on the influence of two frequency encodings in Parallel Coordinates on the user performance. It confirmed our hypothesis: *Parallel Bubbles are a good compromise in terms*

of user performance in frequency and similarity tasks, when compared to Parallel Coordinates and Parallel Sets.

The results are coherent with Stevens' Psychophysical Power law [50]: not all visual channels are equally distinguishable, and we perceive them with different levels of accuracy. The Parallel Sets offer a significantly better performance in all three tasks, probably because the visual encoding of frequency added to *Parallel Sets* is the most accurate. *Parallel Bubbles*' visual encoding of frequency should use a different channel in order to match more closely the visual perception to its true value to. An idea would be to make the radius of each bubble proportional to the frequency of items.

We also noted that Parallel Coordinates deliver the worst performance in all tasks, including similarity tasks. Moreover, the more correlated the dataset, the better the performance in all tasks.

9.2.2 *Stacked Coordinates*

In our second user study, we tested the effectiveness of *Stacked Coordinates* in data analysis and configuration tasks on a mixed multivariate dataset – using the X axis (a magnitude channel) to encode the output dimension should improve the user performance in configuration, frequency and visual mining tasks.

Our main hypothesis was that *"Stacked Coordinates provide a significantly better performance than Parallel Coordinates in configuration tasks"*. After analyzing the completion times, we can tell that the *Stacked Coordinates* provide a better performance than Parallel Coordinates, but the difference is not significant.

For our second hypothesis, *"Stacked Coordinates allow a better time performance on frequency and visual mining tasks"*, we have the following conclusions: *Staked Coordinates* and Parallel Coordinates are equally good for frequency tasks focused on one dimension (T2), but Parallel Coordinates are significantly better than *Stacked Coordinates* in frequency tasks involving two dimensions (T1) and in the visual mining task focused on the search of the purest dimension (T4).

We noted the importance of the dataset in such user experiments: for the task T2 (frequency), the factor `data` had more influence on the user performance than the factor `visualization`. We explain this by the difference of distribution of classes "non-valid" and "valid" in both datasets: the dataset having the best balance between "non-valid" and "valid" classes gave the shortest completion times.

Regarding the exploration task (T5), participants using the Parallel Coordinates gave more insights than with *Stacked Coordinates*. We noted that it is impossible to interpret the insights given by another person. We should have asked participants to classify their insights themselves, at the end of the task: an insight can be classified in several categories (e.g. dependency or correlation) depending on the way it is interpreted.

Overall, *Stacked Coordinates* felt more difficult to use, induced more cognitive load, and looked less appealing than Parallel Coordinates. Parallel Coordinates were qualified as more playful. The interaction with Parallel Coordinates requires more steps, but feels more intuitive for the user. We can confirm the words of Johansson et al. : "[...] a Parallel Coordinates display that is visually appealing with intuitive interactions would attract the attention of more users and stimulate uptake and usage." [29]

9.3 Future works

The first user study should be repeated with another set of tasks, focused on the continuous axis. The magnitude channel used on the *Parallel Bubbles*' visual encoding of frequency should be modified: the radius of a bubble should be linearly proportional to the number of items belonging to the corresponding category.

The second user study should be repeated with a larger number of participants, in order to get more robust results. As we already mentioned, the teething problems of *Stacked Coordinates* were mostly related to the interaction (reset the selection, scrolling problems), and solving them should improve the user experience.

Regarding the types of tasks, Johansson et al. [29] state "It is also necessary to study more advanced relationships and not only focus on clusters, correlations and outliers. A good understanding of how non-linear relationships are interpreted in Parallel Coordinates would be valuable and could potentially increase the use of Parallel Coordinates."

Longitudinal studies are another track for further works [29]: "Concerning Parallel Coordinates there are none." Such studies are time consuming, but would allow users to think of improvements, customization and new ways of interaction.

Appendices

Appendix A

User study 2: Parallel Coordinates vs. *Stacked Coordinates*

A.1 Qualitative feedback

A.1.1 Parallel Coordinates

Usefulness feedback	**Count**	**Positive (+) or Negative (–)**
Identifying tendencies (correlations, similarities) is easy	3	+
Identifying distribution of values regarding the output is faster	3	+
Identifying a single data item is easy	2	+
It is more playful: we can select several options at the same time	2	+
It requires really directed questions	1	–
It is easier to compare the output of all dimensions	1	+
It provides a better overview, even while selecting values	1	+
It is better for data exploration	1	+
It gives a more precise output value of one category	1	+

Table A.1: Feedback of participants regarding the usefulness of the Parallel Coordinates.

Usability feedback	**Count**	**Positive (+) or Negative (–)**
More intuitive to use	3	+
The horizontal layout is easier to read	2	+
It requires more interactions (clicks, brushing) to verify an intuition	1	–
It is possible to select a range of values	1	+
The visualization looks more attractive	1	+

Table A.2: Feedback of participants regarding the usability of the Parallel Coordinates.

A.1.2 Stacked Coordinates

Usefulness feedback	**Count**	**Positive (+) or Negative (–)**
It is better to assess if a category is above or under the threshold	2	+
It is quicker to see the distribution of one category	2	+
It is easier to estimate the output ranges of each category	2	+
It is useful to confirm an existing hypothesis	1	+
The overview gets worse when several values are selected	1	–

Table A.3: Feedback of participants regarding the usefulness of the *Stacked Coordinates*.

Usability feedback	**Count**	**Positive (+) or Negative (–)**
It requires to scroll to get an overview	3	–
It requires to reload the page to reset the selection	2	–
It feels less easy to use	1	–
It does not require clicks to see the distribution	1	+
It requires more cognitive load	1	–
Dots are too small to assess the distribution	1	–
Rectangles are too thin in width	1	–

Table A.4: Feedback of participants regarding the usability of the *Stacked Coordinates*.

Bibliography

[1] ELSA: Exploration tool for sustainable architecture. `http://elsa.epfl.ch`. Accessed: 2016-11-22.

[2] Parallel Sets: A visualisation technique for multidimensional categorical data. `https://www.jasondavies.com/parallel-sets/`. Accessed: 2016-10-23.

[3] Understanding area based plots: Trellis displays. `http://www.theusrus.de/blog/understanding-area-based-plots-trellis-displays/`. Accessed: 2016-10-23.

[4] Mohammad Saad Al-Homoud. Computer-aided building energy analysis techniques. *Building and Environment*, 36(4):421–433, 2001.

[5] Mohini P Barde and Prajakt J Barde. What to use to express the variability of data: Standard deviation or standard error of mean? CONCEPT OF SD AND SEM. *Perspectives in Clinical Research*, 3(3).

[6] R Becker, W S Cleveland, and M J Shyu. The visual design and control of trellis displays. *J Comput Graphical Stat*, 6(2):123–155, 1996.

[7] Paul E Black. Manhattan distance. *Dictionary of Algorithms and Data Structures*, 18:2012, 2006.

[8] Margaret M Bradley and Peter J Lang. Measuring emotion: the self-assessment manikin and the semantic differential. *Journal of behavior therapy and experimental psychiatry*, 25(1):49–59, 1994.

[9] Bertjan Broeksema, Alexandru C Telea, and Thomas Baudel. Visual analysis of multi-dimensional categorical data sets. *Computer Graphics Forum*, 32(8):158–169, 2013.

[10] John Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.

[11] E Cantú-Paz, Shawn Newsam, and Chandrika Kamath. Feature selection in scientific applications. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 788–793, 2004.

[12] Chun-houh Chen, Wolfgang Hrdle, and Antony Unwin. *Handbook of Data Visualization (Springer Handbooks of Computational Statistics)*. 2008.

[13] M C F de Oliveira and H Levkowitz. From visual data exploration to visual data mining: A survey. *IEEE transactions on visualization and computer graphics*, 9(3):378–394, 2003.

[14] Sara Johansson Fernstad and Jimmy Johansson. A Task Based Performance Evaluation of Visualization Approaches for Categorical Data Analysis. *2011 15th International Conference on Information Visualisation*, pages 80–89, jul 2011.

[15] Ying-Huey Fua Ying-Huey Fua, M.O. Ward, and E.a. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. *Proceedings Visualization '99 (Cat. No.99CB37067)*, pages 43–508, 1999.

[16] Zhao Geng, Zhenmin Peng, Robert S. Laramee, Rick Walker, and Jonathan C. Roberts. Angular histograms: Frequency-based visualizations for large, high dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2572–2580, 2011.

[17] Martin Graham and Jessie Kennedy. Using curves to enhance parallel coordinate visualisations. *Proceedings of the International Conference on Information Visualisation*, 2003-January(August 2003):10–16, 2003.

[18] Anthony M. Graziano and Michael L. Raulin. *Research Methods - A Process of Inquiry*. 2010.

[19] Lane Harrison, Fumeng Yang, Steven Franconeri, and Ronald Chang. Ranking Visualizations of Correlation Using Weber's Law. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1943–1952, 2014.

[20] J.A. Hartigan. Printer graphics for clustering. *Journal of Statistical Computation and Simulation*, 4(3):187–213, 1975.

[21] Susan L Havre, Anuj Shah, Christian Posse, and Bobbie-Jo Webb-Robertson. Diverse information integration and visualization. volume 6060, pages 60600M–60600M–11, 2006.

[22] Jeffrey Heer and Michael Bostock. Crowdsourcing Graphical Perception: Using Mechanical Turk to Assess Visualization Design. *Proceedings of the 28th Annual Chi Conference on Human Factors in Computing Systems*, pages 203–212, 2010.

[23] J Heinrich and D Weiskopf. State of the Art of Parallel Coordinates. *Eurographics*, pages 95–116, 2013.

[24] A Inselberg and Bernard Dimsdale. Parallel coordinates: a tool for visualizing multidimensional geometry, 1990.

[25] Alfred Inselberg. *Parallel Coordinates: Visualization, Exploration and Classification of High-Dimensional Data*, pages 643–680. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[26] Dean F. Jerding and John T. Stasko. The information mural: A technique for displaying and navigating large information spaces. *IEEE Transactions on Visualization and Computer Graphics*, 4(3):257–271, 1998.

[27] J.Han, J.Pei, M.Kamber. *Data Mining: Concepts and Techniques*, volume 3. 2012.

[28] E. Jochem et al. *A white book for R&D of energy-efficient technologies. Steps towards a sustainable development.* Novatlantis, Zurich, March 2004.

[29] Jimmy Johansson and Camilla Forsell. Evaluation of Parallel Coordinates: Overview, Categorization and Guidelines for Future Research. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):579–588, 2016.

[30] Thomas Jusselme. Building 2050 - Research program. Technical report, EPFL Fribourg, 2015.

[31] Robert Kosara, Fabian Bendix, and Helwig Hauser. Parallel sets: Interactive exploration and visual analysis of categorical data. In *IEEE Transactions on Visualization and Computer Graphics*, volume 12, pages 558–568. IEEE, jul 2006.

[32] Jing Li, Jean-Bernard Martens, and Jarke J van Wijk. Judging correlation from scatterplots and parallel coordinate plots. *Information Visualization*, 9(1):13–30, 2008.

[33] A.R. Martin and M.O. Ward. High Dimensional Brushing for Interactive Exploration of Multivariate Data. *Proceedings Visualization '95*, pages 271–278, 1995.

[34] The Mathworks, Inc., Natick, Massachusetts. *MATLAB version 8.5.0.197613 (R2015a)*, 2015.

[35] Ayu Miyamoto, Tam Nguyen Van, Damien Trigaux, Karen Allacker, and Frank De Troyer. Visualisation Tool To Estimate the Effect of Design Parameters on the Heating Energy Demand in the Early Design Phases. *Plea*, (August 2016), 2015.

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

[37] Tamara Munzner. *Visualization Analysis and Design*. CRC Press, 2014.

[38] Emanuele Naboni, Yi Zhang, Alessandro Maccarini, Elian Hirsh, and Daniele Lezzi. Extending the use of parametric simulation in practice through a cloud based online service. *Proceedings of first IBPSA-Italy conference BSA 2013*, pages 105–112, 2013.

[39] Patrick Purcell. *How designers think*, volume 2. 1981.

[40] J Ross Quinlan. *C4.5: Programs for Machine Learning*, volume 1. 1992.

[41] SIA Recommandation. 380/1 (1988) l'énergie dans le bâtiment. *Société suisse des ingénieurs et des architectes, Zurich, Suisse*.

[42] Nancy Rodriguez and Rodriguez Nancy. A Survey of Muldimensional data Visualization Techniques. (January), 2016.

[43] Geraldine E. Rosario, Elke A. Rundensteiner, David C. Brown, and Matthew O. Ward. Mapping nominal values to numbers for effective visualization. In *Proceedings - IEEE Symposium on Information Visualization, INFO VIS*, pages 113–120, 2003.

[44] Matthias Schonlau. Visualizing categorical data arising in the health sciences using hammock plots. In *Proceedings of the Section on Statistical Graphics, American Statistical Association*, 2003.

[45] Matthias Schonlau. Visualizing categorical data arising in the health sciences using hammock plots. *Proceedings of the Section on Statistical Graphics, American Statistical Association*, 2003.

[46] David J Sheskin. Handbook of parametric and nonparametric statistical procedures. *Technometrics*, 46:1193, 2004.

[47] Ben Shneiderman. Dynamic queries for visual information seeking. *IEEE Software*, 11(6):70–77, 1994.

[48] Ben Shneiderman. The eyes have it: a task by data type taxonomy for informatio nvisualizations. *Proceedings 1996 IEEE Symposium on Visual Languages*, pages 336–343, 1996.

[49] Drew Skau and Robert Kosara. Arcs , Angles , or Areas : Individual Data Encodings in Pie and Donut Charts. 35(3), 2016.

[50] SS Stevens. Laws that govern behavior.(book reviews: Psychophysics. introduction to its perceptual, neural, and social prospects). *Science*, 188:827–829, 1975.

[51] Soon Tee Teoh and Kwan-Liu Ma. PaintingClass: interactive construction, visualization and exploration of decision trees. *Star*, pages 667–672, 2003.

[52] Antony Unwin. Multivariate visualization methods. In *Encyclopedia of Database Systems*, pages 1866–1870. 2009.

[53] Peter H Westfall. Kurtosis as Peakedness, 1905 - 2014. R.I.P. *The American statistician*, 68(3):191–195, 2014.

Acronyms

BPS Building Performance Simulation. 5, 19, 72–75, 77

EI Embodied Impact. 74

GWP Global Warming Potential. 74

HVAC Heating, Ventilation, Air Conditioning. 74

LCA Life Cycle Analysis. 19, 72, 77

OI Operational Impact. 74