Markus Ruplitsch, BSc.

Interactive Topological Data Analysis

Master's Thesis

to achieve the university degree of

Master of Science

Master's degree programme: Computer Science

submitted to

Graz University of Technology

Supervisor

Dipl.-Ing. Dr.techn. Reinhold Preiner

Institute of Computer Graphics and Knowledge Visualization
Head: Univ.-Prof. Dipl-Volksw. Dr.rer.nat. M.Sc. Tobias Schreck

Graz, July 2022

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Date	Signature

Abstract

Analyzing large high-dimensional datasets has found applications in most fields of research and has drastically increased in importance over the last few decades. There are many ways to analyze large data heaps and especially machine learning has attracted most of the public's attention. However, over the last few decades, other potentially promising data analysis methods have been developed. Topological data analysis is one such analysis method. It provides a general framework to analyze the structure of high-dimensional data and extract information from this structure. Techniques used in topological data analysis include, but are not limited to, manifold estimation, mode estimation, ridge estimation, clustering and dimension reduction. This paper introduces a new topological data analysis method that allows users to interactively reduce the number of dimensions of a given data set, estimate manifolds, parametrize subspaces and inspect subsets of the original data. This new method was tested by developing a highly performant software prototype and conducting a user study on its applicability and usability. The results showed that this novel, interactive approach can enable users to iteratively simplify a given data set and present information about the data's structure.

Contents

Abstract iii

1 Introduction 1
- 1.1 An Overview of Topological Data Analysis . . . 2
- 1.2 Challenges . . . 5

2 Related Work 7
- 2.1 Visual Topological Data Analysis . . . 8
- 2.2 Scatter Plots . . . 13
  - 2.2.1 Scatter Plot Matrices . . . 14
- 2.3 Parallel Coordinate Plots . . . 16
- 2.4 Hybrid Techniques . . . 19
- 2.5 Unsolved Problems . . . 21

3 Concept and Implementation 25
- 3.1 Core Ideas . . . 25
  - 3.1.1 Merging and Unmerging of Dimensions . . . 26
  - 3.1.2 Unrolling of Embedded Dimensions . . . 27
  - 3.1.3 Tracking Data Provenance . . . 29
- 3.2 Supplementary functions . . . 30
- 3.3 Usability . . . 35
- 3.4 Languages and Frameworks . . . 36

4 Results 39
- 4.1 Use Cases . . . 39
  - 4.1.1 Structures in Two Dimensions . . . 39
  - 4.1.2 Structures in Three Dimensions . . . 40
  - 4.1.3 Structures in Higher Dimensions . . . 41

5 User Study **45**
5.1 Setup 45
5.2 Study Results 47

6 Conclusion **57**
6.1 Results 57
6.2 Future Work 58
6.3 Further Reading 60

Bibliography **61**

List of Figures

1.1 The Swiss-roll-data set. The data points lay on a two-dimensional plane within the three-dimensional space. This image was taken from Wasserman (2018) 3

1.2 (a) A two-dimensional density function with a one-dimensional ridge (highlighted in blue). While some points that make up this ridge are local maxima, most are not. (b) Data with higher density along a set of intersecting lines (ridges). Similar structures can be found in galaxy superclusters. Both images were taken from Wasserman (2018) and originally published in Genovese et al. (2014). 4

1.3 An intuitive example of persistent homology. The number of clusters varies depending on the radius with which the data points are visualized. Clusters which are more clearly separated, are visible for a wider range or radii. Similar to the number of clusters, the hole in the middle is not visible for all radii. This image was taken from Wasserman (2018). . 5

2.1 A Star Coordinate visualization of an 8 dimensional car data set. The difference between these two images is how much focus is placed on the car's origin. This image was taken from Kandogan (2000). 10

2.2 Five Star Glyphs arranged on their pivot axis. This image was taken from Fanea, Carpendale, and Isenberg (2005). 12

2.3 A Scatter Plot Matrix of a 7-dimensional car data set. This image was taken from Elmqvist, Dragicevic, and Fekete (2008). 15

2.4 A Parallel Coordinate Plot with six dimensions. This image was taken from Inselberg and Dimsdale (1990). 17

2.5 Common patterns in two dimensions visualized via a scatter plot (top) or a PCP (bottom). This image was taken from Heinrich and Weiskopf (2013). 17
2.6 A simple example of how a normal PCP can be converted to a three-dimensional PCP. Depending on which dimensions are used to create 2D scatter plots, the resulting line patterns change. This image was taken fromJohansson Westberg, Forsell, and Cooper (2013). 20
2.7 The top image shows a normal PCP. The bottom image shows a PCP with scatter points between some dimensions. This image was taken from Yuan et al. (2009). 22

3.1 The standard PCP view, that is shown upon loading a data set. The order of the axes is the same order as in the imported CSV file. 26
3.2 A PCP layer with one-, two- and three-dimensional scatter plots. The three-dimensional scatter plot has been rotated to better showcase its three-dimensional nature. 27
3.3 A polynomial function has been used to calculate the least-squares error in the 2D scatter plot. All data points are then projected onto this polynomial, which results in the new 1D scatter plot in the second PCP layer. 28
3.4 Three PCP layers, created by unrolling embedded dimensions two consecutive times. When the user hovers over the 2D scatter plot to the right in the middle layer, its parents that were merged to create it are highlighted in red and the 3D scatter plot that contains the unrolled dimension is highlighted in blue. All parent-child relations are visualized via the connecting lines. 31
3.5 The user's mouse is hovering over a 1D scatter plot. The recommendation pop-up initially shows the four 2D scatter plots with the best polynomial function fit. More possibilities can be viewed when the slider is moved to the right. 32
3.6 After encircling a subset of the data, a new PCP layer is created in which only the selected points are displayed. In order to use the entire space available, the data points in the upper layer are spread out. 34

3.7 The user interface of the current prototype. The visualized data set is the popular 'iris' machine learning data set (Anderson, 1935). 1. Opacity Sliders for the Points (especially in 2D and 3D) and lines. 2. Different Tools to perform actions. From top left to bottom right: Curve Sketch Tool, Function Fit Tool, Rotate Tool, Selection Tool, 3D Projection Tool. 3. Interaction points at the bottom of each scatter plot. Actions such as dragging, merging and unmerging can be performed by clicking this point. 35

4.1 While there are no clear polyline patterns visible, the color gradient defined by 'dim_0' can be found again in 'dim_4'. This suggests that there exists a correlation between these two dimensions. 40

4.2 A synthetically created data set with three dimensions. (a) After selecting only the data points of the inner cluster, the data is parametrized by sketching along the circle to reduce the number of dimensions. A merge action with the third dimension reveals a smiley face. (b) An alternative way to discover the smiley face. All three dimensions are merged in the first layer. The data points of the inner cylinder are selected, and then unrolled to reveal the smiley face. 42

4.3 An analysis of a synthetically created data set with ten dimensions. The dimensions 0, 1, 4 and 7 form a 4D dragon shape. This dragon shape appears to be curled into a spiral in the dimensions 0 and 4. There appear to be two clusters in dimensions 3 and 6 (third layer). The smaller cluster has a strong quadratic correlation between dimensions 2 and 8 (top layer). Dimensions 5 and 9 only contain white noise. . . 43

5.1 The dataset used to show participants the tool's various functions. 46

5.2 The second dataset used during the study. Users used this dataset to try out the tool on their own. This is a version of the popular 'iris' dataset (Anderson, 1935) with 3 additional artificially created dimensions. 47

5.3 Histograms of the results of the questions concerning the participants' previous knowledge and the usefulness of the prototypes' functionalities. For the first three questions, an answer of 1 represents "Not at all familiar" and an answer of 5 represents "Very familiar". For the last two questions, an answer of 1 represents "Not at all useful" and an answer of 5 represents "Very useful". 53
5.4 Histograms of the results of 5 questions concerning the usefulness of the prototype's functionalities. An answer of 1 represents "Not at all useful" and an answer of 5 represents "Very useful". 54
5.5 Histograms of the results of the first 6 questions of the System Usability Scale. An answer of 1 represents "Strongly disagree" and an answer of 5 represents "Strongly agree". The scores show that overall the system was relatively easy to use. 55
5.6 Histograms of the results of the last 4 questions of the System Usability Scale. An answer of 1 represents "Strongly disagree" and an answer of 5 represents "Strongly agree". The scores show that overall the system was relatively easy to use. 56

1 Introduction

Over the last years and decades the computational power of devices has increased drastically, and with it the rate at which data is collected has also increased. Not just the number of data points has increased, but also the number of attributes has grown and analyses of high-dimensional data has become an important part of many commercial and scientific application areas. Because of this, it is now often the case that answers to problems can be found in a large heap of high-dimensional data, where only a few of the dimensions are actually relevant. As argued by Carlsson (2009), geometrical and topological features can be useful to create a better understanding of the data.

Topological data analysis (TDA) is a recent field of research, that only tracks back about 20 years to the early works on persistent homology by Edelsbrunner et al. (2000) and Zomorodian and Carlsson (2005). TDA is a type of data analysis that uses techniques from the field of topology to extract information from high-dimensional and possibly incomplete and noisy data sets. While TDA has found some practical use cases in some other fields of research, such as astrophysics (Worsley, 1995) or protein folding research (Kovacev-Nikolic et al., 2016), TDA still has not found much popularity for general purpose data analysis.

This thesis concerns itself with the field of TDA and its applications. While Chapter 2 will focus on the current state of this research field, Chapter 3 will explain the new approach of interactively exploring high-dimensional data sets, that was developed as part of this thesis. Chapters 4 and 5 will then go into detail about how well this new approach worked.

1.1 An Overview of Topological Data Analysis

While Some definitions like Munch (2017) focus on the practical applications of TDA and define it as a "collection of powerful tools", this thesis will use the definition of Wasserman (2018), who states that "Topological data analysis (TDA) refers to statistical methods that find structure in data". While the exact definition varies, most agree that the goal of TDA is to analyze data by finding topological structures within the data. "These structures include clustering, manifold estimation, nonlinear dimension reduction, mode estimation, ridge estimation and persistent homology" (Wasserman, 2018). While TDA is not limited to these kinds of analyses, this section only serves the purpose of providing the reader with an overview of some ideas of TDA and will therefore focus mainly on the aforementioned types of structures.

Clustering is one of the simplest examples of TDA and the easiest to understand. According to Kaufman and Rousseeuw (2009): "Basically, one wants to form groups in such a way that objects in the same group are similar to each other, whereas objects in different groups are as dissimilar as possible". There are many algorithms to find clusters in data, which often differ in how they define "similar" and "dissimilar". While relatively simple, clustering is one of the most popular and powerful techniques to analyze data because it generates results that are intuitively understood by humans, as it is in our nature use categorization and classification to simplify and streamline our perception (Macrae and Bodenhausen, 2000).

An intuitive approach to finding meaningful structure in high-dimensional data is to reduce the number of dimensions. This is usually done by finding sets of dimensions which are correlated strongly. In case of a linear correlation this can be done, for example via Principal Component Analysis (Wold, Esbensen, and Geladi, 1987). While PCA belongs to the group of linear dimension reduction techniques, there are also many nonlinear approaches (DeMers and Cottrell, 1992, Teh and Roweis, 2002 and Brand, 2002, just to name a few). Sometimes, dimensionality reduction can also be done by finding a manifold of lower dimension that the data points lay on (Tenenbaum, V. d. Silva, and Langford, 2000, McInnes, Healy, and Melville, 2018). This, however, is oftentimes computationally expensive, even under

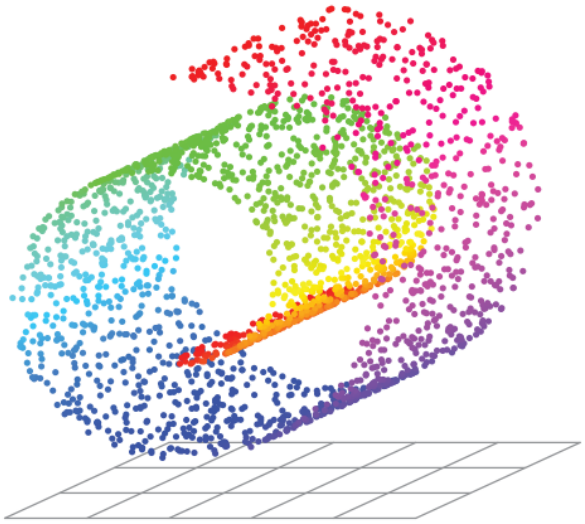

Figure 1.1: The Swiss-roll-data set. The data points lay on a two-dimensional plane within the three-dimensional space. This image was taken from Wasserman (2018)

mild assumptions. An example of such a data set can be seen in Figure 1.1. This is also called manifold estimation. In Chapter 3, a novel approach to reducing dimensionality via manifold estimation is presented.

Mode estimation is another way of finding clusters of data points. "The idea is to find modes of the density and then define clusters as the basins of attraction of the modes" (Wasserman, 2018). Another intuition is that mode estimation interprets a given data set as a probability density function and then tries to find local maxima. These maxima are called modes and can be used to define a center for a data cluster. For a more rigorous and mathematical definition of some of these algorithms see Arias-Castro, Mason, and Pelletier (2016), Chacón (2012), Chacón and Duong (2013), Comaniciu and Meer (2002) and Cheng (1995).

Similar to mode estimation, ridge estimation describes a data set as a probability density function and tries to find critical points. While mode estimation is used to find local and global maxima, ridge estimation merely attempts to locate low-dimensional ridges where the density has a relatively high local concentration. Examples of ridge estimation can be seen in Figure 1.2. For further details see Genovese et al. (2014) or Ozertem and

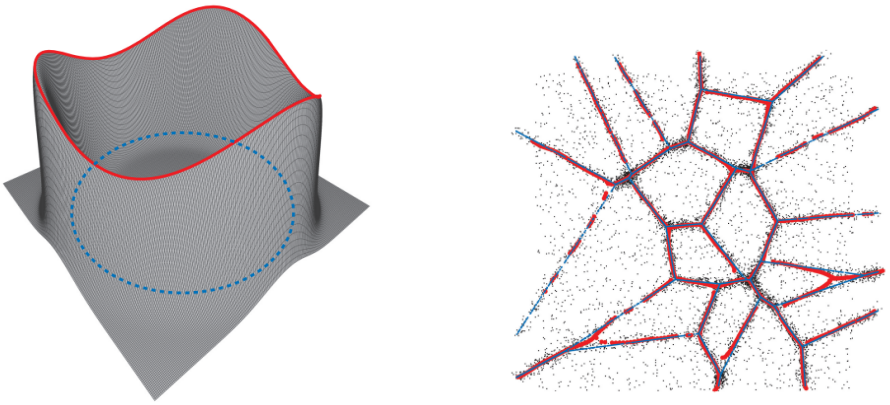

Figure 1.2: (a) A two-dimensional density function with a one-dimensional ridge (highlighted in blue). While some points that make up this ridge are local maxima, most are not. (b) Data with higher density along a set of intersecting lines (ridges). Similar structures can be found in galaxy superclusters. Both images were taken from Wasserman (2018) and originally published in Genovese et al. (2014).

Erdogmus (2011) for a more precise and rigorous definition of ridges.

Persistent homology is the last type of structure mentioned by Wasserman (2018) and, according to him, the branch that gets the most attention and is viewed by some as synonymous with TDA. Persistent homology is a method of finding features or structures that are visible throughout a large range of scopes. For example, at a small scale, every single data point could be considered a cluster. However, this is usually not a useful way of looking at data, which is why in most cases singular data points that are dissimilar to all others are classified as outliers. Similarly, one could view the data at a scale so large that the entire data set could be seen as a single large cluster, which is not useful for most cases either. Usually, the scope is then set as a parameter but this comes with problems of its own. For example, one parameter might yield many more clusters than another. Now a metric would have to be chosen to decide which of the two results is more 'useful'. By contrast, persistent homology tries to find structures that are visible for not just a small fraction of all possible scopes, but instead for a wide range. For our example, this would mean that, when we change the scope through which we look at our data, the most 'useful' result is the one where

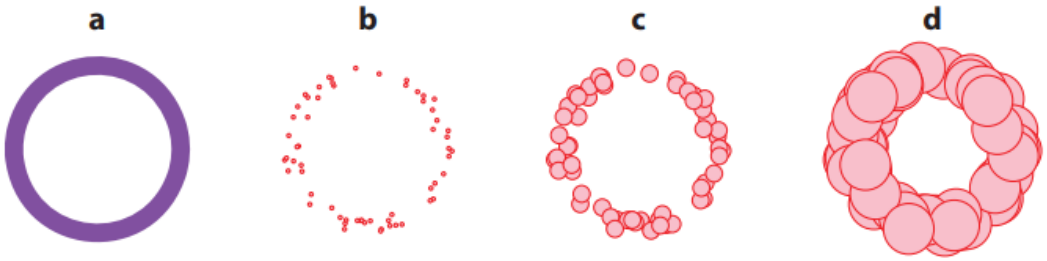

Figure 1.3: An intuitive example of persistent homology. The number of clusters varies depending on the radius with which the data points are visualized. Clusters which are more clearly separated, are visible for a wider range or radii. Similar to the number of clusters, the hole in the middle is not visible for all radii. This image was taken from Wasserman (2018).

the number of clusters does not change (persist) for the largest interval of changing scopes. A visual example can be seen in Figure 1.3.

1.2 Challenges

Because topological data analysis is concerned with high-dimensional data, the 'curse of dimensionality' arises. This phenomenon, originally coined by Bellman and Kalaba (1959), describes the idea that with an increasing number of dimensions, many problems have to be considered that would not be present in lower dimensions.

For example, an increase in the number of dimensions leads to such a fast increase in the volume of the space, that the data becomes sparse. Intuitively, a set of data points might be similar to each other in some number of dimensions. However, when adding more dimensions, they are likely to be dissimilar in at least some of these dimensions. Conversely, data points that are dissimilar from each other in a few dimensions might be correlated in other dimensions. This leads to problems when trying to find clusters, modes or ridges in extremely high-dimensional data, as all data points appear to be more or less equidistant from each other.

While costly, this problem can be overcome by gathering more data to more

densely populate the data space. However, this also leads to another problem. While, as mentioned before, the computing power of modern processors has increased drastically, there are still limits. Data sets with hundreds or thousands of dimensions and many millions of data points can only be subjected to a limited subset of analysis methods with sub-exponential runtime.

Another factor to consider is the quality of data. Of course, there are TDA methods which are quite robust and yield high quality results, even with outliers present. However, according to Wasserman (2018), some TDA methods are not as robust, such as Tenenbaum, V. d. Silva, and Langford (2000). Once again, this problem can be tackled in numerous ways, for example cleaning the data beforehand or fine-tuning parameters, but naturally the number of outliers or incorrect samples will usually increase with the number of dimensions and data points. Detecting outliers and cleaning data has always been a challenge in data analysis and are also important tasks for topological data analysis.

This thesis will introduce a new interactive TDA method that allows users to find low-dimensional manifolds in high-dimensional data. This new approach enables users to iteratively reduce the number of dimensions, remove outliers, and parametrize subspaces.

2 Related Work

Where the previous chapter gave an overview of topological data analysis as a whole, this chapter will focus on visual methods that allow users to find patterns in data. It also serves the purpose of providing context for the developed prototype, to explain which visualization techniques already exist and what their strengths and weaknesses are.

Since the underlying data usually lives in a high-dimensional space, there is no immediate way to visualize all dimensions simultaneously. A common solution to this problem is to project the high-dimensional points to a lower-dimensional embedding, usually 2D, which can be viewed on a screen or sheet of paper. These static images, however, make it difficult to get a good understanding of the high-dimensional structure of the data. Therefore, many visualization techniques are most useful when presented by an interactive visualization tool through which users can manipulate parameters of the visualization. This is especially important for the exploration of such data, which users typically start with no prior understanding of the data's structure.

Another point to consider is that, while many interactive visualizations can solely be modified manually, the changing of parameters can oftentimes be supported by automated systems to reduce the amount of data users have to process.

Visual approaches to TDA also solve some of the previously mentioned problems. For example, humans are notoriously good at finding outliers, when equipped with the right tools. Humans are also great at finding clusters or correlations, which cannot be described as a simple scalar function, such as spirals or concentric circles. These tasks can be very computationally expensive when encountered in high-dimensional settings, and leveraging the strengths of the human mind can be beneficial when analyzing data. It

is also the human mind which is the recipient of the discovered information and it is often helpful to support human cooperation throughout the entire data analysis process.

2.1 Visual Topological Data Analysis

Most visualization techniques do not explicitly focus on topological data analysis and the field of visual TDA strongly overlaps with the field of high-dimensional data visualization. Because of its relevance to the developed prototype, a particular focus is laid on scatter plots and Parallel Coordinate Plots in Sections 2.2 and 2.3.

Before discussing specific visualization techniques, it is also worth mentioning that all the following techniques can be use in conjunction with dimension reduction (DR) techniques and many of the mentioned papers even explicitly mention some form of DR that is used as a first step to pre-process the data for later visualization (Jäckle et al., 2017, Yuan et al., 2009). In fact, it is often the case that when the number of dimensions becomes large enough, some form of DR becomes essential and is just as important as the visualization technique itself. Some popular forms of DR include: Principal Component Analysis (Wold, Esbensen, and Geladi, 1987), Linear Discriminant Analysis (Fisher, 1936) and Latent Semantic Analysis (Deerwester et al., 1990). Because this chapter focuses on the visualization technique rather than the data processing steps taken beforehand, we will not further detail or compare DR techniques but relegate this task to other works such as Cunningham (2008) and Fodor (2002).

Over the last few decades a wide variety of data visualization techniques have been developed, many of which are focused on high-dimensional data since this type of data can be found in abundance in many real world settings. However, not nearly all of these visualizations are useful for finding topological features in high-dimensional data. For this reason, and because it would go beyond the scope of this thesis, this section will only cover a few of the most popular visualization techniques that are best suited for topological data analysis.

Star Coordinates (Kandogan, 2000) are one such technique. When visualizing data via Star Coordinates, the different dimensions are initially spread out equally in a radial pattern. The dimensions are normalized to all have the same length of 1. Each data point is then represented as a single point in this circular Cartesian plane. For each dimension, a normalized vector is calculated corresponding to the data point's relative position along the normalized dimension. All of these vectors are then summed up, which results in the point's final coordinates. The exact definition of the mapping of a high-dimensional point $D = (d_0, d_1, ..., d_n)$ to the two-dimensional point P in Cartesian Coordinates is as follows:

$$P = O + \sum_{i=1}^{n} \vec{a}_i \cdot \frac{d_i - min_i}{max_i - min_i} \tag{2.1}$$

where O is the origin of the 2D plot, $\vec{a}_i$ is the i-th axis vector of the plot, and min_i and max_i are the minimum and maximum values of the i-th coordinate in the whole data set.

Because the coordinates of the resulting point are simply a sum of the different dimensions opposite dimensions cancel each other out. Naturally, this is not always wanted, so users can change the angle of dimensions by dragging them. Additionally, users can stretch or compress the length of dimensions to change the represented importance of a dimension. Star Coordinates have two main applications: cluster analysis and decision-making. An example of such an analysis can be seen in Figure 2.1.

By changing the orientation and length of the dimensions, clusters can appear and disappear. Similar to the example in Figure 1.3, such clusters usually exist throughout a range of parameter scales, and depending on how large this range is, statements with varying degrees of certainty can be made about them. Star Coordinates also offer the benefit that each data point is only represented as a single point in a space of fixed area, which means that even for high-dimensional data, the visualization is still compact and exhibits little cluttering. Naturally, this comes with a loss of information, but if a user is mostly interested in finding clusters, this can be a desirable trade-off.

The Grand Tour (Asimov, 1985) is another visualization technique, which can allow users to find topological structures in high-dimensional data.

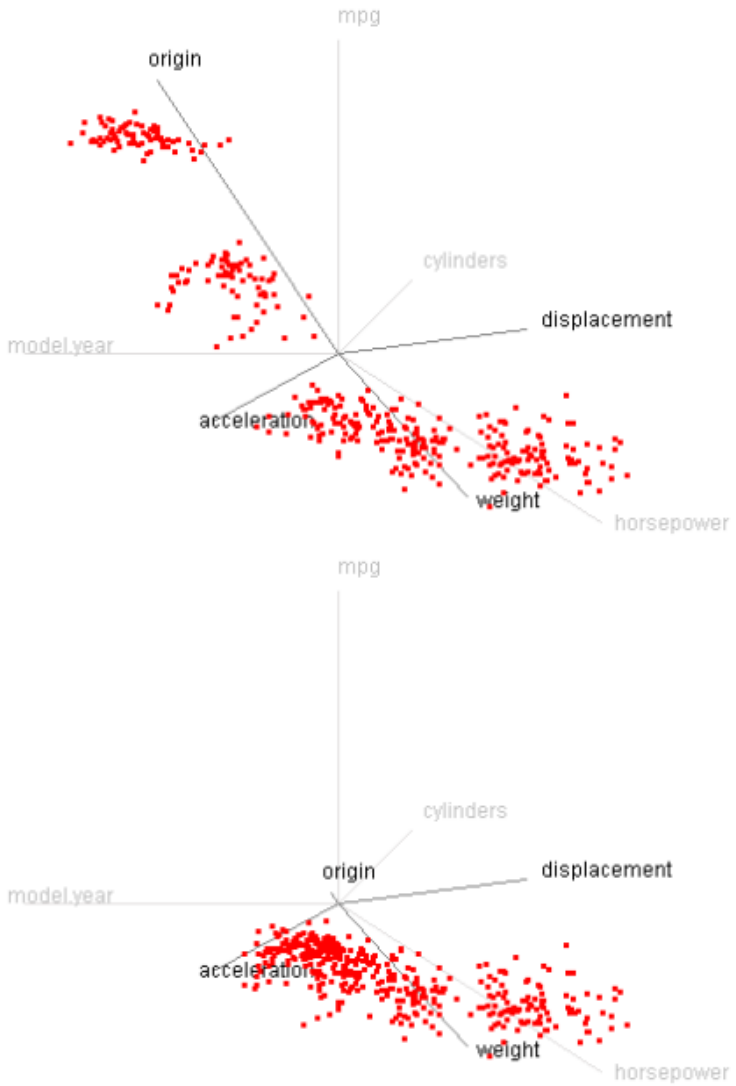

Figure 2.1: A Star Coordinate visualization of an 8 dimensional car data set. The difference between these two images is how much focus is placed on the car's origin. This image was taken from Kandogan (2000).

The Grand Tour is an animated visualization technique in which the high-dimensional points are projected orthogonally onto a 2D subspace from different angles. For a 3D structure, this would be the same as a camera moving around and object, each frame representing a different angle from which the data is viewed. This also means that each frame is basically a 2D scatter plot, but instead of projecting the data exactly along a set of dimensions, the data is projected from an intermediate angle. While animated visualizations are often hard to grasp when only viewing them once, an interactive animation, in which users would be able to play back, or slow down or speed up the animation could allow them to find clusters, ridges or other structures in higher dimensions. Since the Grand Tour is inherently animated, no example is included.

There has also been a lot of research concerning iconography, which, relating to data visualization, refers to the visual representation of high-dimensional data as simple shapes or forms called icons. While there are several types of iconography, and especially 'Chernoff faces' (Chernoff, 1973) have garnered a lot of attention, Star Glyphs seem to be best-suited for topological data analysis. Star Glyphs, just like Star Coordinates, spread out the different dimensions radially from a common center. However, when using Star Glyphs each data point is represented as a unique glyph, where the values for each dimension are plotted along an axes and connected via a polygon line. These glyphs are oftentimes is star shaped, hence the name.

While almost all types of iconography can be used to detect similarities or clusters between data points, Star Glyphs make it relatively easy to find correlations between different dimensions. These correlations can then be used to infer higher-dimensional structures. Additionally, they scale well to higher dimensions, which makes it not only possible to represent the data points as glyphs, but also the dimensions themselves, even for large data sets.

To compare different dimensions with the help of Star Glyphs, one can represent each dimension with a glyph instead of each data point. This means that the axes represent the corresponding values of different data points instead of dimensions. While similar methods are theoretically possible for many visualization techniques, since this is nothing more than a transposed version of the same data, most data sets have a larger number of data points

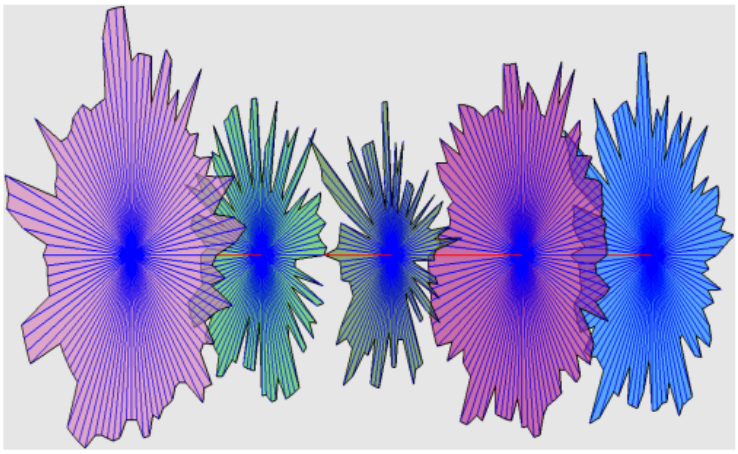

Figure 2.2: Five Star Glyphs arranged on their pivot axis. This image was taken from Fanea, Carpendale, and Isenberg (2005).

than dimensions. By transposing the data set, the result would then be a very high-dimensional space with only a few data points, which is usually undesirable and makes it harder to visualize. For example, consider the Chernoff faces.Visualizing a 5-dimensional data set with 100 data points is comparatively straightforward, however a 100-dimensional data set with only 5 data points is harder to visualize because there are not enough facial parameters to assign each dimension a different parameter. Usually each dimension is mapped to a single facial parameter such as length of nose or curvature of mouth, but there are only so many different features to map dimensions to. While this problem could be overcome by mapping several dimensions to a single facial feature, it would make it harder to differentiate between the resulting faces.

A question that still needs answering is how to order and present a set of such glyphs. Depending on the arrangement, groups of similar glyphs can be easier or harder to find. While there are many possible ways to do this, an interesting idea was proposed by Fanea, Carpendale, and Isenberg (2005), who put forward the idea of arranging Star Glyphs along an axis, similar to a Parallel Coordinate Plot. An example can be seen in Figure 2.2.

2.2 Scatter Plots

Scatter plots are one of the most popular ways to represent data in general. Scatter plots have been around for many decades and, while many other visualization techniques are hard to grasp for the uninitiated, they are also easy to understand and simultaneously extremely powerful. For a history of how the scatter plot was invented, see Friendly and Denis (2005). A scatter plot is usually defined by one to three orthogonal axes that each represent one dimension of the data. Data points are then represented as circles or dots in the Cartesian space, spanned by the dimensional axes. It is also possible to vary the points' shape, size and color to represent additional data attributes or dimensions. However, these additional visual differentiations can interfere with each other.

While scatter plots are usually used to represent continuous data, they can also be used to visualize categorical data, by equally spacing the different categories along an axis. When using continuous data, it can both be useful to normalize the scales of the axes or leave them at their original relative scales.

While the number of axes is not fixed, the two-dimensional scatter plot is by far the most popular choice. While three-dimensional scatter plots might appear more useful, because they can represent an additional dimension, they suffer from the same problem as all three-dimensional data representations, which is that most media is viewed on either 2D screens or sheets of paper. While three-dimensional scatter plots are also often used, but it is easier to get an understanding of the data's 3D distribution when users have the option to rotate the scatter plot. Since this is not always possible, especially in print or with static images, 2D scatter plots are considered to be the default.

One aspect to consider about 2D scatter plots of high dimensional data is that they are a projection of the data along the other axes. For example, in a 3D data set, any 2D scatter plot consisting of two of the three axes simply represents a different angle from which the 3D points are viewed. While this axis-aligned point of view is oftentimes interesting, structures may not necessarily appear more clearly when viewed like this. For example, consider a set of three-dimensional points that lie on the surface of a cylinder.

When the cylinder's flat sides are aligned with one of the axes, the patterns that would appear by looking at a two-dimensional scatter plot would be a clear circle and rectangles, depending on which two dimensions are chosen as the base for the scatter plot. By looking at all possible 2D scatter plots, one could then ascertain the 3D shape of the data. However, if the cylinder were to be rotated in 3D space, it can be almost impossible to see any correlation whatsoever. This means that scatter plots can have varying degrees of usefulness when trying to ascertain the topological structure of a high dimensional data set. The idea that low-dimensional scatter plots are nothing more than projections of the original data along particular axes is important to keep in mind when talking about more complex visualization techniques involving scatter plots.

There is also the one-dimensional scatter plot, but they generally only allow us to get a broad overview of how the data is distributed in this one dimension. Linked together, however, this can also be a useful representation of high-dimensional data as will be discussed in Section 2.3.

At first glance, scatter plots might not appear to be particularly well-suited for topological data analysis, since the number of dimensions of a scatter plot is inherently limited to three. This, however, is not the case since a combination of 2D scatter plots can be used to give the user different 2D perspectives into the high dimensional data space.

2.2.1 Scatter Plot Matrices

The most natural approach to representing high-dimensional data via scatter plots is to create a 2D scatter plot for every combination of dimensions. By arranging these scatter plots in a $2x2$ matrix one ends up with a Scatter Plot Matrix. An example can be seen in Figure 2.3. Scatter Plot Matrices are a popular tool because they are easy to understand and oftentimes correlations can already be found between just 2 sub-dimensions. However, they also have an obvious limitation in that a Scatter Plot Matrix grows quadratically with the number of dimensions. This means that for data sets with more than a few handful of dimensions the matrix becomes so large that it takes users a long time to sift through the numerous 2D scatter

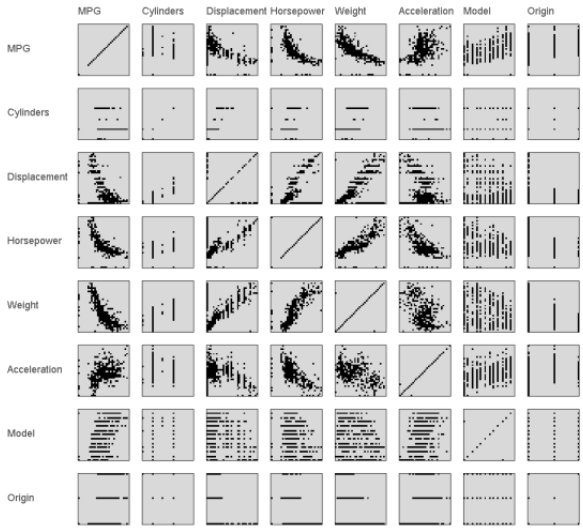

Figure 2.3: A Scatter Plot Matrix of a 7-dimensional car data set. This image was taken from Elmqvist, Dragicevic, and Fekete (2008).

plots. Additionally, only a few scatter plots will contain relevant information to a user. Scatter Plot Matrices are a popular research subject and many approaches have been proposed through which to alleviate this problem.

For example, Shao, N. Silva, et al. (2017) propose a recommendation-based approach, which shows users interesting and previously unseen scatter plots based on the user's eye movements. They allow users to freely explore the Scatter Plot Matrix while simultaneously recording the eye movements of the user. The duration for which a scatter plot is looked at is then used as the basis for the recommender system, which uses a k-nearest-neighbor (KNN) search to identify similar scatter plots and then recommend scatter plots which are different to the ones previously explored. The idea of adding recommender systems to the exploration process has been used before and can be a powerful aid in the exploration process.

Another paper by the same author (Shao, Behrisch, et al., 2014) explored the idea of using manual sketches as a search modality to find scatter plots with similar patterns. They gave users the ability to sketch a visual pattern they were looking for and returned a range of 2D scatter plots which best

matched the sketch. This made it relatively easy to find structures of a certain kind. They also added a shadow drawing mechanism (Lee, Zitnick, and Cohen, 2011), which overlays similar results already during the drawing process and allows users to get an overview of similar results or even adapt their query with immediate feedback.

Elmqvist, Dragicevic, and Fekete (2008) proposed a set of tools that allows users to quite literally traverse a Scatter Plot Matrix by animating the change between two scatter plots where one axis stays the same. In their prototype, the Scatter Plot Matrix was used as both an overview of the entire exploration space and a navigation tool. They allowed users to look at a single 2D scatter plot, and via a variety of movement options travel through the Scatter Plot Matrix. While still only using two- and three-dimensional scatter plots, this method is intuitive and allows for a target-oriented exploration of a Scatter Plot Matrix.

2.3 Parallel Coordinate Plots

Parallel Coordinates or Parallel Coordinate Plots (PCPs) were first invented by Hewes and Gannett (1883) and then later independently reinvented by Inselberg and Dimsdale (1990) and are commonly used to display multivariate data. When using parallel coordinates, each dimension is mapped onto onto one of N (usually vertical) lines with equal lengths. These lines, or *axes*, are then placed next to each other and each data point is represented as a polygon line that intersects each axis at the respective coordinate value of this corresponding dimension. An example can be seen in Figure 2.4.

PCPs derive much of their popularity from the fact that their size scales linearly with the number of dimensions, since each dimension is represented by a single coordinate. This is a clear benefit over Scatter Plot Matrices which scale quadratically. Additionally, many common correlations between two dimensions can be discerned by looking at the line intersections between them. Some examples can be seen in Figure 2.5.

It is worthy of mention that many of these correlations, especially the exponential e^x correlation, can be hard to see when too many lines are

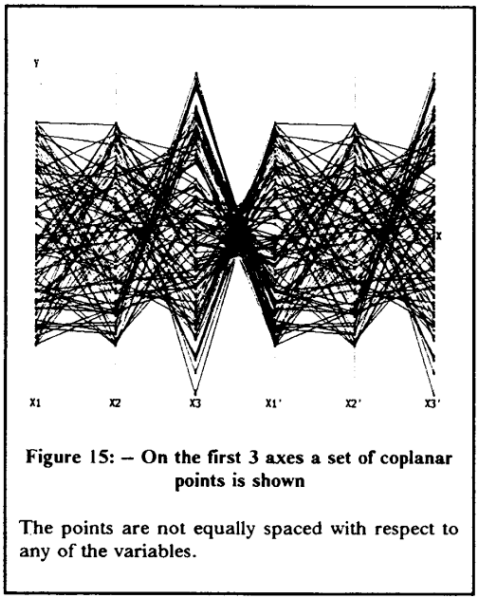

Figure 2.4: A Parallel Coordinate Plot with six dimensions. This image was taken from Inselberg and Dimsdale (1990).

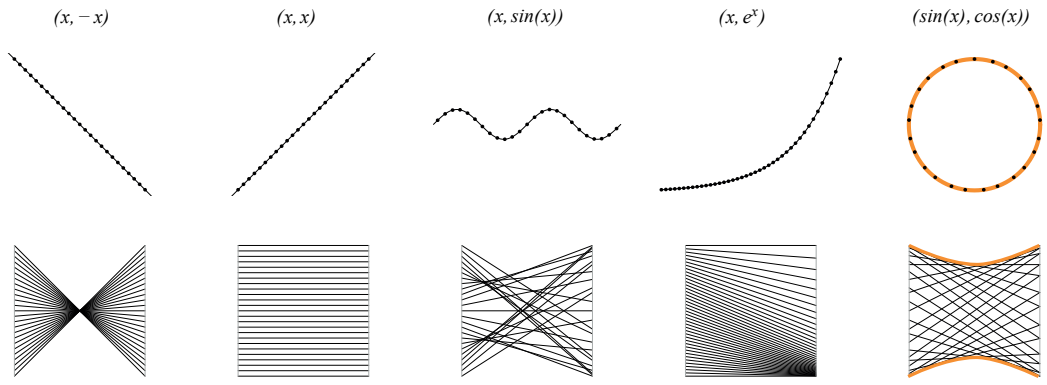

Figure 2.5: Common patterns in two dimensions visualized via a scatter plot (top) or a PCP (bottom). This image was taken from Heinrich and Weiskopf (2013).

shown simultaneously, because when drawn fully opaque, the differences in line densities can be difficult to discern. These patterns can become clearer by lowering the line opacity, but finding an opacity value that fits for all types of data sets is a difficult task. An easier solution would be to allow users to change the opacity on their own, but this is not always an optimal solution either.

While these patterns help users understand the topology of the data, they are also just combinations of two sub-dimensions. This means that the order of the dimensions is of high relevance since interesting patterns can only be found in adjacent dimensions. Naturally, it would also be possible to visualize all 2D combinations, similar to a Scatter Plot Matrix, as suggested by Heinrich, Stasko, and Weiskopf (2012), which results in a visualization they aptly named a parallel coordinates matrix. However, this approach forfeits the main benefit of PCPs over scatter plots, which is their linear scaling with the number of dimensions. Other researchers have also focused on how to arrange the dimensions in PCPs (Tilouche, Partovi Nia, and Bassetto, 2021) but this problem is still a topic of research.

While the order of dimensions in PCPs is important, reordering dimensions only allows users to see more relevant patterns in 2D subspaces. However, Li, Martens, and Van Wijk (2010) even showed that generally scatter plots are more effective at finding correlations between two dimensions than PCPs, but this does not mean that there is no benefit to also visualizing these correlations in PCPs. Sometimes patterns become clearer and can be more easily found when they are visualized in several different ways. Furthermore, PCPs visualize more than just a subset of all 2D combinations, they show all dimensions simultaneously in a single window, which makes them a powerful tool for finding structures that might not be visible in any 2D subspace. Especially clusters can be easily detected because they appear as tight bundles of polylines across all dimensions. Naturally these clusters could also become more or less visible when rearranging the order of dimensions, but generally they can be detected either way. Clusters also become more apparent when using different line opacities or line colors. However, besides clusters, it seems difficult to find structures such as ridges or low dimensional manifolds on which the data points lie solely by looking at the different polygon lines.

2.4 Hybrid Techniques

Both scatter plots and PCPs have different strengths and weaknesses and there have been efforts to combine both techniques into one visualization. These combined techniques aim at leveraging the strengths of scatter plots and PCPs while minimizing redundancy and clutter.

The first technique that combines both techniques directly was proposed by Wegenkittl, Löffelmann, and Gröller (1997). They replaced the normal one-dimensional lines in a PCP plot with two-dimensional planes and therefore called their technique "three-dimensional parallel coordinates". In their paper they aligned the planes in such a way that the sides of the planes were facing each other. Figure 2.6 shows an example. This also reduces both the number of lines that have to be shown on screen simultaneously and the number of possible arrangements within the PCP. This last point is especially noteworthy, since as discussed, the arrangement of PCPs is of high relevance. However, Johansson Westberg, Forsell, and Cooper (2013) showed, that users prefer the normal PCP plot over its three-dimensional counterpart for detecting subspace correlations. While their sample size was too small to make general assumptions with certainty, it seems reasonable as there are many more patterns between the different dimensions that would have to be recognized than in a normal PCP. While this version allows the connecting lines to occupy the most space and thus encode the most information, three-dimensional parallel coordinates can also be created by rotating the planes by 90°such that all of them face in the same direction. This allows for a representation with a static camera and the 2D patterns within the planes are all visible simultaneously. Especially the fact that all 2D scatter plots are visible at the same time seems to have a lot of merit, since scatter plots show some correlations more clearly than parallel coordinates. However, in this approach the connecting polygon lines all have to be projected onto a plane, which in turn leads to a loss of information.

Similar to how three-dimensional parallel coordinates expand the normal PCP by increasing the number of dimensions represented by each layer, three-dimensional parallel coordinates can also be extended to four-dimensional parallel coordinates by replacing the 2D scatter plots by 3D

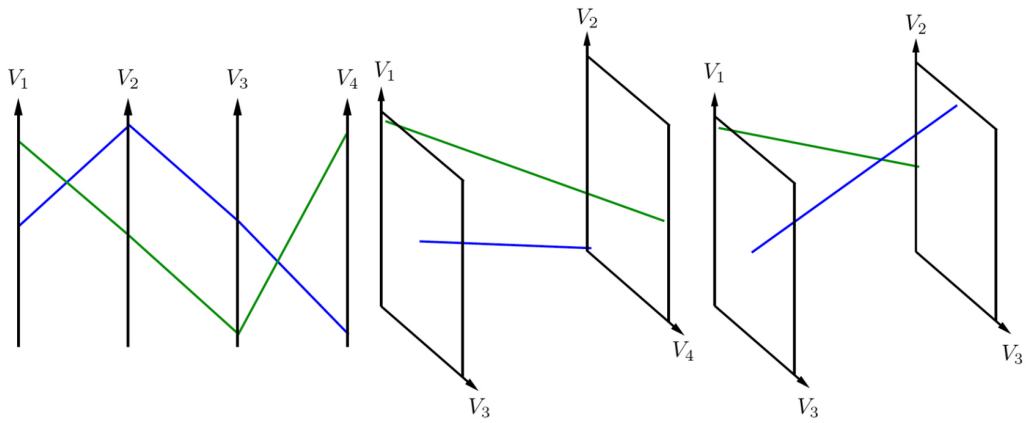

Figure 2.6: A simple example of how a normal PCP can be converted to a three-dimensional PCP. Depending on which dimensions are used to create 2D scatter plots, the resulting line patterns change. This image was taken fromJohansson Westberg, Forsell, and Cooper (2013).

ones. Naturally, this exacerbates the problems previously mentioned, but can sometimes reveal structures in 3D that would not be visible otherwise.

Yuan et al. (2009) introduced another idea, which bears similarity to a three-dimensional scatter plot. They proposed to insert scatter points into the space between two parallel coordinate axes, P_a and P_b. With their default setting, the coordinates of the scatter plot points within two such axes would be the value of the data point on P_a as y coordinate and the value of the data point on P_b as the x value. While a normal PCP would produce a straight line for each data point between P_a and P_b, they proposed to use splines instead, such that three conditions are fulfilled:

- Each spline intersects the parallel coordinate axes at their usual location.
- Each spline intersects the newly inserted scatter point.
- Each spline connects smoothly with the following spline connecting P_b and P_c

The use of these splines leads to fewer overlapping lines and, according to the authors, improves data comprehensibility, especially compared to systems with multiple views. Furthermore, there are many possible ways to

distribute the scatter points. It is also possible to represent several dimensions simultaneously with these scatter points, for example by projecting them from a high dimensional space to a 2D space via multidimensional scaling (Wong and Bergeron, 1997).

As mentioned before, the patterns created by PCPs between two dimensions are helpful in understanding correlations in the data, but even more interesting is the way in which the polygon lines flow through all of the dimensions. Jäckle et al. (2017) investigated this topic by exploring how patterns in subspaces can be found by observing the change in subspace patterns. They first find interesting subspaces and project them into a 2D subspace. These 2D subspaces are then aligned into a three-dimensional PCP. They grouped these subspaces based on similarity and finally highlight the change of a user-selected pattern. The selected data points are connected across all subspaces. The connecting polygon lines can be analyzed to infer the structure of high-dimensional data sets.

2.5 Unsolved Problems

Almost all of the aforementioned visualization techniques can be used to detect clusters in high-dimensional data sets. While the approaches vary drastically, it seems that clusters are the topological structure that is easiest to identify visually. Other structures such as holes or ridges seem to be much harder to identify.

For example, consider a data set in which all the data points lie on the surface of a 4D hypercube. The fact that this hypercube is completely empty on the inside would be difficult to detect with any of these visualization techniques. Similarly, the fact that all data points lie on a 2D manifold, embedded in the four-dimensional space would also be hard to detect.

Now imagine that the data points form some sort of interesting pattern on the surface of this hypercube. Since a cube is a relatively simple structure, such a pattern could still be detected in a subspace, for example, via a simple Scatter Plot Matrix. However, if we were to rotate the cube such that its surfaces are no longer axis-aligned and we were to stretch and compress

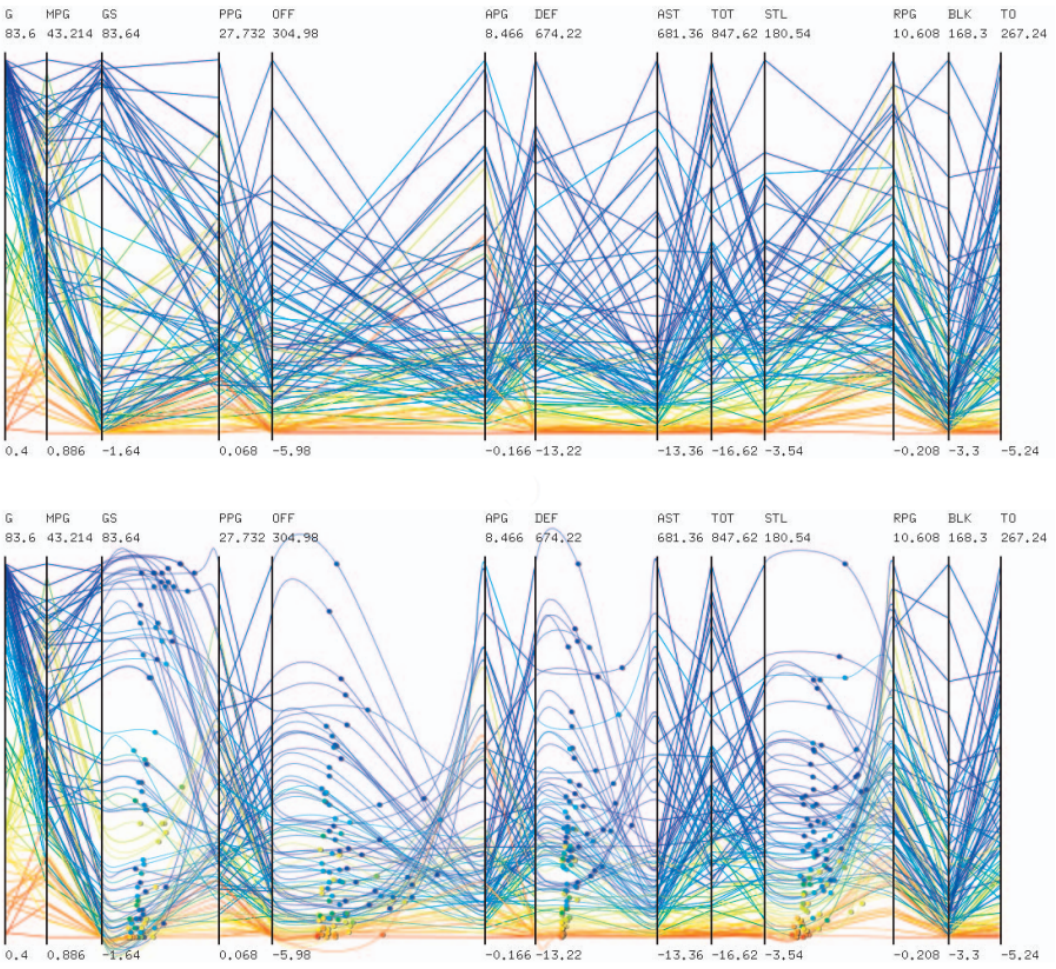

Figure 2.7: The top image shows a normal PCP. The bottom image shows a PCP with scatter points between some dimensions. This image was taken from Yuan et al. (2009).

parts of its surface, such a pattern would become increasingly difficult to find.

This example illustrates that manifold estimation or even just detection is still a largely unsolved problem in visual topological data analysis. The aim of this thesis is to address these problems by introducing a novel interactive visualization concept, which should allow users to simplify the data by reducing the number of dimensions and reveal structural properties of the data.

3 Concept and Implementation

This chapter covers the functions and ideas behind the interactive TDA tool that was developed as part of this master thesis. This includes a description of the initial ideas and concepts and how these ideas changed over the course of the development, as well as a technical description of the frameworks and languages used.

3.1 Core Ideas

Because it was desirable for the chosen visualization technique to not be limited by the number of dimensions of the data set, it was decided to choose a Parallel Coordinate Plot as the foundation of the prototype. One of the most obvious problems with Parallel Coordinate Plots is the question of how to arrange the dimensions. As discussed in Chapter 2, there are many approaches to tackling this issue. Due to the fact that this work focuses on interactive data exploration, the chosen method of arranging the parallel coordinate axes was to use the order provided in the data set as a starting point, and then allow users to rearrange them interactively as desired. A screenshot of the PCP that is shown immediately after loading a data set can be seen in Figure 3.1. This decision gives users more freedom to explore the data on their own, but comes with the obvious drawback of having to sift through many arrangements.

In addition to the base PCP, the initial design concept consisted of 3 main pillars:

- merging and unmerging of dimensions
- unrolling of embedded dimensions
- tracking and visualizing data provenance

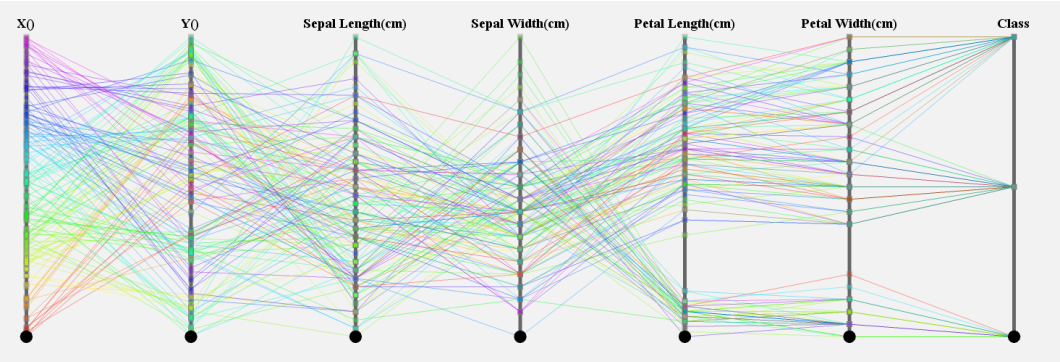

Figure 3.1: The standard PCP view, that is shown upon loading a data set. The order of the axes is the same order as in the imported CSV file.

These functionalities are the main ways by which users can interact with and explore their data.

3.1.1 Merging and Unmerging of Dimensions

Merging different dimensions within the PCP allows users to create two- and three-dimensional scatter plots. Because parallel coordinate axes can already be seen as one-dimensional scatter plots, this does not change the way the data is projected, but simply changes the relation between two dimensions, from being parallel to being perpendicular to each other. Note that the term *scatter plot* refers to the general concept of a scatter plot including points on a one-dimensional axis of a PCP, as well as the more commonly known two and three-dimensional scatter plots.

Merging scatter plots into higher-dimensional scatter plots allows users to find correlations and clusters within these dimensions more easily. A PCP with one-, two- and three-dimensional scatter plots can be seen in **??**. While any pattern that can be found in a two-dimensional scatter plot could also be found by looking at the same two dimensions when they are adjacent in the PCP, they are oftentimes harder to spot. Furthermore, it gives users the option to choose their preferred way of looking at the data, and three-dimensional scatter plots can reveal structures that cannot be seen by just looking at a PCP. Merging dimensions should be reversible, and it

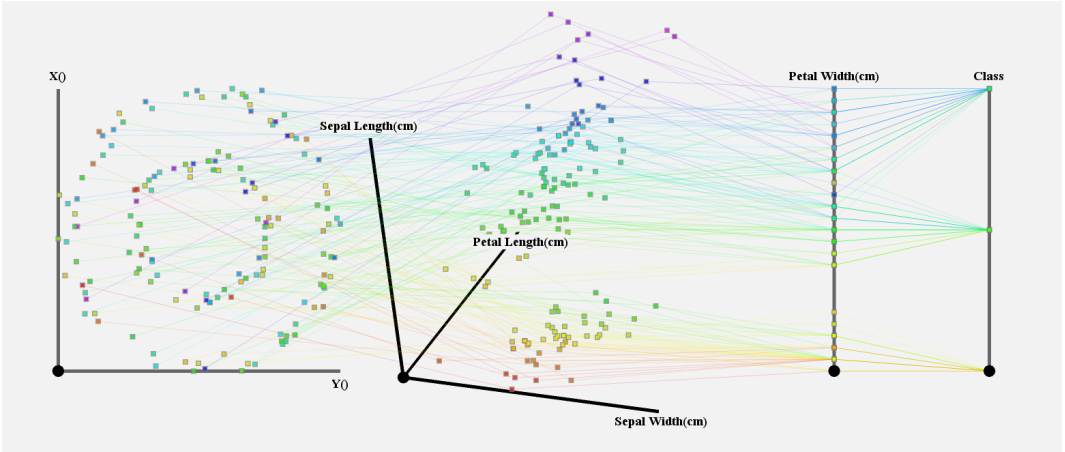

Figure 3.2: A PCP layer with one-, two- and three-dimensional scatter plots. The three-dimensional scatter plot has been rotated to better showcase its three-dimensional nature.

should be possible to restore the original PCP by unmerging scatter plots back to their original one-dimensional form.

Additionally, when looking at three-dimensional data points through a two-dimensional screen, the user needs to have the ability to rotate either the data points, or the camera to get a better feeling for the three-dimensional position. Otherwise, most users struggle to accurately identify the depth of objects. For this prototype, it was decided to let users rotate three-dimensional scatter plots by dragging them with the mouse after entering rotation mode (see Figure 3.2, second scatter plot from the left).

3.1.2 Unrolling of Embedded Dimensions

After merging two or three dimensions, users are able to project the data points into a space with one less dimension. This can be done in three different ways. The simplest way to achieve this is to fit a polynomial function onto a 2D scatter plot via linear regression (Montgomery, Peck, and Vining, 2021). After constructing such a polynomial function, all data points are projected to the nearest point on the polynomial and reparameterized along it to create a new one-dimensional scatter plot. An example of a polynomial

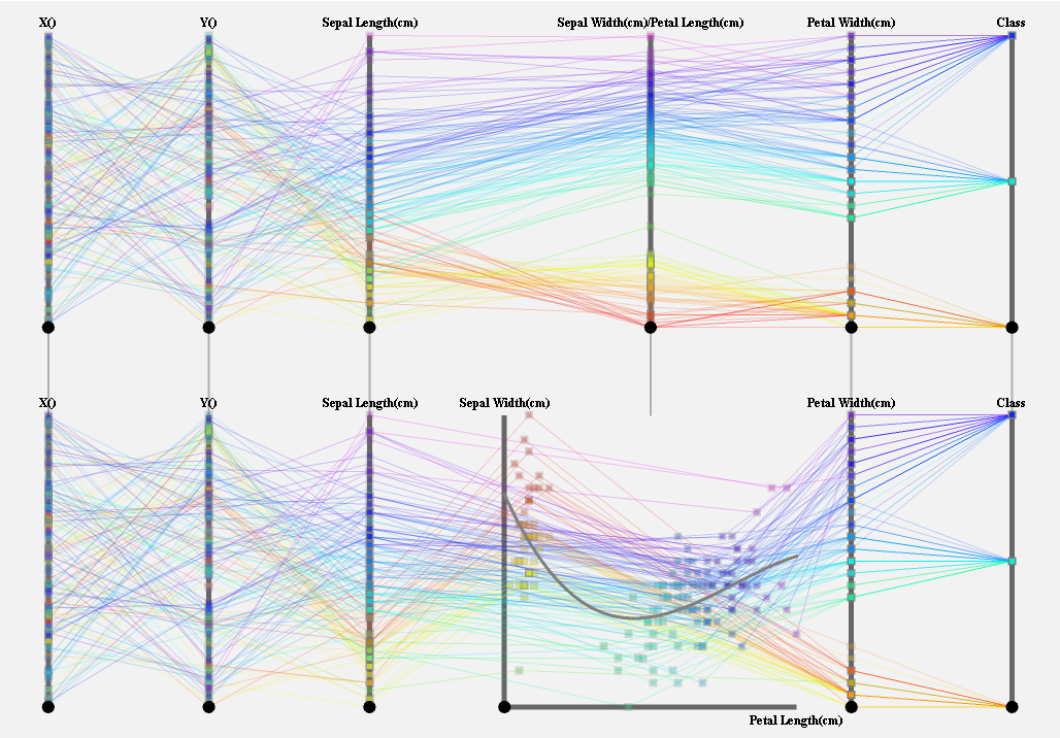

Figure 3.3: A polynomial function has been used to calculate the least-squares error in the 2D scatter plot. All data points are then projected onto this polynomial, which results in the new 1D scatter plot in the second PCP layer.

line fit can be seen in Figure 3.3. Alternatively, users can interactively draw a curve into a 2D scatter plot, in which case points are similarly projected onto the curve to create a new, reduced one-dimensional axis. Lastly, users can also draw a curve on a 3D scatter plot, in which case the line is extruded along the view-space z-axis to form a developable surface. In this case, the resulting scatter plot has two dimensions. Drawing such curves by hand is especially helpful for structures such as circles, which cannot be described by a polynomial function.

Projecting and unrolling a new axis this way leads to the creation of a new PCP within the user interface, which is a duplicate of the original PCP, except that the scatter plot with the fitted function or drawing is replaced by a scatter plot of lower dimension. This allows users to iteratively reduce

the number of dimensions by merging and unrolling embedded dimensions. Of course, this leads to a greater loss of information when the chosen dimensions are correlated less strongly. Therefore, it is only meaningful to unroll such embedded dimensions when a relatively clear correlation can be found.

3.1.3 Tracking Data Provenance

The third pillar of this prototype is the visualization of data provenance. Users are allowed to modify and simplify the underlying data in various ways, but the topological structures that can be found with these processes have to be tracked back to the data's original form in order to draw conclusions. This was done in three ways.

Firstly, some actions, like unrolling an embedded dimension, lead to the creation of a copy of the previous PCP, as explained above. This means that users can scroll down to see which embedded dimensions have been unrolled, since the drawing or fitted function is still visible in the lower layer.

Secondly, scatter plots that represent the same dimensions reflect the same horizontal position as their counterpart in higher or lower layers, even when moved. Although each axis of a scatter plot possesses a label, it can be hard to find a counterpart to a scatter plot when the data set consists of dozens or hundreds of dimensions. By keeping them vertically aligned, it becomes easier to focus on the differences between the PCP layers, like the unrolled dimensions.

Finally, the relation between plots of different layers is visualized via highlights and connecting lines. A child plot is a plot that either represents the same dimension or is a product of unrolling an embedded dimension of a plot in a lower layer. For example, when unrolling an embedded dimension of a two-dimensional scatter plot, the resulting one-dimensional scatter plot would be considered the child of the original 2D plot. Similarly, if a user were to merge two one-dimensional plots in a higher PCP layer, the two one-dimensional plots in the layer below would be considered its parents and highlighted correspondingly. When hovering over a scatter plot, the

axes of its children and parent plots are highlighted in different colors, red for the parents and blue for the children. Additionally, they are connected via a line from the top of the parent plot to the origin of the child plot. An example of how the state of the application may look like after several actions have been performed can be seen in Figure 3.4. This figure shows a more complex state which was achieved after unrolling an embedded dimension and then merging the resulting 1D axis in the second layer. This 2D scatter plot was then unrolled again and the 1D axis in the third layer was merged with two more one-dimensional axes to form a 3D scatter plot. Additionally, two 1D scatter plots were merged in the second layer, revealing circular structures.

3.2 Supplementary functions

During the development phase of the prototype, it became clear that some more functions were needed make it more accessible and practical for the user.

A problem that occurred during the early stages of development was that it was very hard to see correlations in two and three-dimensional scatter plots due to the oftentimes large number of lines that would pass through the plot. While it would be possible to simply set the line opacity lower, or the point opacity higher, this would have varying results based on the density of data points. Therefore, density sliders for both opacities were added, to allow the users to change opacities during runtime.

Additionally, it became clear that when loading a data set with many dimensions, it would take users a long time to find correlations in two or three dimensions when they would have to go through all combinations by hand. For this reason, a recommender system was added that creates all possible 2D scatter plots from a chosen 1D scatter plot and fits a polynomial function via linear regression. For a pair of dimensions x and y, both the scatter plot $Plot(x, y)$ and the transposed version $Plot(y, x)$ are created and their respective least-squares errors E are calculated as defined in Equation 3.1.

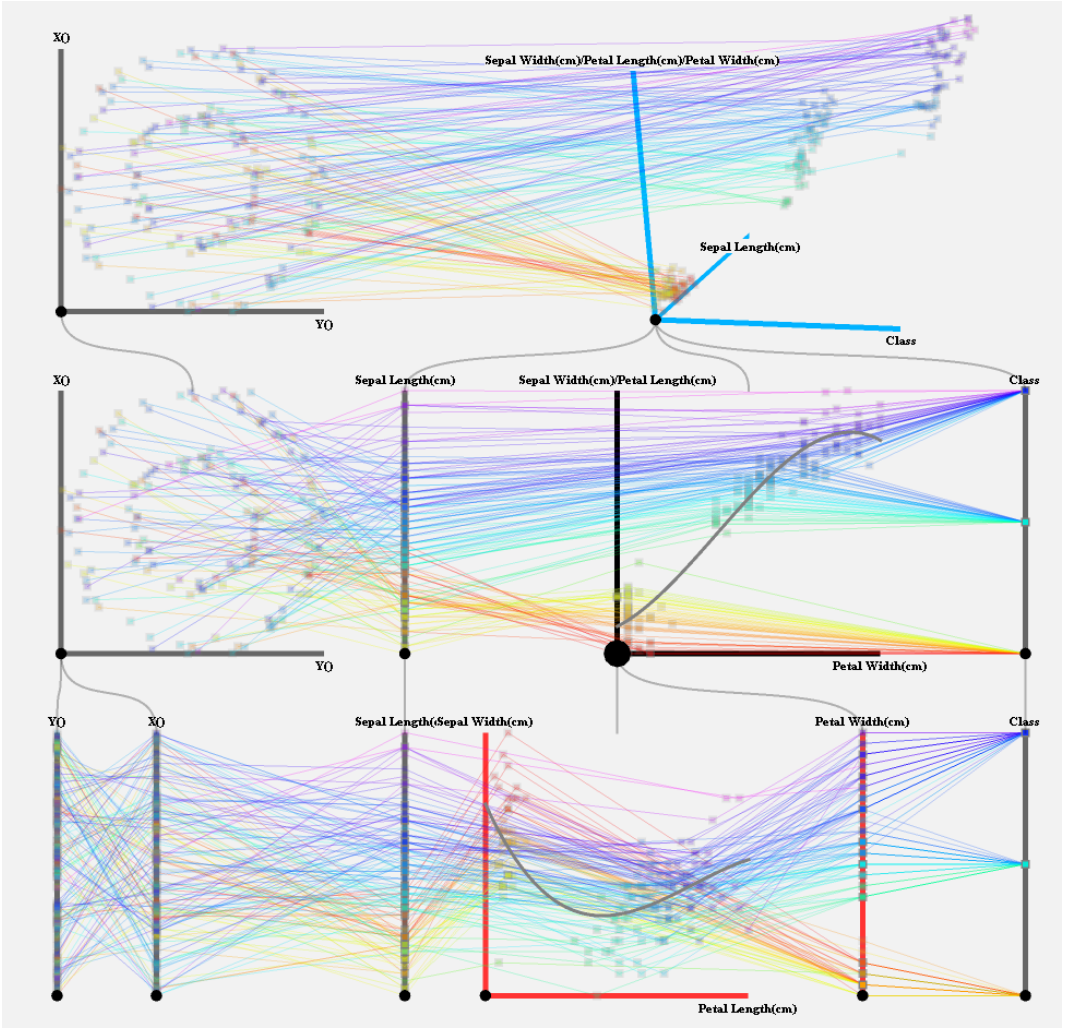

Figure 3.4: Three PCP layers, created by unrolling embedded dimensions two consecutive times. When the user hovers over the 2D scatter plot to the right in the middle layer, its parents that were merged to create it are highlighted in red and the 3D scatter plot that contains the unrolled dimension is highlighted in blue. All parent-child relations are visualized via the connecting lines.

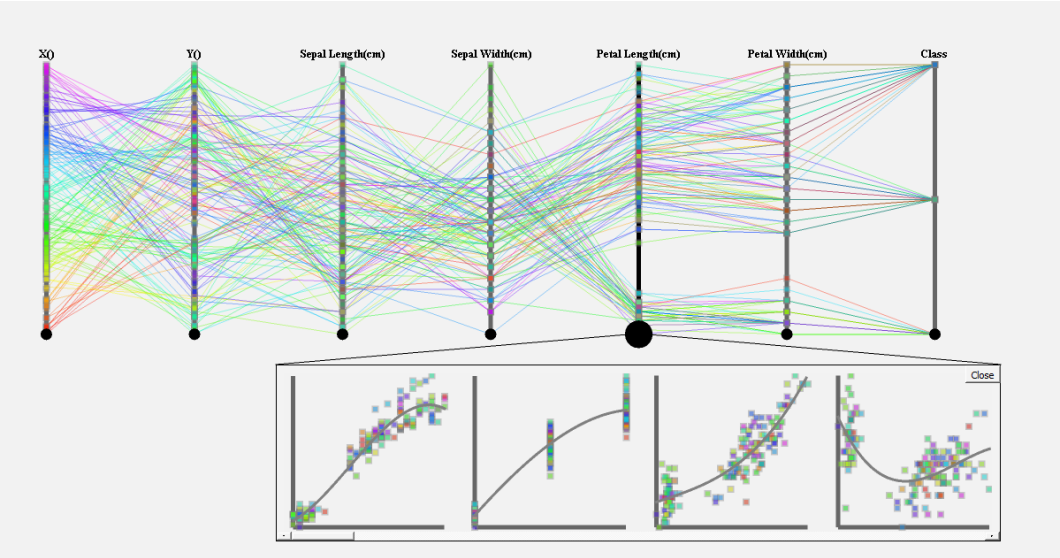

Figure 3.5: The user's mouse is hovering over a 1D scatter plot. The recommendation pop-up initially shows the four 2D scatter plots with the best polynomial function fit. More possibilities can be viewed when the slider is moved to the right.

$$E = \sum_{i=0}^{n} (f(x_i) - y_i)^2, \tag{3.1}$$

where x_i and y_i are the X- and Y-Axis values of a point $P_i = (x_i, y_i)$ and f is the polynomial function. Since the same structures can be seen in both plots, the one with the higher error E is discarded. These 2D scatter plots are then sorted in descending order by their least-squares error E and shown below the chosen 1D scatter plot.

Small previews of these possible merges are presented to the user in a pop-up, when hovering over a 1D scatter plot. An example of how this might look can be seen in Figure 3.5. The same recommendation, with a few more details, can also be viewed in a separate window by right-clicking a 1D scatter plot. The user can then click on a 2D scatter plot to automatically merge them. They could also choose to additionally use the polynomial to unroll an embedded dimension.

Another common feature that was added was a color gradient for the polylines of the PCP. This color gradient is a simple rainbow color spectrum based on the Y-position of a data point in the selected scatter plot. When starting the application, the color is based on the leftmost 1D scatter plot. Afterwards, the user can choose which scatter plot to use to define the color. If the user selects a 2D or 3D scatter plot the Y-Axis is chosen to define the polyline colors. Many PCP implementations only use a single color because it more clearly shows line density and is supportive of people with color deficiencies. However, a colored PCP can make it easier to spot clusters and follow lines through several dimensions. This trade-off was considered, and at least for the current prototype, it seemed that a colored PCP was beneficial.

Additionally, it became clear that selecting a subset of data would open up interesting exploration options. Selecting a subset of the current data is done via lasso selection, by encircling the data the user wants to take a closer look at. An example can be seen in Figure 3.6. This can be done in one, two or three-dimensional scatter plots. Just like when unrolling an embedded dimension, this action creates a new PCP layer which contains copies of the same scatter plots as the last layer, but this time only the selected data points are shown. This can sometimes mean that the data points are spread out more among certain dimensions to still make use of the entire space of a scatter plot. The lasso sketch is then permanently displayed to show which selection was made.

Lastly, during the testing state we observed that another tool was needed to unroll embedded dimensions. Projecting points onto a hand-drawn hyperplane (line or 2D plane) was the most powerful tool for users, but sometimes not very intuitive. Additionally, it is difficult to draw perfectly straight lines. This is a shortcoming, since many correlations found in real-world data are linear. Therefore, another option was added to project the points of a 3D plot onto a plane that is parallel to the screen. This can be helpful when structures are found by rotating the 3D plot and the user would like to use the image they have in front of them to flatten the 3D plot to a 2D plot in which the same structures are visible. An example can be seen in Figure 4.3, where the second layer contains a 3D scatter plot with the extruded dragon shape. Without this new function, it would have been difficult to accurately project this structure onto a plane.

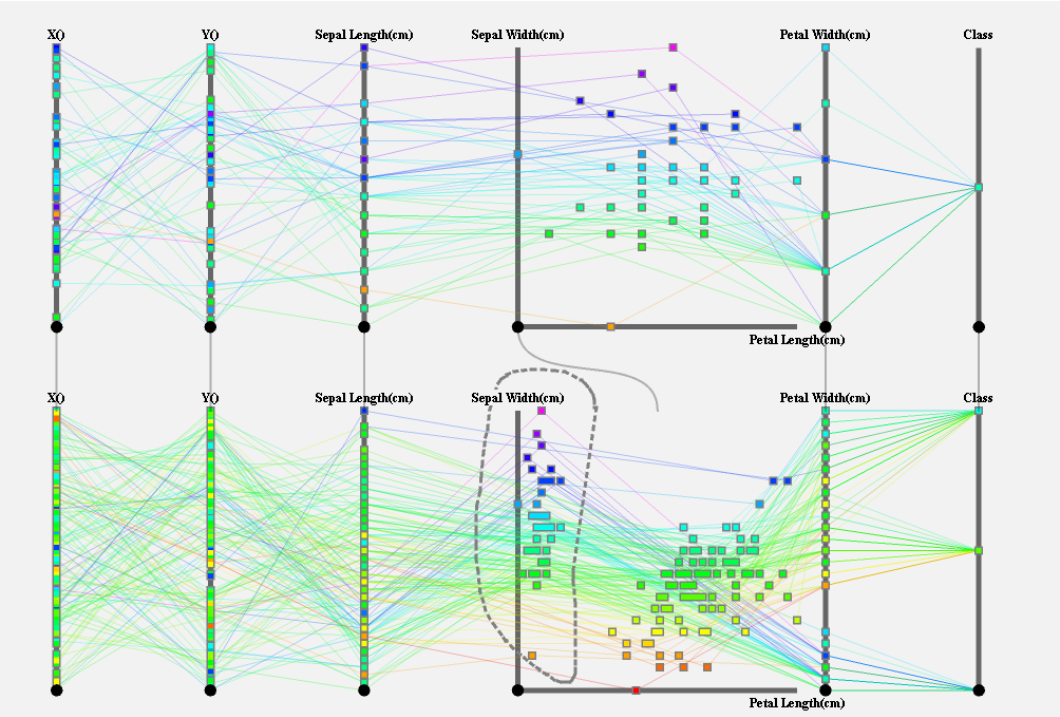

Figure 3.6: After encircling a subset of the data, a new PCP layer is created in which only the selected points are displayed. In order to use the entire space available, the data points in the upper layer are spread out.

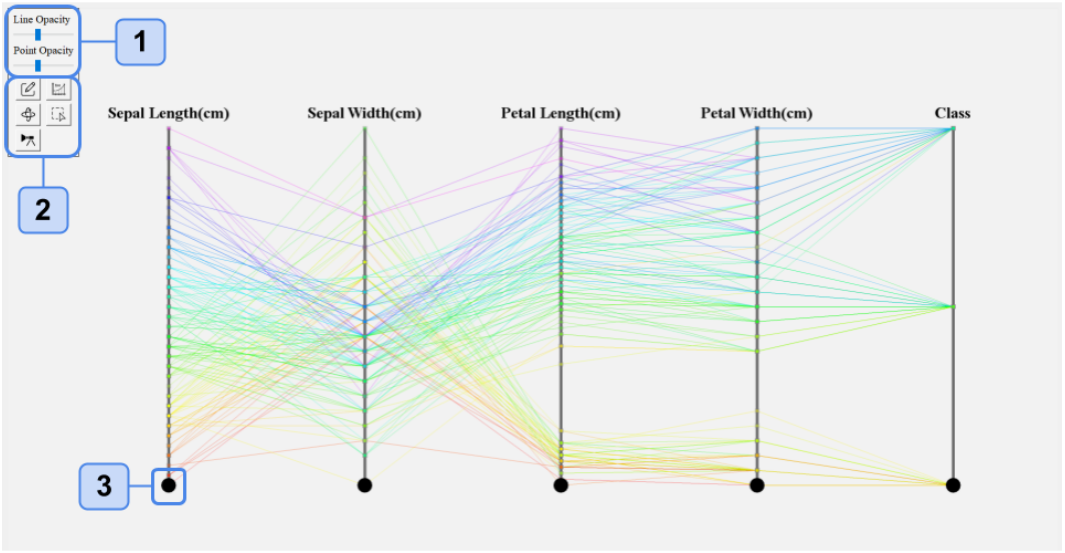

Figure 3.7: The user interface of the current prototype. The visualized data set is the popular 'iris' machine learning data set (Anderson, 1935). 1. Opacity Sliders for the Points (especially in 2D and 3D) and lines. 2. Different Tools to perform actions. From top left to bottom right: Curve Sketch Tool, Function Fit Tool, Rotate Tool, Selection Tool, 3D Projection Tool. 3. Interaction points at the bottom of each scatter plot. Actions such as dragging, merging and unmerging can be performed by clicking this point.

3.3 Usability

Currently, the user interface only contains the most basic features, with a small widget at the top left to select which action should be taken and to change the point and line opacities. A screenshot is shown in Figure 3.7. This allows users to focus on the main visualization. An effort was made to integrate all visualization into one window, instead of having many different views, but clearly this has some limitations. For example, it might be useful to also have a scatter plot matrix in a different window, to explore many 2D scatter plots with ease.

To keep the user interface as simple as possible, we decided to have users mainly interact via the black circle at the origin of each scatter plot axis. This

means that different actions can be performed by using different actions with this button. A right click undoes the last action. Therefore, if the scatter plot contains a selection or curve drawing, it is deleted. Otherwise, a single axis is removed from the scatter plot, reversing the last merge. If the rotation mode is active, right-clicking the origin will reset a 3D scatter plot's rotation. Double-clicking the origin will use the Y-axis of the selected plot to define the colors of the polylines and clicking and dragging the origin moves the axis.

Another interesting design decisions was made when it comes to creating or deleting new layers. Currently, a layer is only created when selecting a subset of the data for future analysis, or when an embedded dimension is unrolled. However, a layer is only deleted after all the selections and unrolled dimensions in the layer below are undone and all the newly merged scatter plots are unmerged. This was decided to keep the user's analysis safe from accidental deletion, as large amounts of work could otherwise be deleted by an accidental right-click. For the same reason, it is not possible to undo actions in lower layers, when the resulting scatter plots have been used in a higher layer. For example, in the scenario shown in Figure 4.3, it would not be possible to unmerge the 2D scatter plot in the lowest layer because the layers above depend on this merge. This design decision leads to the fact that layers have to be undone one by one, which can sometimes take several clicks.

3.4 Languages and Frameworks

Since topological data analysis can oftentimes involve large datasets with many data points, each having many attributes, the decision was made to use a language which allows for high-performance computations to provide interactive frame rates. Therefore, the C++ programming language was chosen.

The fact that the C++ standard libraries are not well suited for front end development, meant that a framework for user interaction was required. For this purpose, Qt was chosen as it is one of the most popular GUI frameworks for C++.

While Qt contains a data visualization framework, the requirements for this project were different enough such that it was necessary to use a graphics API. Qt does offer relatively low-level drawing functionality in their QPainter class, but after some tests it became apparent that with this approach only a few thousand data points could be rendered while keeping the application running smoothly and responsively. This led to the decision to also use the OpenGL framework to achieve an even better performance and to allow for larger data sets to be visualized. To use OpenGL in conjunction with Qt, Qt's class QOpenGLWidget was used.

The combination of these highly performant frameworks and libraries allows the prototype to visualize quite large data sets. The largest ones tested on a standard PC, with an NVIDIA GEFORCE GTX 1660 Ti GPU and an Intel Core i7-8700k CPU and 64 GB of RAM contained just over 32,000 data points and 15 dimensions. The prototype was also tested with a data set containing 2,000 data points and 65 dimensions, still running smoothly.

4 Results

This chapter illustrates the capabilities of the current version of our interactive topological data analysis tool.

4.1 Use Cases

The prototype's capabilities have been tested using different data sets with varying numbers of data points and dimensions. To illustrate some specific analysis scenarios, synthetic data sets were created.

4.1.1 Structures in Two Dimensions

Because the recommended 2D plots are sorted by their least-squares error, after fitting a polynomial function of degree 3, linear or polynomial correlations can be found within a few seconds. Patterns that cannot be expressed as polynomial functions are harder to find via this automatic ranking of the recommendation system because they often produce high least-squares errors. Scrolling through all recommendations would be similar to inspecting and comparing all plots in a scatter plot matrix, and can therefore take a long time. However, here the coloring can come in useful. When coloring the polygon lines according to the order of a particular dimension, other uncorrelated dimensions usually look similar to each other when the line opacity is low enough. In contrast, dimensions with clear correlations often seem to show the color gradient at least partially. An example can be seen in Figure 4.1. This, however, is not a reliable method to detect correlations and cannot be used in every scenario. Additionally, it would still require the user to try out each dimension as the color defining dimension.

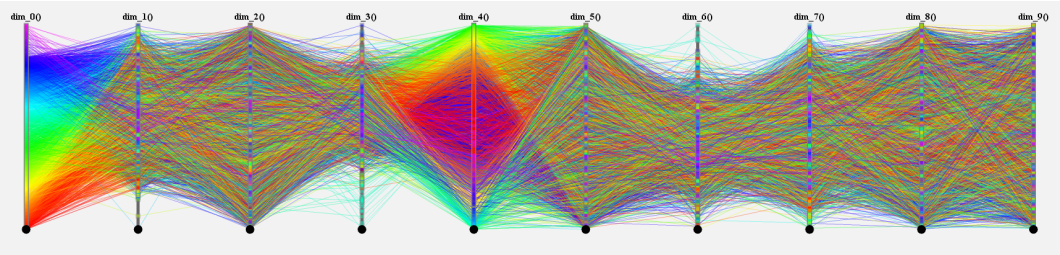

Figure 4.1: While there are no clear polyline patterns visible, the color gradient defined by 'dim_0' can be found again in 'dim_4'. This suggests that there exists a correlation between these two dimensions.

There is currently no automated way to find clusters. However, clusters can be found found in 2D scatter plots or even in the PCP view when comparing two dimensions. After clusters are found the dynamic coloring allows users to easily track the polygon lines through all dimensions and compare them. Additionally, single clusters can be selected for further analysis in higher layers.

4.1.2 Structures in Three Dimensions

While many 3D structures can be found by looking at the possible 2D scatter plot combinations, and can therefore be found via the 2D recommender system, some structures are not axis-aligned and are harder to find. This means that such patterns can be hard to find in high-dimensional data sets, because many 3D scatter plot combinations would have to be tested. However, once a 3D scatter plot has been created, 3D structures are clearly visible when rotating the scatter plot in 3D.

More examples of analyses of a 3D data set can be seen in Figure 4.2. After analyzing this data set, a user could describe the data to lie on the surface of two disjointed concentric 3D cylinders. This means that there are two distinct clusters. After selecting the inner cylinders and unrolling it, it is revealed that the surface of the smaller cylinder is shaped like a smiley face. This means that the inner cylinder also consists of four smaller clusters, two for the eyes, one for the mouth and one for the outer circle. When viewing

the data in a simple 3D scatter plot, it would be hard to recognize such patterns, or how the data is distributed on the surface of such a cylinder.

4.1.3 Structures in Higher Dimensions

To demonstrate how high dimensional structures can be analyzed with our approach, a data set has been synthetically created. A screenshot of the analysis can be seen in Figure 4.3. Naturally, the structure of the data set was known beforehand, so this example is not an accurate illustration of how easy or hard it is to find structures but rather should demonstrate what kinds of statements can be made about a data set just by looking at the finished analysis.

The most prominent feature of this data set is that all points, except for three outliers lie on a four-dimensional dragon shape. This dragon shape has been rolled up into a spiral in dimensions 0 and 4. After the user interactively unrolls this spiral to a new axis (Subsection 3.1.2) and merges it with dimensions 1 and 7 (Subsection 3.1.1), it is revealed that there is a dragon shaped structure, which is not clearly aligned with any of the three dimensions, but rather is rotated 45 degrees along the unrolled dimension. Additionally, the dragon shape has been extruded in the other two dimensions. By projecting the dragon shape onto a plane (Subsection 3.1.2), a dimension can be removed with minimal loss of information.

Furthermore, there are two clusters in dimensions 3 and 6, both of which are 2D Gaussian distributions. After selecting the smaller cluster for further analysis, it is revealed that there is a strong nonlinear correlation between dimensions 2 and 8, which was previously hidden by the larger cluster. Dimensions 5 and 9 only contain noise and can be disregarded when only considering the topology of the data.

Finally, some statements can be made about the connection between these patterns. By coloring the points along the unrolled dimension, it can be seen that the dragon's tail corresponds to the innermost part of the spiral, and also to the smaller cluster visible in the combination of dimensions 3 and 6, which in turn form the parabola in dimensions 2 and 8.

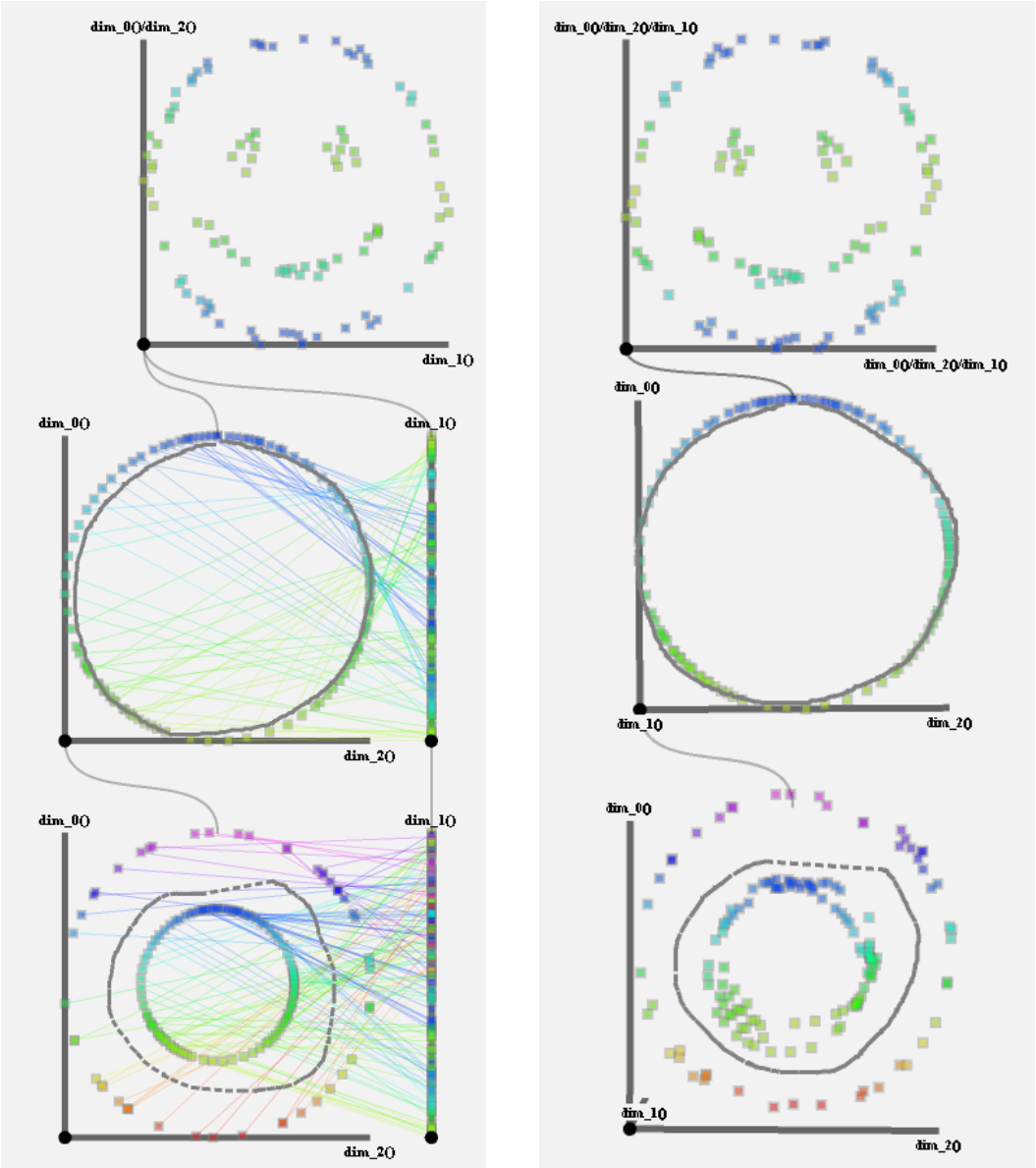

Figure 4.2: A synthetically created data set with three dimensions. (a) After selecting only the data points of the inner cluster, the data is parametrized by sketching along the circle to reduce the number of dimensions. A merge action with the third dimension reveals a smiley face. (b) An alternative way to discover the smiley face. All three dimensions are merged in the first layer. The data points of the inner cylinder are selected, and then unrolled to reveal the smiley face.

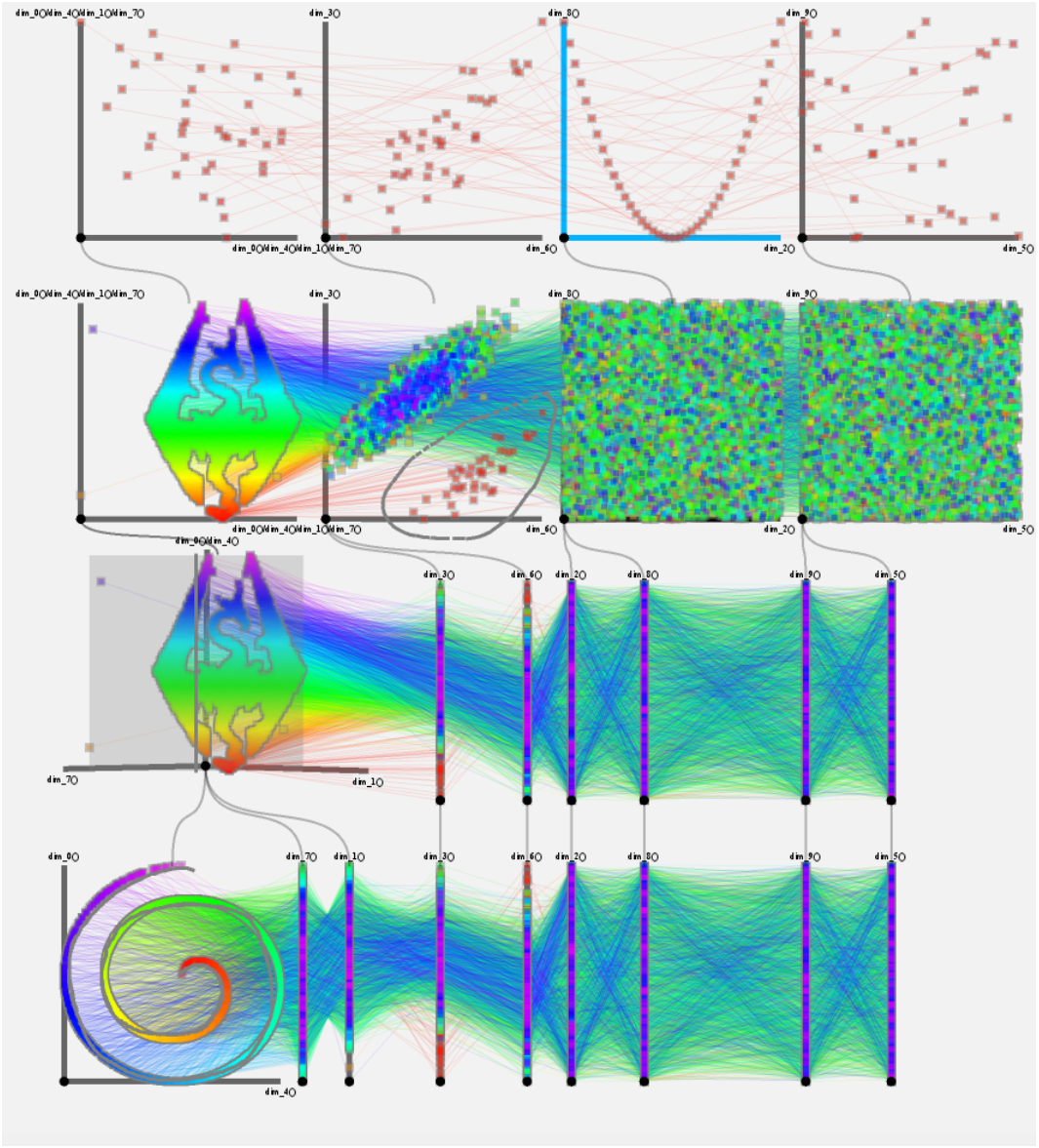

Figure 4.3: An analysis of a synthetically created data set with ten dimensions. The dimensions 0, 1, 4 and 7 form a 4D dragon shape. This dragon shape appears to be curled into a spiral in the dimensions 0 and 4. There appear to be two clusters in dimensions 3 and 6 (third layer). The smaller cluster has a strong quadratic correlation between dimensions 2 and 8 (top layer). Dimensions 5 and 9 only contain white noise.

To summarize, it can be said that when disregarding the completely noisy dimensions 5 and 9, the data appear to lie on a rolled-up dragon shape which can be separated into two clusters visible in the combination of dimensions 3 and 6. While the dimensions 2 and 8 are also mostly noisy, they also contain a strong quadratic correlation of the points contained in the smaller cluster, which is also the dragon's tail.

While this data set has been created synthetically with the explicit purpose of demonstrating the capabilities of our technique, we believe that a similar analysis would be hard to conduct with conventional data analysis approaches.

As previously mentioned, this example is not representative of natural data sets, for which the structure is not known beforehand as it does not consider the exploration process. For this reason, a user study was conducted with this exact data set, to see if other users can effectively use this tool, and understand how the data is visualized. In the next chapter, we give a detailed description of this user study and its results.

5 User Study

In order to evaluate whether users can understand the visualization method of the developed prototype and to see whether they can effectively and efficiently find interesting patterns in high-dimensional data sets, a user study was conducted. The study contained 11 participants who used the tool for explorative analysis in individual interactive sessions. While this number of participants is not enough to draw statistically significant conclusions, it still provided an overview of how uninitiated users would understand the visualizations and how they interact with the application.

5.1 Setup

Because the prototype is meant for expert users to analyze large, high-dimensional data sets, only users with some experience in information visualization or data analysis were asked to participate. The study was conducted online via video calls, and was divided into four parts. The users were asked to participate separately, so that no information was shared between them.

First, the participants were given a 10-minute demo of our tool to show how data can be visualized, explored, and the visualization be manipulated. Although we assumed all participants to have basic knowledge in information visualization, the introduction also included a short summary of how scatter plots and parallel coordinates visualize data, as well as a brief overview of TDA. This demo was conducted by running the prototype on the study supervisor's computer and sharing the screen to the participants. For the demo a data set with 150 data points and 7 dimensions was chosen. The data set contained 2 clusters which could be found in all dimensions

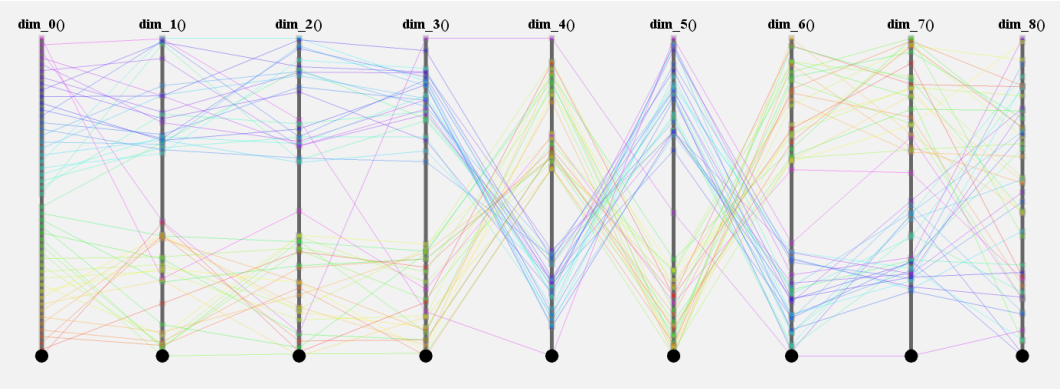

Figure 5.1: The dataset used to show participants the tool's various functions.

but one. This last dimensions contained only noise. A PCP visualization of this dataset can be seen in Figure 5.1

During the second part of the study, the participants were asked to run the prototype on their computer and share their screen. They were given 15 minutes to analyze a second data set with 150 data points and 8 dimensions and were explicitly asked to look for clusters, outliers, topological structure or correlations. During this time, the supervisor would answer any questions they had and give some helpful advice on how the visualization could be interpreted or how they might proceed. The data set contained three clusters in several dimensions, two concentric circles in two dimensions and a polynomial correlation of degree 3 in two other dimensions. A visualization of the second dataset can be seen in Figure 5.2.

The third and longest part of the study consisted of the users again running the tool on their own computer, to analyze a larger data set with 3736 data points and 10 dimensions. The data set for this part is the data set shown in Figure 4.3. Once again, users were explicitly asked to look for clusters, outliers, topological structures or correlations. Additionally, they were asked to think out loud and to verbalize any problems they might encounter or patterns they find interesting. This time, however, the supervisor provided as little help as possible and did not give hints on how a user might proceed. The supervisor did, however, ask questions to see if the users understood how the data was visualized and what statements could be made about the structure and topology of the data.

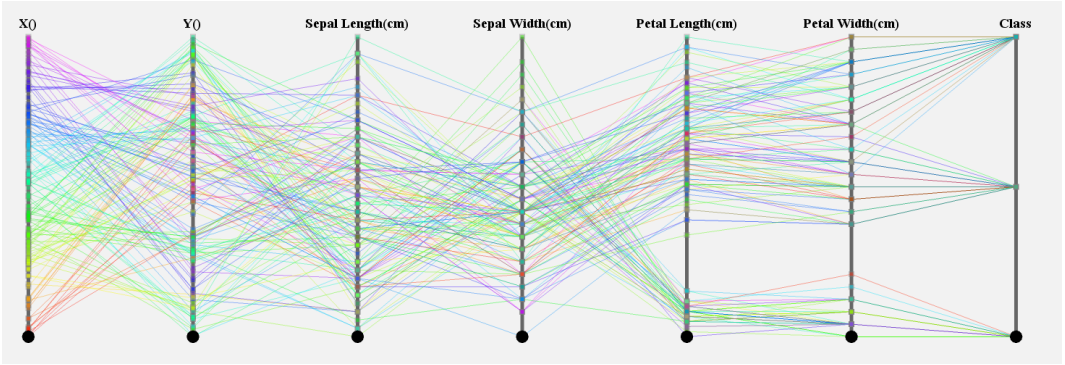

Figure 5.2: The second dataset used during the study. Users used this dataset to try out the tool on their own. This is a version of the popular 'iris' dataset (Anderson, 1935) with 3 additional artificially created dimensions.

Finally, users were asked to answer several questions about themselves and their experience during the study in order to record their thoughts and opinions, and to compare results. A list of the questions can be found in Table 5.1 and Table 5.2, along with the participants' answers.

5.2 Study Results

After the study was concluded, the questionnaire data was aggregated and evaluated. The participants' answers can be found in Table 5.1 and Table 5.2.

The questionnaire contained questions about the participants' age, gender, their highest degree of education and whether they have a color vision deficiency. To keep the participants' identities anonymous, the answers to these questions are not listed separately. All the participants were age 27 - 50, the participants' highest degrees varied between BSc., MSc. and PhD and none of them had a color vision deficiency. Three participants were women and the remaining eight participants were men.

The questionnaire contained two questions about the structure of the data set.

User ID	1	2	3	4	5	6	7	8	9	10	11	Avg.
How familiar were you with scatter plots prior to this study?	5	3	3	5	4	3	4	5	5	5	2	4.00
How familiar were you with paralell coordinate plots prior to this study?	4	1	4	4	2	1	4	5	5	5	3	3.45
How familiar are you with topological data analysis?	3	1	1	2	1	3	3	3	3	4	2	2.36
How useful did you find the dynamic polyline coloring for exploring the dataset?	4	2	5	4	4	5	5	5	5	4	5	4.36
How useful did you find inline 2D scatter plots for exploring this dataset	5	5	5	4	5	5	4	5	5	5	5	4.82
How useful did you find the 2D scatter plot previews for exploring the dataset?	5	5	5	4	5	5	4	4	4	4	4	4.45
How useful did you find inline 3D scatter plots for exploring the dataset?	4	5	5	4	5	4	1	2	5	4	5	4.00
How useful did you find the function fit for dimension unwrapping when exploring this dataset?	5	3	4	3	4	2	3	5	2	4	3	3.45
How useful did you find the curve sketch tool for dimension unwrapping when exploring this dataset?	4	4	5	4	5	3	4	2	5	4	3	3.91
How useful did you find the provenance lines for exploring this dataset?	3	3	5	5	4	4	3	5	5	4	3	4.00

Table 5.1: The first half of the questionnaire with the participants' answers. For the first three questions, an answer of 1 represents "Not at all familiar" and an answer of 5 represents "Very familiar. For the remaining questions, an answer of 1 represents "Not at all useful" and an answer of 5 represents "Very useful".

User ID	1	2	3	4	5	6	7	8	9	10	11	Avg.
I think that I would like to use this system frequently.	4	4	5	3	4	4	4	4	5	4	5	4.18
I found the system unnecessarily complex.	2	1	2	3	1	1	2	1	1	2	1	1.55
I thought the system was easy to use.	4	5	4	4	5	4	3	4	5	3	4	4.09
I think that I would need the support of a technical person to be able to use this system.	2	4	1	2	1	2	2	1	1	1	1	1.64
I found the various functions in this system were well integrated.	5	5	5	4	4	5	4	5	5	4	4	4.55
I thought there was too much inconsistency in this system.	3	1	1	3	1	1	1	1	1	2	1	1.45
I would imagine that most people would learn to use this system very quickly.	4	4	5	3	4	3	2	2	5	3	3	3.45
I found the system very cumbersome to use.	3	1	2	2	1	2	2	1	1	1	1	1.55
I felt very confident using the system.	4	3	4	3	5	3	4	4	5	4	4	3.91
I needed to learn a lot of things before I could get going with this system.	3	4	1	3	1	2	1	1	1	1	2	1.82

Table 5.2: The second half of the questionnaire with the participants' answers. These questions comprise the System Usability Scale (Brooke, 1996), in which an answer of 1 represents "Strongly disagree" and an answer of 5 represents "Strongly agree".

- "After analyzing the data set with the tool, how would you describe its structure (clusters, correlations, topological features, etc.)?"
- "Which features of the dataset did you find interesting (one paragraph)?"

As mentioned before, the data set which the questions refer to is visualized in Figure 4.3. All users, without exception, found the most prominent structures in two dimensions, that being: the spiral between dimensions 0 and 4, the two clusters between dimensions 3 and 6 (although some users described the smaller cluster as a set of outliers) and the diamond shape between dimensions 1 and 7, which is part of the four-dimensional dragon shape. These three structures were also the ones the participants found most interesting, especially the spiral.

Four users found the quadratic polynomial correlation between dimensions 2 and 8, which is only contained in the subset of the data that is the smaller of the two clusters in dimensions 3 and 6. Two users found the 4-dimensional dragon shape and one found that there was a pattern on the 3D spiral after unrolling one dimension, but could not identify it, and did not think of using the curve sketch tool to unroll the spiral. Four users found that there are some outliers which can be found between dimensions 1 and 7. All users expressed that the data set contained a lot of noise and some even explicitly stated, that dimensions 2, 5, 8 and 9 only contain noise. This is mostly correct, except for the quadratic polynomial, which can only be seen after selecting a subset of the data. 5 of the 11 participants also found varying correlations in higher dimensions. For example, the fact that the inner part of the spiral consists of the smaller of the two clusters or that there are several correlations between the rhombus shape in the dimensions 1 and 7 and the spiral shape in dimensions 0 and 4. Both of these examples showcase correlations in 4 dimensions.

Concerning the first 3 questions, it can be said that the participants were generally familiar with data visualization techniques and most of them had at least heard of topological data analysis. This means that the participants can be seen as an expert group in the field of data visualization. This was a deliberate choice, because the prototype is meant to be primarily used by individuals with a certain degree of expertise in this field. The users found the basic functionalities such as the inline 2D and 3D scatter plots,

the polyline coloring and the 2D previews very useful, with average scores greater than or equal to 4.0 with the inline 2D scatter plots achieving the highest average score of 4.82.

Most participants found both the curve sketch tool and the function fit tool to be useful, with average scores of 3.45 and 3.91. Here it is worth mentioning that the two lowest scores of 2 were given by people with a high level of expertise in data visualization. This might be the case because these users were familiar with other tools and did not quite know how to use this new approach to extract additional information from the data. Users with more experience in data visualization also tended to spend more time analyzing the parallel coordinates instead of experimenting with the different tools and trying to merge and unmerge 2D and 3D scatter plots, which is not the optimal approach for analyzing this data set. The most useful features, according to the participants, are the inline 2D scatter plots and the 2D scatter plot previews, with average scores of 4.82 and 4.45 respectively. This might indicate that most participants had more previous experience interpreting 2D scatter plots than PCPs, which appears plausible. More surprising was how useful most people found the dynamic polyline coloring. While this feature was not the main focus of this research, it seems that the simple ability to use a PCP axis to set a rainbow color scheme for the polylines seemed effective. Lastly, the provenance lines and the 3D scatter plots scored an average of 4.0. For a clearer picture of the answer distributions, see Figure 5.3, Figure 5.4, Figure 5.5 and Figure 5.6.

The participants were also asked the questions contained in the System Usability Scale (Brooke, 1996). Overall, the usability of the prototype seemed relatively high, with participants strongly agreeing with most positive statements (average above 4.0) and disagreeing with negative statements (average below 2.0). The only two statements with a less positive response were "I would imagine that most people would learn to use this system very quickly" and "I felt very confident using the system".

Concerning the first statement, most participants questioned what was meant by "most people", because they thought that it would be quite difficult for the average population to use the prototype effectively. However, since this technique was developed for data analysts using it to analyze complex

data sets, these questions might not accurately represent the time needed to learn how to use the system for expert users.

The statement with the second lowest score concerned the confidence of the users while using the prototype. The answers to this question seemed to correlate with the user's previous knowledge of scatter plots, parallel coordinates and topological data analysis. This again shows that the application is meant for users with previous knowledge of data visualization.

While the remaining statements showed that the system's usability was decent, there are several chances for improvement of the user's experience when using a system. A discussion of possible future improvements is given in Section 6.2.

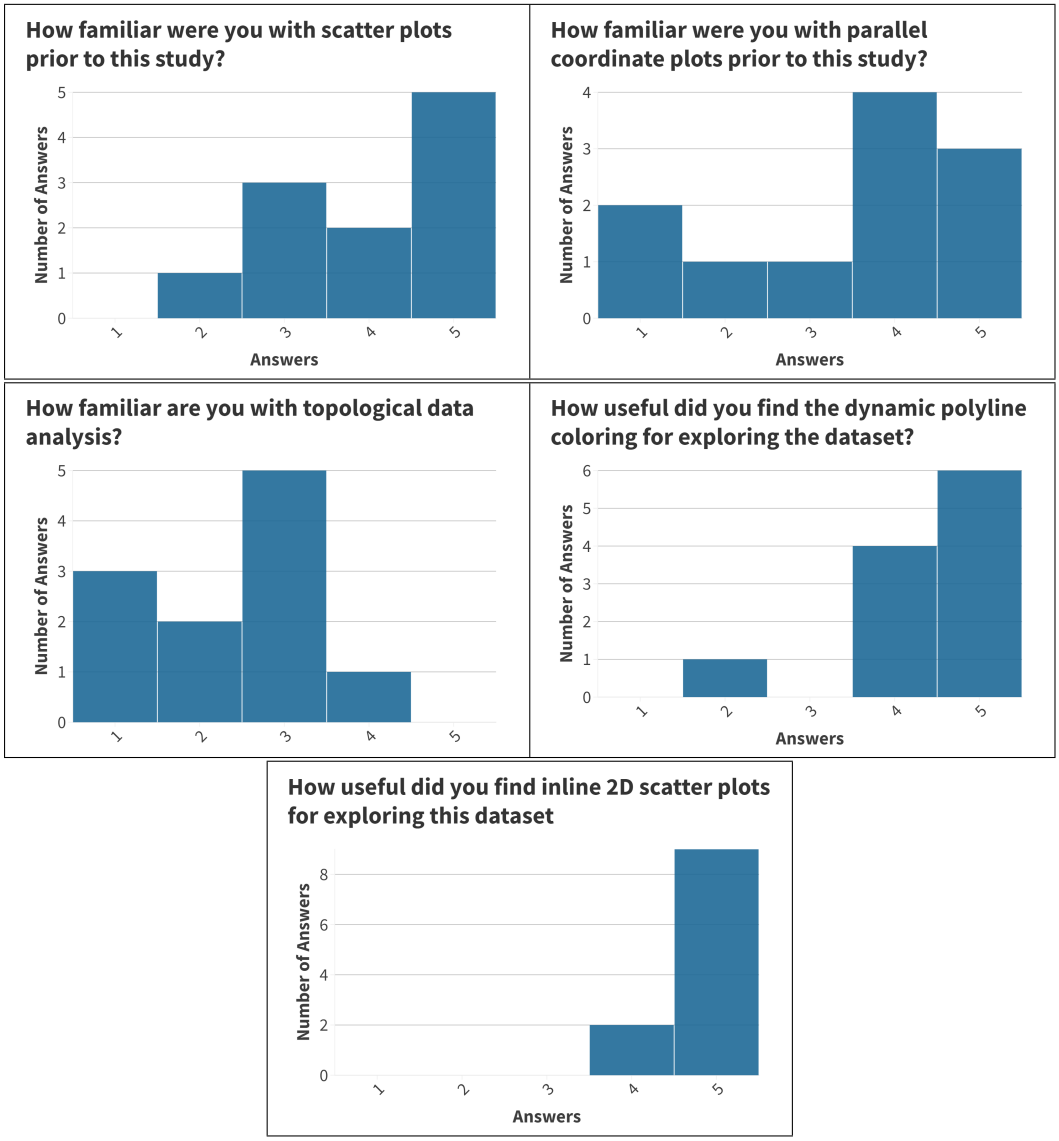

Figure 5.3: Histograms of the results of the questions concerning the participants' previous knowledge and the usefulness of the prototypes' functionalities. For the first three questions, an answer of 1 represents "Not at all familiar" and an answer of 5 represents "Very familiar". For the last two questions, an answer of 1 represents "Not at all useful" and an answer of 5 represents "Very useful".

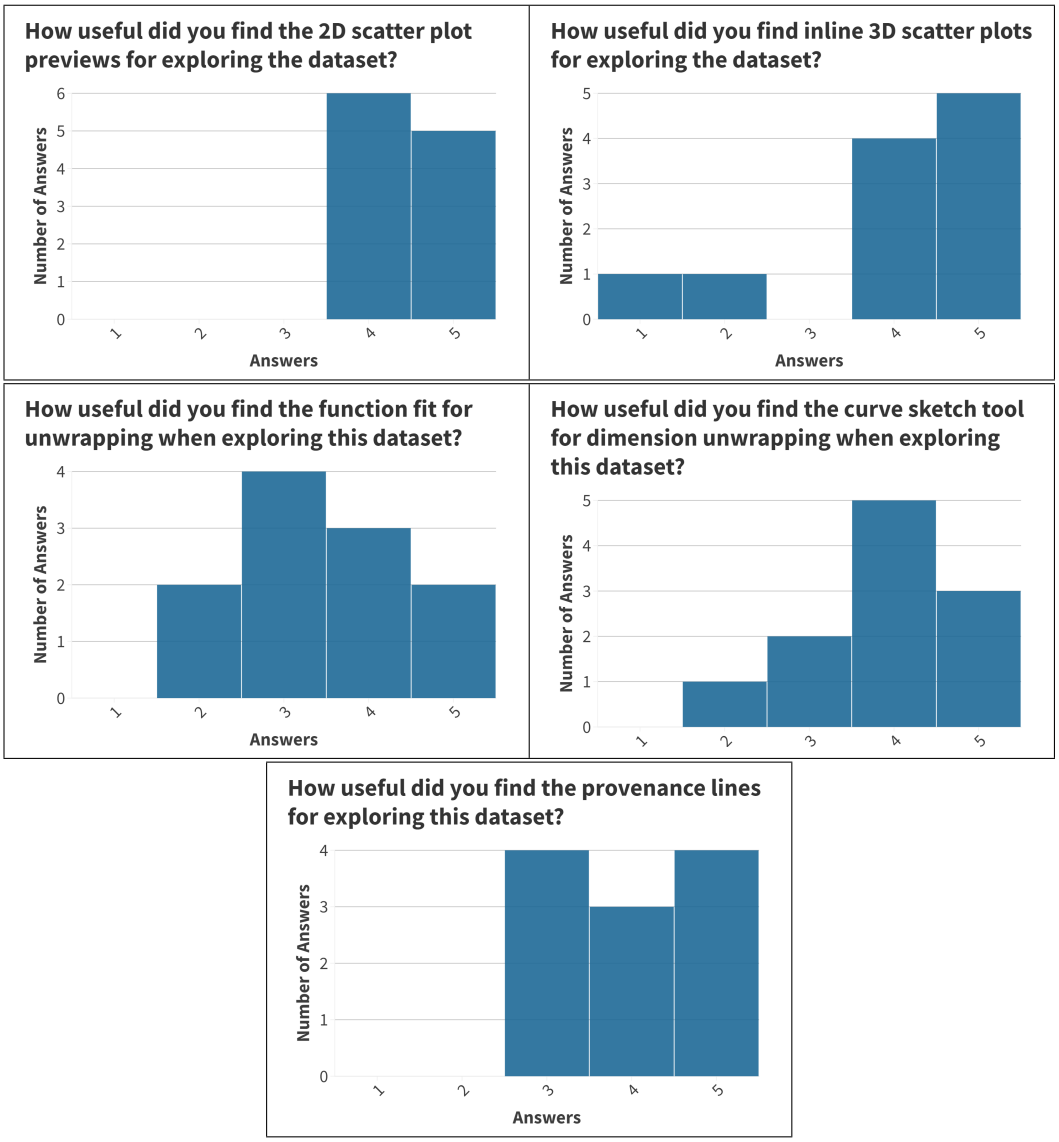

Figure 5.4: Histograms of the results of 5 questions concerning the usefulness of the prototype's functionalities. An answer of 1 represents "Not at all useful" and an answer of 5 represents "Very useful".

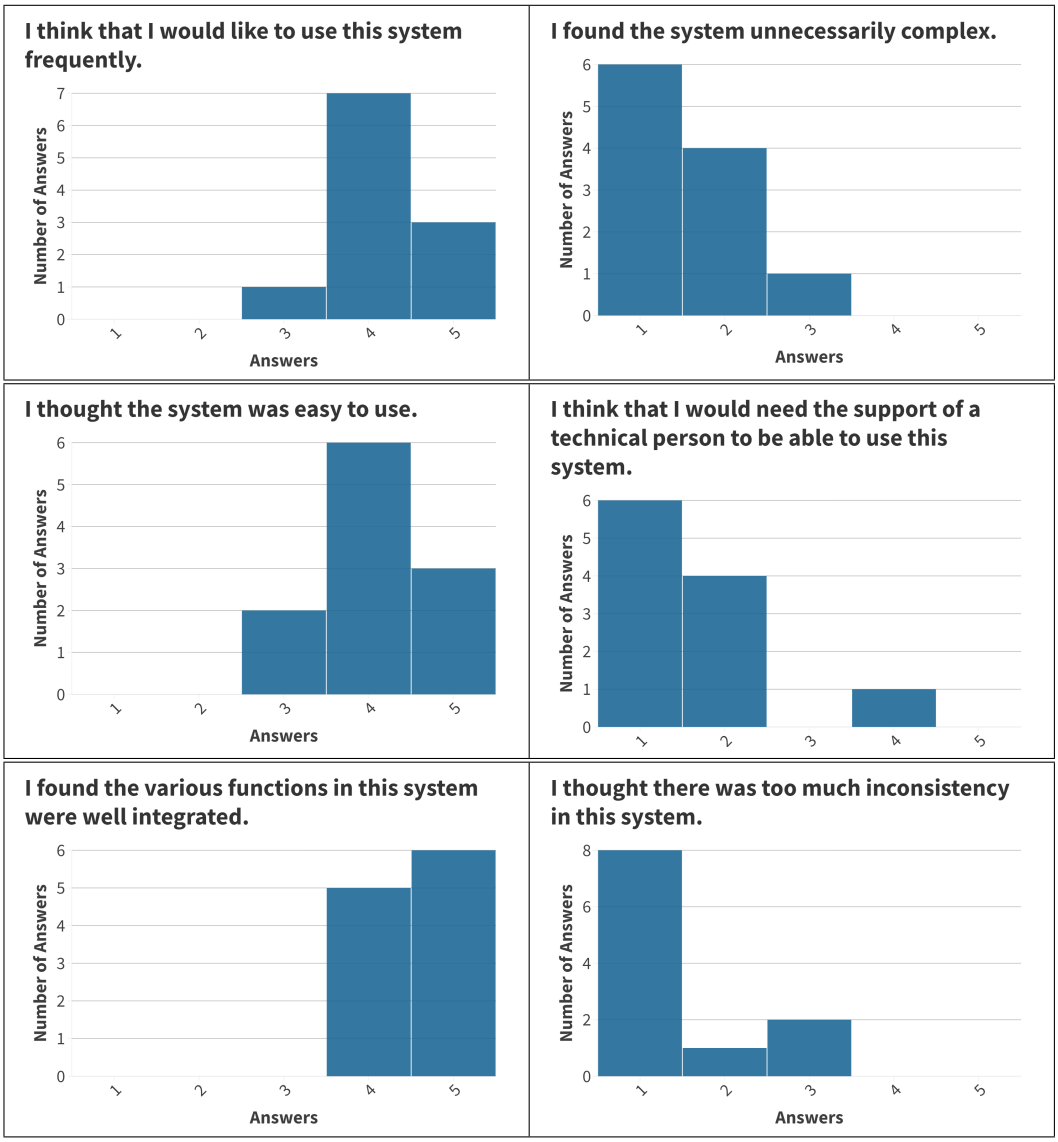

Figure 5.5: Histograms of the results of the first 6 questions of the System Usability Scale. An answer of 1 represents "Strongly disagree" and an answer of 5 represents "Strongly agree". The scores show that overall the system was relatively easy to use.

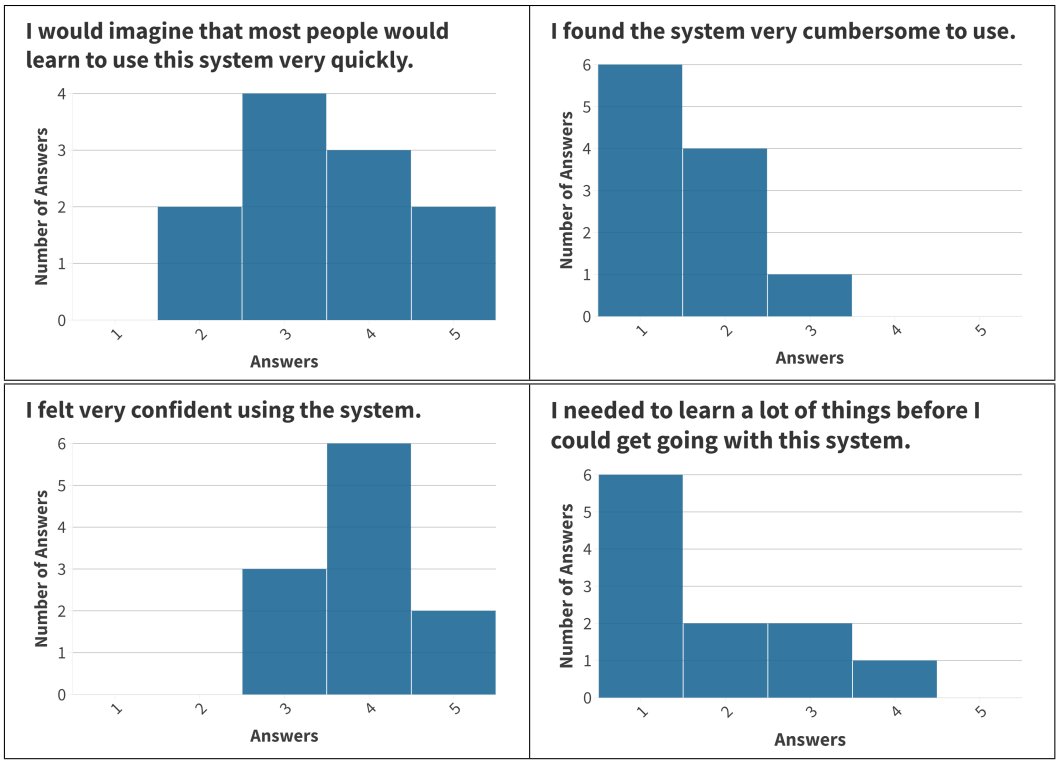

Figure 5.6: Histograms of the results of the last 4 questions of the System Usability Scale. An answer of 1 represents "Strongly disagree" and an answer of 5 represents "Strongly agree". The scores show that overall the system was relatively easy to use.

6 Conclusion

While the previous chapter already contained the results of the user study, this chapter will contain a summary of the work done, an overview of the user study's results as well as an outlook to the future and some reading recommendations.

6.1 Results

A new and interactive way to visualize and analyze high-dimensional data sets was developed in the form of a software prototype. The prototype combined several popular visualization techniques such as 2D and 3D scatter plots, parallel coordinates and a recommender system to give users an idea of what to explore next. The prototype was written in C++ with OpenGL and Qt as user interface and allows for the visualization of many thousands of data points simultaneously in tens of dimensions.

The new interactive approach of unrolling embedded dimensions allows expert users to reduce dimensions, explore high-dimensional manifolds and find structures in high-dimensional spaces. The parametrization of subspaces can also be used to separate clusters and parametrize structures which can not be described by a normal function, such as circles, spirals, or other overlapping shapes. The prototype also offers a variety of tools to explore and analyze subsets of two and three dimensions and find correlations between such subspaces. The recommender system for interesting 2D subspaces worked well, but might not be sufficient to explore data sets with hundreds of dimensions.

The user study showed that users can use the prototype's functions effectively, but that users did not immediately know how to effectively use the

curve sketch tool. More time might be needed to get a good understanding of how this functionality can be applied. However, users were able to find structures in higher dimensions and understand how they related to the original data. The user study also showed that currently the exploration process is still heavily reliant on trial and error and that more and stronger recommender systems are required.

6.2 Future Work

The current prototype is a proof of concept which illustrates the analysis methods that are possible by unrolling embedded dimensions. To develop the prototype into a practically applicable tool, some more features could be implemented and a bigger emphasis should be placed on usability.

The most requested feature during the user study was a recommendation system for 3D scatter plots. This could be done similarly to how the 2D recommendation system in the current version of the prototype. Instead of 2D scatter plots, the 3D scatter plots could be shown in a small popup, while slowly rotating to give users different angles to view the data. The ordering could again be done by fitting a plane into the 3D space and sorting by the least squares error.

Currently, the application requires a data set to be selected on startup, but for a stand-alone application it would be necessary to have 'load' and 'save' options to save the current analysis state, and later load it again. This would also allow users to more easily switch between different data sets.

Other tools offer more interaction methods with the parallel coordinate plot. For example: selection of poly line bundles (currently, only the selection of data points on 1D scatter plots is possible), different color schemes (currently only the rainbow color scheme is available) or dynamic opacity settings to highlight more or less dense parts of the PCP. Especially, a selection of different color schemes would seem like a useful feature, not just because it would also allow users with a color vision deficiency to use the tool, but also because the current rainbow color scheme is generally not regarded

as a good color scheme to convey information (Stoelzle and Stein, 2021, Borland and Ii, 2007).

The current version of the prototype is also limited in the number of dimensions that can be analyzed because of the limited recommendation and exploration options. A 3D merge recommendation has been mentioned before, but other visualizations such as a scatter plot matrix or star coordinates could also be integrated and could help users to explore larger data sets, or at least to give them a starting point in their analysis.

The most novel features of the prototype are the different possibilities a user has to project points onto a line or plane. Currently, there are three options:

- Draw a line by hand. In case of a 3D scatter plot, this line is extruded in the third dimension.
- Automatically fit a polynomial in two dimensions.
- project 3D points onto a plane parallel to the screen.

All of these options could be enhanced and more could be added. For example, when drawing by hand, it might be helpful to be able to draw straight lines or poly lines instead of drawing completely free hand. The automatic polynomial fit could be extended to 3D, which would be necessary either way for a 3D merge recommendation system. Finally, all 3D drawings, including the plane projections, could be modified in retrospect by dragging edges or vertices of the 3D plane, similarly to how such modifications are made in 3D modelling software.

Finally, it is worth mentioning that a second user study with more participants could give better insights into what users struggle with, and which additional features would help users in analyzing high-dimensional data sets. It could also be interesting to see how users participating in the first study benefited from additional features such as a 3D recommendation system and how much their analysis improved.

6.3 Further Reading

While many of the cited publications are interesting in their own right, this section provides some reading recommendations, chosen by the author.

In case the reader finds themselves interested in Topological Data Analysis in general, Wasserman (2018) is a great starting place. It also served as a starting point for the first chapter of this thesis, because it is easily readable, contains illustrations to make points more easily digestible, and references many other publications which can be sought out to further deepen the reader's knowledge.

While it was not the focus of this thesis, dimension reduction is an interesting and important topic when discussing high-dimensional data. While some techniques such as PCA (Wold, Esbensen, and Geladi, 1987) are generally well-known, Cunningham (2008) provides a well-structured and detailed list of many dimension reduction techniques. The techniques are well explained and compared to understand what their respective strengths and weaknesses.

Similar to how Cunningham (2008) provides an overview of different dimension reduction techniques, Chan (2006) and Grinstein and Trutschl (2001) provide great overviews of high-dimensional data visualization techniques. They both include pictures of the mentioned visualizations, which allows for a quick overview and can serve as a starting points if the user is interested in any specific kind of visualization.

Bibliography

Anderson, Edgar (1935). "The irises of the Gaspe Peninsula." In: *Bull. Am. Iris Soc.* 59, pp. 2–5 (cit. on pp. 35, 47).

Arias-Castro, Ery, David Mason, and Bruno Pelletier (2016). "On the estimation of the gradient lines of a density and the consistency of the mean-shift algorithm." In: *The Journal of Machine Learning Research* 17.1, pp. 1487–1514 (cit. on p. 3).

Asimov, Daniel (1985). "The grand tour: a tool for viewing multidimensional data." In: *SIAM journal on scientific and statistical computing* 6.1, pp. 128–143 (cit. on p. 9).

Bellman, Richard and Robert Kalaba (1959). "A mathematical theory of adaptive control processes." In: *Proceedings of the National Academy of Sciences of the United States of America* 45.8, p. 1288 (cit. on p. 5).

Borland, David and Russell M Taylor Ii (2007). "Rainbow color map (still) considered harmful." In: *IEEE computer graphics and applications* 27.2, pp. 14–17 (cit. on p. 59).

Brand, Matthew (2002). "Charting a manifold." In: *Advances in neural information processing systems* 15 (cit. on p. 2).

Brooke, John (1996). "Sus: a "quick and dirty usability scale." In: *Usability evaluation in industry* 189.3 (cit. on pp. 49, 51).

Carlsson, Gunnar (2009). "Topology and data." In: *Bulletin of the American Mathematical Society* 46.2, pp. 255–308 (cit. on p. 1).

Chacón, José E (2012). "Clusters and water flows: a novel approach to modal clustering through morse theory." In: *arXiv e-prints*, arXiv–1212 (cit. on p. 3).

Chacón, José E and Tarn Duong (2013). "Data-driven density derivative estimation, with applications to nonparametric clustering and bump hunting." In: *Electronic Journal of Statistics* 7, pp. 499–532 (cit. on p. 3).

Chan, Winnie Wing-Yi (2006). "A survey on multivariate data visualization." In: *Department of Computer Science and Engineering. Hong Kong University of Science and Technology* 8.6, pp. 1–29 (cit. on p. 60).

Cheng, Yizong (1995). "Mean shift, mode seeking, and clustering." In: *IEEE transactions on pattern analysis and machine intelligence* 17.8, pp. 790–799 (cit. on p. 3).

Chernoff, Herman (1973). "The use of faces to represent points in k-dimensional space graphically." In: *Journal of the American statistical Association* 68.342, pp. 361–368 (cit. on p. 11).

Comaniciu, Dorin and Peter Meer (2002). "Mean shift: A robust approach toward feature space analysis." In: *IEEE Transactions on pattern analysis and machine intelligence* 24.5, pp. 603–619 (cit. on p. 3).

Cunningham, Pádraig (2008). "Dimension reduction." In: *Machine learning techniques for multimedia*. Springer, pp. 91–112 (cit. on pp. 8, 60).

Deerwester, Scott et al. (1990). "Indexing by latent semantic analysis." In: *Journal of the American society for information science* 41.6, pp. 391–407 (cit. on p. 8).

DeMers, David and Garrison Cottrell (1992). "Non-linear dimensionality reduction." In: *Advances in neural information processing systems* 5 (cit. on p. 2).

Edelsbrunner, Herbert et al. (2000). "Topological persistence and simplification." In: *Proceedings 41st annual symposium on foundations of computer science*. IEEE, pp. 454–463 (cit. on p. 1).

Elmqvist, Niklas, Pierre Dragicevic, and Jean-Daniel Fekete (2008). "Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation." In: *IEEE transactions on Visualization and Computer Graphics* 14.6, pp. 1539–1148 (cit. on pp. 15, 16).

Fanea, Elena, Sheelagh Carpendale, and Tobias Isenberg (2005). "An interactive 3D integration of parallel coordinates and star glyphs." In: *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* IEEE, pp. 149–156 (cit. on p. 12).

Fisher, Ronald A (1936). "The use of multiple measurements in taxonomic problems." In: *Annals of eugenics* 7.2, pp. 179–188 (cit. on p. 8).

Fodor, Imola K (2002). *A survey of dimension reduction techniques*. Tech. rep. Lawrence Livermore National Lab., CA (US) (cit. on p. 8).

Friendly, Michael and Daniel Denis (2005). "The early origins and development of the scatterplot." In: *Journal of the History of the Behavioral Sciences* 41.2, pp. 103–130 (cit. on p. 13).

Genovese, Christopher R et al. (2014). "Nonparametric ridge estimation." In: *The Annals of Statistics* 42.4, pp. 1511–1545 (cit. on pp. 3, 4).

Grinstein, Georges and Marjan Trutschl (Jan. 2001). "High-Dimensional Visualizations." In: *In: 7th Data Mining Conference-KDD 2001: San Francisco, California* (cit. on p. 60).

Heinrich, Julian, John T Stasko, and Daniel Weiskopf (2012). "The Parallel Coordinates Matrix." In: *EuroVis (Short Papers)* (cit. on p. 18).

Heinrich, Julian and Daniel Weiskopf (2013). "State of the Art of Parallel Coordinates." In: *Eurographics (State of the Art Reports)*, pp. 95–116 (cit. on p. 17).

Hewes, Fletcher and Henry Gannett (1883). *Scribner's statistical atlas of the United States, showing by graphic methods their present condition and their political, social and industrial development*. New York, C. Scribner's sons [c1883] (cit. on p. 16).

Inselberg, A. and B. Dimsdale (1990). "Parallel coordinates: a tool for visualizing multi-dimensional geometry." In: *Proceedings of the First IEEE Conference on Visualization: Visualization '90*, pp. 361–378. DOI: 10.1109/VISUAL.1990.146402 (cit. on pp. 16, 17).

Jäckle, Dominik et al. (2017). "Pattern trails: visual analysis of pattern transitions in subspaces." In: *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, pp. 1–12 (cit. on pp. 8, 21).

Johansson Westberg, Jimmy, Camilla Forsell, and Matthew Cooper (Feb. 2013). "On the usability of three-dimensional display in parallel coordinates: Evaluating the efficiency of identifying two-dimensional relationships." In: *Information Visualization* 13, pp. 29–41. DOI: 10.1177/1473871613477091 (cit. on pp. 19, 20).

Kandogan, Eser (2000). "Star coordinates: A multi-dimensional visualization technique with uniform treatment of dimensions." In: *Proceedings of the IEEE information visualization symposium*. Vol. 650. Citeseer, p. 22 (cit. on pp. 9, 10).

Kaufman, Leonard and Peter J Rousseeuw (2009). *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons (cit. on p. 2).

Kovacev-Nikolic, Violeta et al. (2016). "Using persistent homology and dynamical distances to analyze protein binding." In: *Statistical applications in genetics and molecular biology* 15.1, pp. 19–38 (cit. on p. 1).

Lee, Yong Jae, C Lawrence Zitnick, and Michael F Cohen (2011). "Shadowdraw: real-time user guidance for freehand drawing." In: *ACM Transactions on Graphics (TOG)* 30.4, pp. 1–10 (cit. on p. 16).

Li, Jing, Jean-Bernard Martens, and Jarke J Van Wijk (2010). "Judging correlation from scatterplots and parallel coordinate plots." In: *Information Visualization* 9.1, pp. 13–30 (cit. on p. 18).

Macrae, C Neil and Galen V Bodenhausen (2000). "Social cognition: Thinking categorically about others." In: *Annual review of psychology* 51.1, pp. 93–120 (cit. on p. 2).

McInnes, Leland, John Healy, and James Melville (2018). "Umap: Uniform manifold approximation and projection for dimension reduction." In: *Journal of Open Source Software* 3.29, p. 861 (cit. on p. 2).

Montgomery, Douglas C, Elizabeth A Peck, and G Geoffrey Vining (2021). *Introduction to linear regression analysis*. John Wiley & Sons (cit. on p. 27).

Munch, Elizabeth (July 2017). "A User's Guide to Topological Data Analysis." In: *Journal of Learning Analytics* 4, pp. 47–61. DOI: 10.18608/jla.2017.42.6 (cit. on p. 2).

Ozertem, Umut and Deniz Erdogmus (2011). "Locally defined principal curves and surfaces." In: *The Journal of Machine Learning Research* 12, pp. 1249–1286 (cit. on p. 3).

Shao, Lin, Michael Behrisch, et al. (2014). "Guided Sketching for Visual Search and Exploration in Large Scatter Plot Spaces." In: *EuroVA@ EuroVis* (cit. on p. 15).

Shao, Lin, Nelson Silva, et al. (2017). "Visual exploration of large scatter plot matrices by pattern recommendation based on eye tracking." In: *Proceedings of the 2017 ACM Workshop on Exploratory Search and Interactive Data Analytics*, pp. 9–16 (cit. on p. 15).

Stoelzle, Michael and Lina Stein (2021). "Rainbow color map distorts and misleads research in hydrology–guidance for better visualizations and science communication." In: *Hydrology and Earth System Sciences* 25.8, pp. 4549–4565 (cit. on p. 59).

Teh, Yee and Sam Roweis (2002). "Automatic alignment of local representations." In: *Advances in neural information processing systems* 15 (cit. on p. 2).

Tenenbaum, Joshua B, Vin de Silva, and John C Langford (2000). "A global geometric framework for nonlinear dimensionality reduction." In: *science* 290.5500, pp. 2319–2323 (cit. on pp. 2, 6).

Tilouche, Shaima, Vahid Partovi Nia, and Samuel Bassetto (2021). "Parallel coordinate order for high-dimensional data." In: *Statistical Analysis and Data Mining: The ASA Data Science Journal* 14.5, pp. 501–515 (cit. on p. 18).

Wasserman, Larry (2018). "Topological Data Analysis." In: *Annu. Rev. Stat. Appl* 5, pp. 501–32 (cit. on pp. 2–6, 60).

Wegenkittl, Rainer, Helwig Löffelmann, and Eduard Gröller (1997). "Visualizing the behaviour of higher dimensional dynamical systems." In: *Proceedings. Visualization'97 (Cat. No. 97CB36155)*. IEEE, pp. 119–125 (cit. on p. 19).

Wold, Svante, Kim Esbensen, and Paul Geladi (1987). "Principal component analysis." In: *Chemometrics and intelligent laboratory systems* 2.1-3, pp. 37–52 (cit. on pp. 2, 8, 60).

Wong, Pak Chung and R Daniel Bergeron (1997). "Multivariate visualization using metric scaling." In: *Proceedings. Visualization'97 (Cat. No. 97CB36155)*. IEEE, pp. 111–118 (cit. on p. 21).

Worsley, Keith J (1995). "Boundary corrections for the expected Euler characteristic of excursion sets of random fields, with an application to astrophysics." In: *Advances in Applied Probability* 27.4, pp. 943–959 (cit. on p. 1).

Yuan, Xiaoru et al. (2009). "Scattering points in parallel coordinates." In: *IEEE Transactions on Visualization and Computer Graphics* 15.6, pp. 1001–1008 (cit. on pp. 8, 20, 22).

Zomorodian, Afra and Gunnar Carlsson (2005). "Computing persistent homology." In: *Discrete & Computational Geometry* 33.2, pp. 249–274 (cit. on p. 1).