Visualization of multivariate data
network monitoring data case study

Claessen, J.H.T.

Award date:
2011

Eindhoven University of Technology
Department of Mathematics and Computer Science
Visualization Group
Master Thesis

Visualization of Multivariate Data

Network monitoring data case study

J.H.T. Claessen
January 2011

Supervisor: prof.dr.ir. Jarke J. van Wijk

Preface

This thesis is the result of my master project done at the department of Mathematics and Computer Science of the Eindhoven University of Technology. This document describes the work for the initial research topic: "*Visualization of network monitoring data*" afterwards generalized to "*Visualization of multivariate data*". The research project conducted in the previous months has led to two results; an application, called FLINAview, and this thesis. Several people have contributed to the results for my master project and I would like to take the opportunity to thank them here.

First of all I would like to thank my supervisor, prof.dr.ir. Jarke J. van Wijk. His support and advice during the project was very beneficial, his enthusiasm was contagious and his ideas valuable. Furthermore I would like to thank dr.ir. Aiko Pras and dr. Anna Sperotto from the University of Twente. They provided the research question, while Anna was my contact person at the University of Twente and provided valuable feedback during the project.

I would like to thank my colleagues from the visualization group for their feedback and ideas. Furthermore I would like to thank all the users that participated in the user study, my friends for their support and all the other persons that contributed to the research project. I would like to thank my boss A.H. Samuel for his flexibility so I could combine my study and work, while being able to fulfill my financial obligations. And finally I would like to thank my parents for their continuous support and for providing me with the opportunity of a good education.

Abstract

The ability to connect computers to one another, by means of a network, has led to computers being more vulnerable for malicious behavior. Network operators have the task to prevent attackers from performing malicious activities on their networks. In this thesis we present an extension to visualization techniques, providing added value for displaying network monitoring datasets. The resulting visualization technique is suitable for displaying multivariate data in general, hence not only applicable to network monitoring dataset.
The main enhancement made to the existing visualization techniques: flexibility, allows for layouts enabling the user for easier and quicker identification of dataset characteristics, structures and outliers. We present cases for several datasets where the introduced idea proved valuable, as well as an interface capable of supporting the suggested visualization technique. The user study proved that the presented idea of *Flexible Linked Axes* is considered useful. The visualization technique provides a solid base for dataset exploration and investigation, and gave new insights to network monitoring experts during the investigating of their datasets.

Contents

Chapter 1

Introduction

Today most computers have a network connection to communicate with other computers. The amount of data passing network connections is rapidly increasing, and new applications and protocols continuously emerge. Not all network connection attempts are of a good nature. There are persons and computers looking to perform malicious activities and gain control over other computers. Network administrators have the task to protect their network from misbehavior.

Since the amount of data is increasing, the network administrators have to monitor more and more data. A variety of algorithms and applications aid the network administrators with their task. The algorithms are only capable of detecting the activities they are intended for, hence not capable to detect new types of malicious behavior. Adjusting the algorithms takes time, while network operators should be able to immediately detect and react when misbehavior occurs.

Visualization can help to support network administrators; to give overviews of analyzed data, but also to enable them to detect malicious behavior. The topic for this research is: "V*isualization of network monitoring data*". The result of this research project is a new method for visualization of network monitoring data, which is also generally applicable to multivariate data.

This document starts with an overview of network monitoring data, and the transformation from the original research topic into the more generic research topic: "*Visualization of multivariate data*". Furthermore existing visualization techniques that could be used for visualizing multivariate data are discussed in that chapter. In Chapter 3 requirements are given that a visualization technique must fulfill to address the research question. The discussed visualization techniques are compared to the given requirements and two visualization techniques are chosen that are used as basis for a solution to the research topic. These two techniques, Parallel Coordinate Plots and Scatter Plots, are then used in a network monitoring data example to test their capabilities to visualize multivariate data. Some problems that occur while using those visualization techniques are discussed and in Chapter 4 an idea is proposed and elaborated upon to overcome those problems. The idea is to make use of "Flexible Linked Axes".

In Chapter 5 the visualization interface is discussed of a tool, capable of displaying the suggested Flexible Linked Axes idea, called FLINAview. Here the operations, concepts and metaphors from FLINAview are discussed. Furthermore some problems that occurred while implementing the tool are discussed and solutions are presented.

Chapter 6 gives some examples using FLINAview on some well known datasets. These examples were shown to the users participating in a user study, to investigate both the concept of Flexible Linked Axes and the tool. The results from the user study are given in Chapter 7, as well as an overall evaluation. The last chapter gives suggestions for future work. The idea of Flexible Linked Axes proved to have a big potential, but there are certain additions that can enhance the concept.

Chapter 2

Background

The initial research topic for this project was to develop new and powerful methods for the visualization of network monitoring data. Network monitoring datasets were provided that were collected with an application called nfdump[1]. The network monitoring data used in this research consists of information for connections between pairs of network devices. In this case the network devices are computers. The dataset can be considered as a table, consisting of rows and columns. Each row, also referred to as record, describes a connection.

2.1 Network data

Connections are made between a source host and a destination host. A host consists of a unique identifier denoted by two values: an *IP address* and a *port number*. The connections are made at a certain point in *time* for a particular *duration*. A certain *amount of data* is transferred, packed in a *number of packets*. Each data record consists of eight fields, which are italicized. Columns from the dataset, representing the data values for all records for a certain record type, are referred to as attributes or dimensions. The value at cell $A_{i,j}$ from the table, having *P* columns and *R* rows, represents the value for attribute *j* at row *i*, where $i \in \{1..R\} \wedge \wedge\ j \in \{1..P\}$.

The ranges for the network monitoring data attributes are large. Table 1 shows the attributes with their possible ranges. The test range denotes the attribute range for a given dataset consisting of one week of network monitoring data.

Attribute	Range possibilities	Test range
IP address IPv4 (IPv6)	4,294,967,296 ($3.4 * 10^{38}$)	4,294,967,296 ($3.4 * 10^{38}$)
Port number	65536	65536
Time (milliseconds)	$1.8869 * 10^{12}$ (60 year)	604,800,000 (1 week)
Duration (milliseconds)	$6.2899 * 10^{10}$ (2 year)	604,800,000 (1 week)
Amount of data (bytes)	$6.7537 * 10^{18}$	1,073,741,824 (1 GB)
Number of Packets.	$1.1725 * 10^{16}$	715828

Table 1: Network monitoring attributes and their ranges.

Network monitoring

Network monitoring can be used for a variety of reasons. A network administrator at a hosting company is interested in the source hosts connecting to their network for malicious activity. This user will be interested in connection attempts to non standard ports and large amounts of connection attempts from source hosts. A network administrator within a company will be interested in data transfers from their computers, to find out whether there are security risks or users performing illegal activity. Internet service providers (ISP) network administrators are interested in the amount of data flowing through their network, making sure there is enough capacity and finding outliers concerning data amount.

[1] http://nfdump.sourceforge.net/

These different needs have led to several network monitoring visualization tools, ranging from two-dimensional graphs to higher dimension visualizations. A previous network visualization literature study [9] showed that existing solutions aim at displaying characteristics for a particular problem. The visualization techniques lack the ability to investigate all possible combinations of the data, but instead focus on a fixed set of data attributes. Although the techniques are good at visualizing the particular problem, thorough network monitoring requires multiple tools for investigating all possible malicious activities that might occur in the network.

Algorithms

Algorithms have been developed to aid the user in monitoring malicious network activities. The algorithms are static and do not provide the flexibility to adjust to rapidly changing network data characteristics. Intruder detection systems are systems implementing algorithms to aid the user in the network monitoring task. The algorithms have the tendency to generate false positives or do not detect certain malicious activities, referred to as false negatives. The problem with algorithms is that network monitoring specialists should be able to immediately react on new network related activities, hence not having the time to wait for algorithms to detect and identify the new types of malicious behavior. Although algorithms are useful in monitoring a network, the network administrator cannot solely rely on them.

Generic

The examples given of network monitoring users interests show that each of the attributes is considered important by some of them, although it rarely occurs that one user is interested in all attributes simultaneously. In general, while investigating the dataset, users are interested in a subset of attributes that might change over time. Whenever they have pinpointed a possible problem that required more investigation, they often required the visualization of other attributes that were not yet shown.

This characteristic gave rise to the idea to aim at a flexible visualization method, capable of showing an image with multiple attributes and which can easily be transformed into another image showing a different set of attributes.

The network monitoring dataset can be seen as a multivariate dataset consisting of eight attributes. There is only one particular attribute type specific for network monitoring data: the *IP address* attribute. The types of the other attributes are standard data types. However, the IP address can also be transformed to a more generic datatype. An IPv4 (Internet Protocol version 4) address consist of 4 bytes of data referred to as octets, separated by the dot symbol. The IPv4 address is encoded as:

$$X_1.X_2.X_3.X_4, \text{ where } \forall i: 1 \leq i \leq 4: 0 \leq X_i \leq 255$$

The IP address can be encoded into an integer in a straightforward way, and an integer can be encoded back to an IP address. Having the possibility to encode every attribute into a primitive data type, the research question could be interpreted as the problem of visualizing multivariate data. Henceforth the adapted research topic is "*Visualization of multivariate data*" for which the initial research topic is only a problem instance.

2.2 Visualization techniques

Transforming the research question into a more generic version provides a more abstract view on the problem. Visualization of multivariate data is an area where much research has already been conducted. Existing visualization techniques for multivariate data all share one common ground: displaying higher dimensional information using a lower dimensional space.

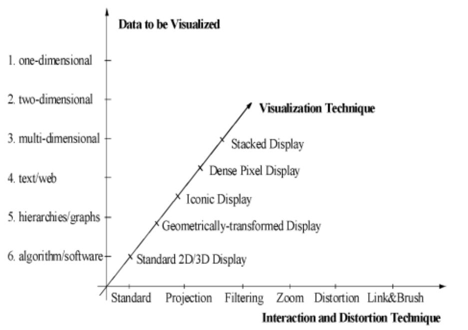

Figure 1: Classification overview from Keim [6].

Keim [6] reasons that visualizations can be classified by three categories: '*the data to be visualized*', '*visualization technique*' and the '*interaction and distortion technique*'. Figure 1 gives an overview of his classification, where the axes represent the categories. The categories are further classified into subcategories. The categories are independent, meaning that every combination of the subcategories is possible.

The subcategories from the 'visualization technique' category are briefly discussed and shown by an example. The 'Geometrically transformed display' category is subdivided, since it consists of multiple interesting and unique visualization techniques worthwhile to discuss individually.

Standard two-/three-dimensional Displays

Standard two-/three-dimensional techniques, also known as business graphics, are the most commonly used visualization techniques. These include x-y or x-y-z plots, bar charts, line graphs, pie charts, etc. Many computer applications, such as spreadsheets, provide elementary two-/three-dimensional visualization options. Figure 2 shows an example of a two-dimensional visualization, displaying the amount of network data transferred over time.

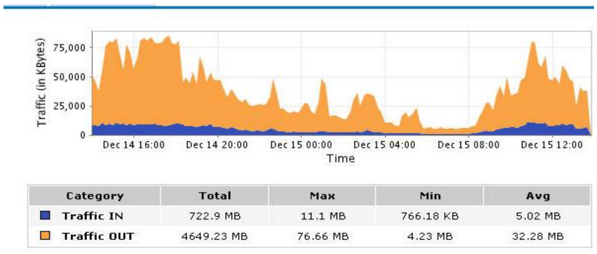

Figure 2: two-dimensional network monitoring visualization example from: http://ww.manageengine.com/products/netflow/.

Scatter Plot Matrices

A scatter plot is a point projection of data values in a 2-dimensional space. The point is positioned at the coordinate representing the value at both of the axes. An axis encodes one particular dimension. Scatter plot matrices consist of multiple scatter plots, combined in a larger view, each displaying a unique combination of two attributes.

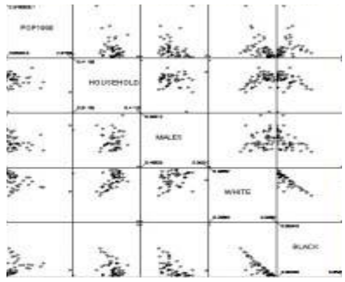

Figure 3: Scatter plot matrices [33].

Parallel Coordinate Plots

Parallel Coordinate Plots (PCP) [1][15] map n-dimensional space onto a two-dimensional space. This is done by using n equidistant parallel axes, where each axis denotes one of the dimensions. The axes are linearly scaled dependent on the range of the particular dimension. The data records are presented as polylines, intersecting each of the axes at the point corresponding to the value of the dimension belonging to the axis, for that record.

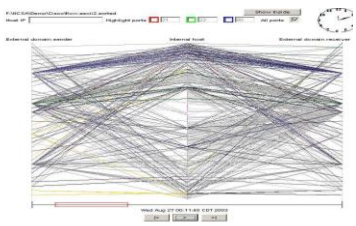

Figure 4: Parallel coordinate plot [2].

Radar Chart

A radar chart [16] uses a different arrangement for the axes. The radar chart arranges axes in a uniform way around one single point, referred to as a starlike pattern. Each axis represents a dimension. For each record the values on the axes is plotted. These are connected with their closest neighbors, resulting in a polygon that represents the record. It can be seen as a circular axes layout of a Parallel Coordinate Plot.

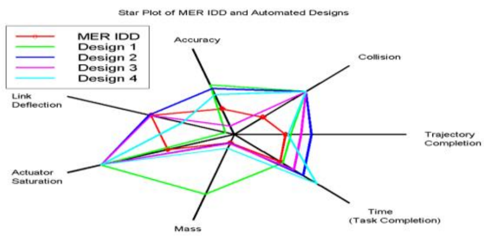

Figure 5: Radar chart example http://start1.jpl.nasa.gov/caseStudies/autoTool.cfm.

Star Coordinates

Star coordinates [13] also arrange axes in a starlike pattern, where the axes initially all have the same length and angle. Each record is transformed into a single point representing the sum of the individual axes values. Figure 6 gives an example how the point is calculated. The thicker lines radiating from the center denote the record values for each attribute along its axis. Those lines are considered as vectors and are summed, the lesser thick line from the center to point P, resulting into point P being the single point projection for the record. Multiple data records can lead to projections on the same point, hence this representation can be ambiguous.

By changing the angle and length of an axis the user can give more weight to a certain attribute to be able to better distinguish the individual values per attribute for a certain point. Figure 7 shows an example of a star coordinates plot where the axes are non uniform.

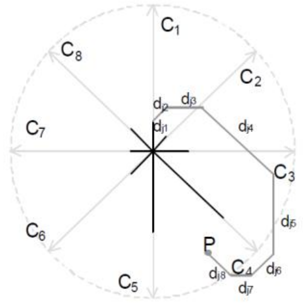

Figure 6: Star coordinates [13] coordinates explanation.

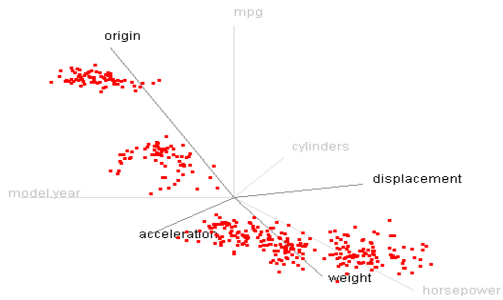

Figure 7: Star coordinates example [13].

Iconic Displays

Iconic displays use icons to map attribute values of multi-dimensional data. All kinds of icons can be used such as faces, star icons, stick Figures [30] or color icons. The data values are mapped to features of the icon. For instance a stick Figure can be used where the length and angle of the limbs denote the values for a certain dimension. Figure 8 shows an example of an iconic display using stick figures. The characteristics of the icons represent the data values, from where differences and similarities can be detected between the icons.

Dense Pixel Displays

Dense pixel techniques [29] map each attribute value to a color coded pixel, grouping them into adjacent areas for each attribute. The main issue is the arrangement of the pixels. Recursive pattern techniques and circle segments techniques are commonly used pixel arrangement techniques. Figure 9 shows an example of a dense pixel display, where concentric circles from the centre of the circle represent years and the circle is divided in areas using fixed degrees to denote the attributes.

Stacked Displays

Stacked displays show data partitioned in a hierarchical fashion. They basically show one coordinate system within the other. The partitions and hierarchies have to be selected in case of multi-dimensional data. In a two-dimensional layout the outer coordinates can be used to visualize the first two attributes, hence dividing the area into smaller areas. The next two other attributes are used to visualize data within smaller areas, reducing the area sizes again, but representing more attributes. This idea can continue a number of iterations. Examples of stacked displays techniques are Cone Trees [27], Worlds-within-Worlds [26] and Treemaps [28]. Figure 10 gives an example of a stacked display image.

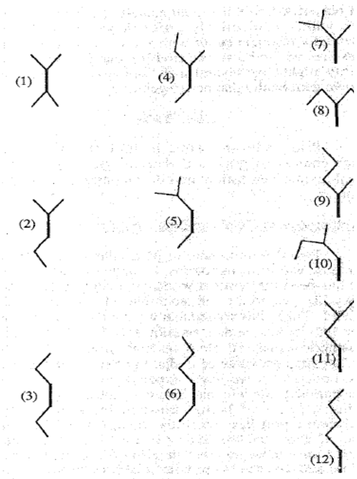

Figure 8: Iconic display example [30].

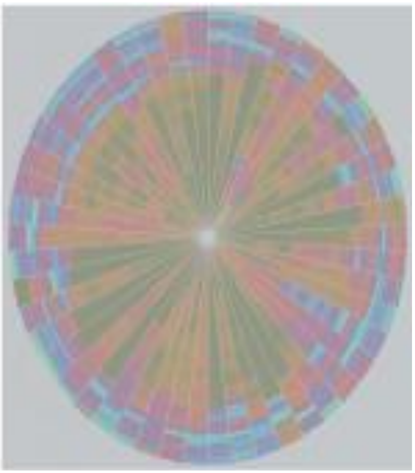

Figure 9: Dense pixel display example using circle segment technique [29].

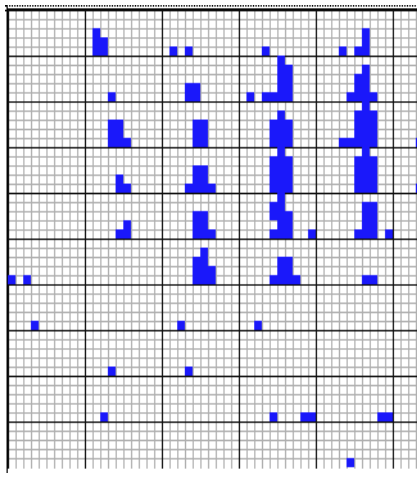

Figure 10: Stacked display example [6].

Chapter 3

Requirements and analysis

We argue that a useful visualization technique for multivariate data needs to fulfill a number of requirements. In this chapter such requirements, numbered as R# where # is a number, are given. The requirements are explained, discussed and are used to evaluate the previously given visualization techniques.

R1: Single view
There exist many visualization techniques targeting parts of a specific problem. Investigating different parts and characteristics of a dataset often requires the combination of different visualization techniques. The user has to be able to interpret those visualization techniques, hence requiring much effort to get familiar with them. Tools able of visualizing different characteristics of a dataset often split the screen into subparts or windows, each displaying a different visualization technique. The interaction between the visualization techniques is often limited, if available at all. The 'single view' requirement is to have one single view in which all required information is presented simultaneously in a generic way.

R2: Attributes
Multivariate data consist of multiple attributes. The user working with the dataset has domain knowledge and might be interested in all attributes or only a subset of the attributes. The user should be enabled to select the attributes of interest and increase or decrease the number of visualized attributes. Changing the number of selected and visualized attributes should not have an effect on the interpretability. Understanding visualizations with two attributes should be as straightforward as visualizations with ten attributes.

R3: Generic
The visualization should be completely generic, hence not dataset specific. It should provide the option to load any multivariate dataset and visualize that dataset.

R4: Value distribution
Attributes have a value range, defined by the minimum and maximum available value within the specific attribute. The distribution of attribute values should be clearly represented within the visualization.

R5: Correlation
Every attribute has a certain distribution, but there might also be correlations between attributes. The correlation between attributes should be depicted in a clear way.

R6: Unambiguous value
The data values should be represented unambiguously. They should clearly represent the attribute value at a certain point. An example of an ambiguous data representation can be seen in Figure 11. Here a three-dimensional object is represented on a two-dimensional display. The three-dimensional points (5,5,0) and

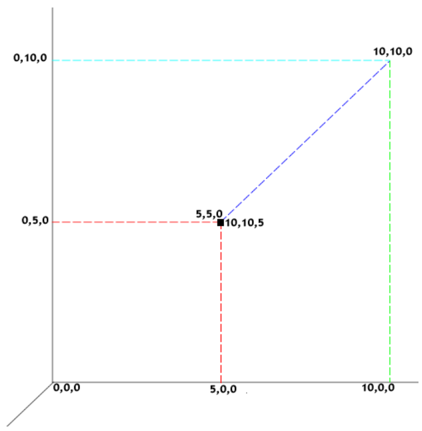

Figure 11: Ambiguous values example.

(10,10,5) are displayed on the same pixel position in the two-dimensional plane. This should be avoided since this leads to `*data hiding*`. The user is, without additional interaction, unable to see the different values, represented by the coordinates. Preferably the user does not require interaction to cope with ambiguous values. Since all visualizations eventually are mapped back to a two-dimensional space, this implicitly gives rise to the requirement of a two-dimensional visualization.

R7: Unambiguous occurrences

Since a dataset can consist of many items, identical values might occur within the dataset, or there exists values for which the difference between them might be very small. Those values are displayed on the same position which also leads to data hiding. The user should be enabled to detect multiple occurrences for a single value or for a small range of data values, all positioned at the same point in space.

R8: Easy to understand

When users are confronted with an image they should be enabled to quickly understand what is visualized and how it is visualized. Users should not need many hours of research and training before they are able to understand the image.

R9: Fast usability

Users should be enabled to quickly understand the data visualized in an image and to pinpoint outliers and other parts of the image that require more investigation.

R10: Correct interpretation

Correct interpretation of a visualization is important since the user might otherwise draw wrong conclusions from the data. Having a visualization that the can easily and quickly understand does not necessarily imply that it is correctly interpreted.

The previous requirements concerned visual presentation, but there are also requirements concerning the interaction. Since the amount of data that should be visualized is large, it might not be possible to show all information at once. The visualization should take Shneiderman's mantra [22] into account: "*Overview* first, *zoom and filter*, then *details-on-demand*".

R11: Overview

The user should be provided with overviews of the data. The overview can form the starting point from where the user is enabled to pinpoint problem areas requiring more investigation. The overviews should be clear and capable of showing all data while still being able to see data characteristics.

R12: Zoom and filter

The user must be enabled to zoom and filter on interesting areas to retrieve more or less detail. Zooming can be used to increase the level of detail for a particular part of the image by enlarging it, whereas filtering can be used to get rid of uninteresting information. This way the user is enabled to narrow the dataset and investigate smaller and more interesting parts of the dataset.

R13: Details on demand

After zooming and filtering the dataset, the user should be enabled to retrieve details for the resulting subset of data, hence filtered data values should be easily browseable and exportable.

3.1 Visualization techniques analysis

In Section 2.2 visualization techniques are discussed and we have given requirements on multivariate data visualization in this chapter. Next the visualization techniques are evaluated against the requirements, to find visualization techniques most appropriate for the research topic. Table 2 gives an overview of the results. The last three requirements are not used in the evaluation because they are '*tool specific requirements*' and not '*visualization technique specific requirements*'. Every visualization technique can potentially support the last three requirements, if they are properly implemented in a tool. The last column gives a visualization technique score, the sum of the '+' signs in the row minus the number of '-' signs in the row.

	R1: Single view	R2: Attributes	R3: Generic	R4: Value distribution	R5: Correlation	R6: Unambiguous value	R7: Unambiguous occurrences	R8: Easy to understand	R9: Fast usability	R10: Correct interpretation	Visualization technique Score
Standard 2D/3D Displays	□	-	+	+	+	+	□	++	++	+	**8**
Scatter Plot Matrices	□	+	+	+	+	+	□	+	+	+	**8**
Parallel Coordinate Plots	□	+	+	+	+	+	□	+	+	+	**8**
Star Coordinates	++	+	+	-	□	--	--	-	-	--	**-3**
Radar Chart	□	□	+	+	+	+	□	□	□	□	**4**
Iconic displays	+	□	+	-	-	□	□	-	-	□	**-2**
Dense Pixel Displays	++	+	+	□	--	++	□	--	+	-	**2**
Stacked Displays	+	□	□	+	-	+	+	□	+	□	**4**

Legend	
++	Good
+	Above average
□	Neutral/average
-	Below average
--	Bad

Table 2: Evaluation of requirements per visualization technique.

As can be seen from the table, there are three visualization techniques that perform better than the others. Since this research focuses on multivariate data, the `Standard 2D/3D displays` visualization technique will not be investigated, hence leaving the `Scatter Plot Matrices` and `Parallel Coordinate Plot` for further investigation. These are now more extensively investigated to find out their strong points and their shortcomings, with an emphasis on dynamic exploration.

3.2 User interest

We argue that a user investigating multivariate data is primarily interested in two things:

A) Attribute value distributions: *attributes;*
B) Pairwise relations between attribute value distributions: *attribute relations.*

These two aspects are the base elements for the user's interest. The user can be interested in combinations of those elements. To reason about this, we can present the current interest of a user as a graph, where nodes represent attributes and edges represent the attribute relations between the nodes.

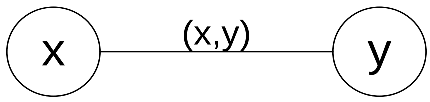

Figure 12: ARGOI example. x and y denote attributes and the edge (x,y) denotes the attribute relation between x and y.

The current set of attributes and attribute relations the user is interested in is called the "Attribute Relation Graph Of Interest" or ARGOI. Note that an ARGOI is dynamic and there are many possible ways to represent the data of an ARGOI. In the following cases we show how a user explores a data set, using different representations, and meanwhile we show how his ARGOI builds up and changes. The dataset used is a SSH networking monitoring dataset consisting of three hours of SSH data, consisting of 93,384 rows and thirteen attributes, so in total over 1.2 million data values. SSH is a network protocol that allows for secure data exchange between pairs of network devices, mainly used for remote administration of Linux and Unix based systems.

3.3 Visualization practices

The dataset is being used to find attackers trying to gain access to the local network to perform malicious activities. Axes are depicted as lines with an arrowhead, where the latter indicated the direction of the axis.

Base case

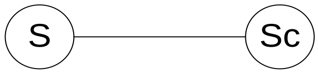

Figure 13: ARGOI for base case

We start with investigating a base case, that is the relation between two attributes. The first relation that is visualized is the relation between the '*Source IP Address*' (S) and the '*Total number of occurrences for the Source IP*' (Sc) in the dataset. This gives information about the most active hosts in the dataset. Figure 13 shows the ARGOI for this case, whereas Figure 14 and Figure 15 show the ARGOI visualized as a scatter plot and as a PCP respectively. The lines or points between the two axes represent the data values and denote the attribute correlation. Each individual item is called a livo, which will be described and discussed in detail in Chapter 5.

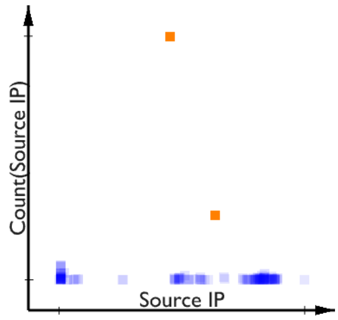

Figure 14: Base case visualized as a scatter plot.

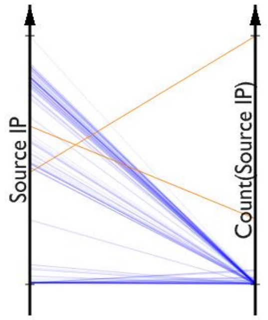

Figure 15: Base case visualized as PCP.

It can be clearly seen that most hosts appear only a limited number of times, but there seem to be two hosts, marked in orange, standing out. The images are reasonably clear and can be easily understood. Note that the scatter plot livo's are enlarged to make them clearly visible, while making sure no information gets hidden.

Case: Two links

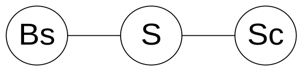

Figure 16: ARGOI for two links.

The previous case shows something interesting that requires more investigation. Large amounts of connections could point to malicious activities, or normal connections that transferred large amounts of information. To make a distinction between those cases additional information is required, and we display the '*total number of transferred bytes*' (Bs) per Source IP next to the current information. The resulting ARGOI is shown in Figure 16, whereas the representations using the visualization techniques can be found in Figure 17 and Figure 18.

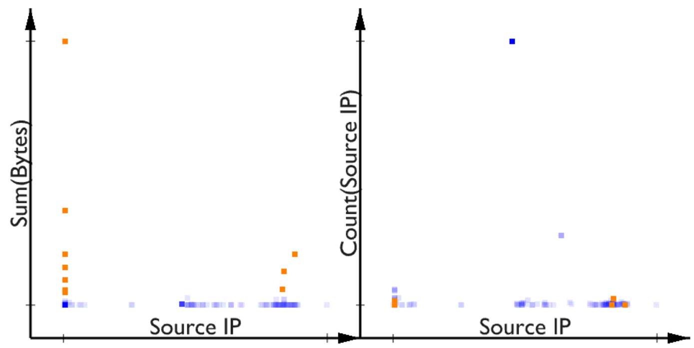

Figure 17: Two links displayed as scatter plot.

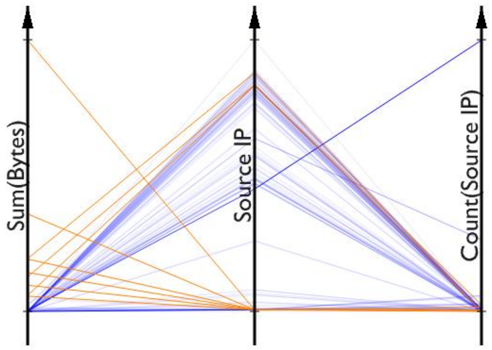

Figure 18: Two links displayed as PCP.

The marked items in those two figures are the items with a large amount of data being sent. It becomes clear that the hosts with a large number of connections do not transfer large amounts of data, as one would expect. So it could be that those hosts with a large amount of connections are trying to perform malicious activities, since trying to connect to a system does not transfer a large amount of data.

Case: Three links

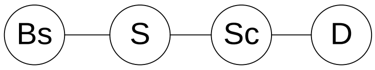

Figure 19: ARGOI for three links.

The two hosts connecting to many hosts without much data transfer are interesting, as they show characteristic behavior for port scans or port sweeps. Still no conclusions can be drawn without additional information about the destination. The next thing that is important is to show the '*destination IP address*'(D), resulting in the ARGOI from Figure 19. Unfortunately the destination IP does not provide us with useful information as can be seen in Figure 20 and Figure 21. We can only see that the interesting hosts connect to computers in the local network, the bottom part of the Destination IP, but more details are required to draw any conclusion.

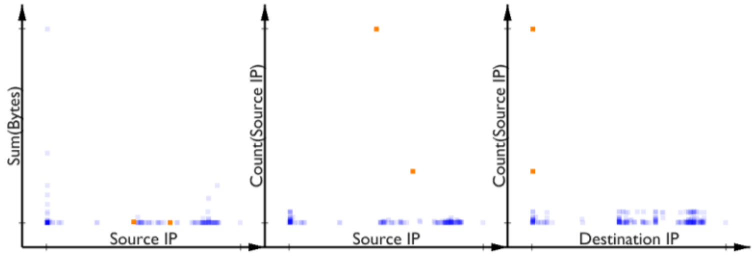

Figure 20: Three links visualized as scatter plot matrix.

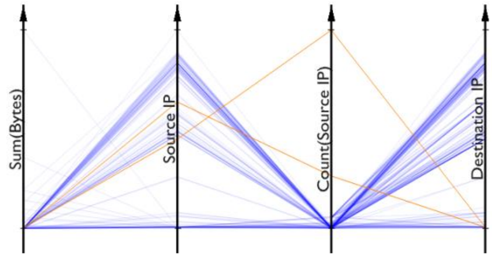

Figure 21: Three links visualized as PCP.

Case: Three-way connection

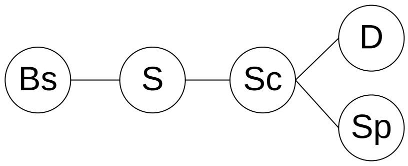

Figure 22: ARGOI for three-way connection.

Since the previous example did not give additional useful information, we still have to investigate what is going on with the interesting hosts. To see if the host is performing a port scan or port sweep additional information is required. We are first interested in the source port the selected host is using. If this turns out to be the same port every time, this might indicate one connection that might have been open for a long time. Figure 23 and Figure 24 show the results from this new ARGOI, displayed in Figure 22. In the figures we focus on the host with the most connections.

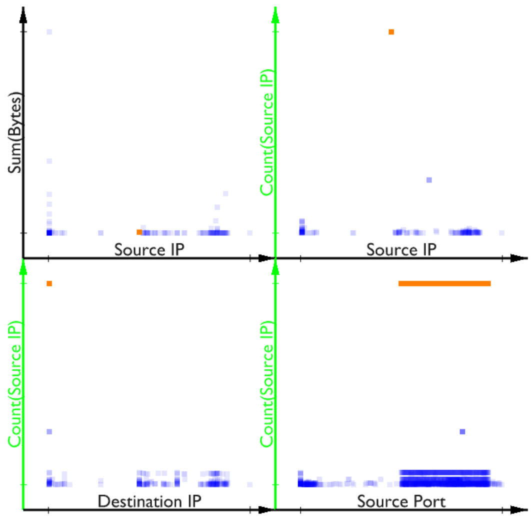

Figure 23: Three-way connection visualized as Scatter plot matrix.

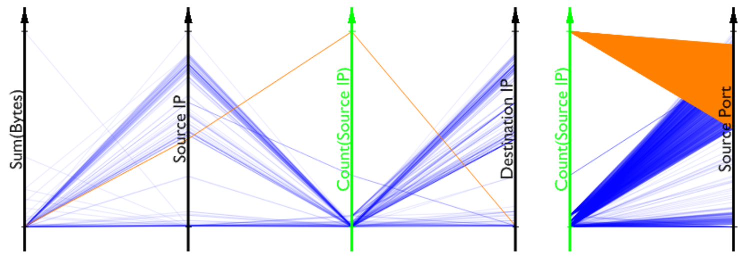

Figure 24: Three-way connection visualized as Parallel Coordinate Plot.

The new attribute shows that the source ports, where the connections originate from, seem to be continuous. The host seems to be creating new connections to hosts, while each time the source port used for the connection is incremented. This is default behavior for port scans and port sweeps. It becomes more likely that we are dealing with a hack attempt from the selected host.

The problem that occurs with the images however, is that they are getting harder to interpret. It takes more time to see all the information embedded within the image, especially since users have to change their point-of-view to see all information. The important attribute within the last two figures: 'Count(Source IP)', is located at several places, hence the user is required to switch their viewpoint between those places to see all the characteristics for that attribute.

3.4 Other connections

The ARGOI from the three-way connection is not the most complicated ARGOI that can be imagined. There are many other examples resulting in more complex ARGOIs, leading to larger problems when visualizing with the two discussed visualization techniques. A couple of them are displayed below:

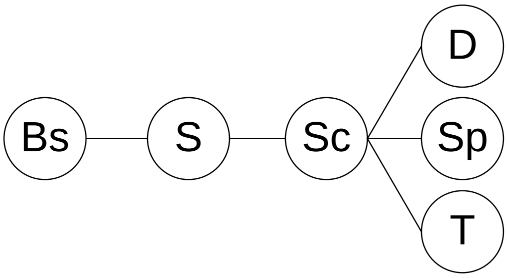

Figure 25: ARGOI for four-way connection.

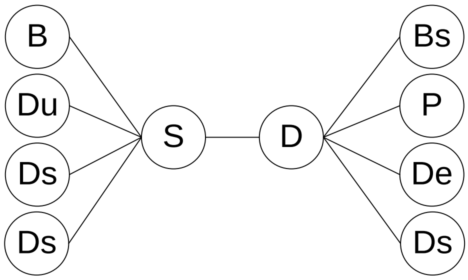

Figure 26: ARGOI for combination.

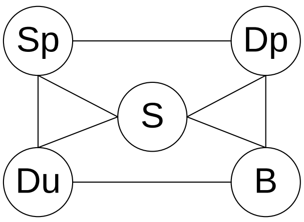

Figure 27: ARGOI for focus.

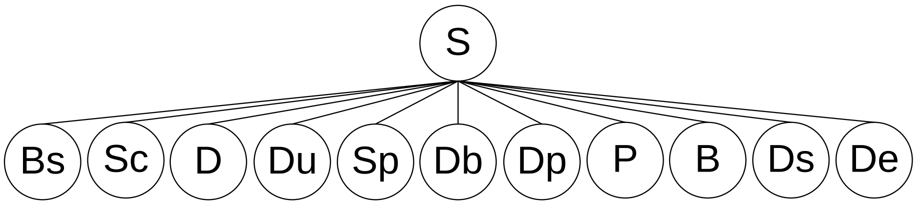

Figure 28: ARGOI for fan out.

In Section 4.2 examples of use visualizations for these ARGOIs are given. Scatter plot matrices have the property to display all relations independently. This makes larger paths of attribute relations harder to understand, for instance: A→B→C→D.

Parallel coordinate plots fall short due to the fact that they are only capable of correctly showing linear sequences of relations, whereas every node can be connected to at most two other nodes. This requires the reintroduction of a node to be able to link them to other attributes. Reintroduction of nodes gives rise to the same problem that scatter plots have; the need for a constantly changing point of view.

Other visualization techniques also fall short, for instance the radar chart. The radar chart has the same problems as the PCP, but it is possible to show one additional relation; the relation between the first and last node, hence only capable of showing cycles of attributes.

Chapter 4

Problem investigation

One of the major problems with the currently existing visualization techniques is that more complex patterns in the relations between different attributes, i.e., more complex ARGOIs, are hard to visualize. The key idea is now to enable the user to define a visualization that reflects the ARGOI, inspired by scatter plots, PCP's and radar charts.

4.1 Many-to-Many relational Parallel Coordinates Display

The many-to-many relational coordinates display [3] presents a different layout for displaying PCP's. The axes are more freely placed to support for grouping of attributes. Figure 29 shows all the relations between seven attributes for a particular dataset at the same time using this technique, whereas Figure 30 shows the same relations in a standard PCP layout. The axes in both Figures are color encoded and labeled, to show clearly which axis denotes the same attribute. The example given in [3] corresponds to an ARGOI with seven nodes, where the layout is identical to that of Figure 29, whereas also an idea for a layout with four nodes is introduced.

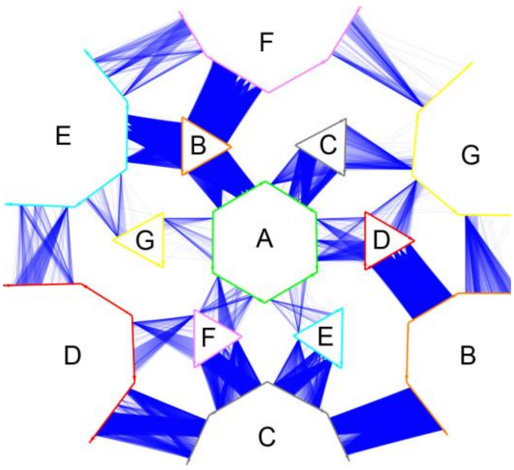

Figure 29: Many-to-Many relation Parallel Coordinates Display.

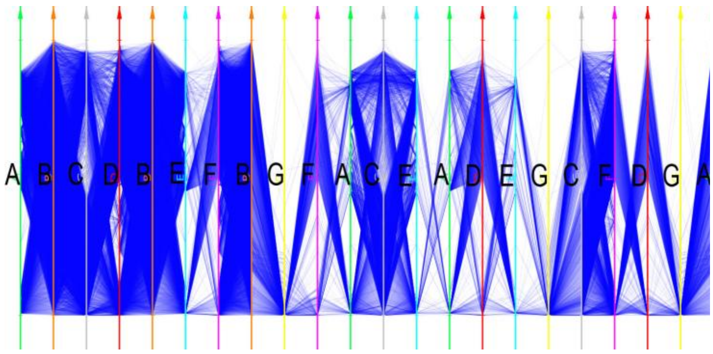

Figure 30: PCP showing all connections between 7 attributes. Showing the same information as Figure 29.

Attributes, except for the central attribute, still have the property of being replicated to different locations within the image, to show all relations. Lind et al. [3] have chosen a layout where similar attributes are placed at opposite sides of the central attribute making the orientation easier for the user. The result with seven attributes was derived from a four attributes example. They did not investigate possibilities for dealing with other numbers of attributes. The user study that Lind et al. conducted, comparing standard PCP and their method, focused on subjects having to identify negatively correlated attributes, while having only positive and negative correlations. The amount of errors was evenly distributed between both layouts, but the subjects performed 20% faster in the Many-to-Many layout.

4.2 Flexible Linked Axes

The shortcomings for scatter plots and PCPs, as shown in the previous chapter, combined with the many-to-many relational parallel coordinates display brought up the idea of Flexible Linked Axes. Providing the ability to place axes in any position and direction gives more flexibility. This flexibility leads to more possible visualizations for complex ARGOI's, while still keeping the strengths of the original visualization techniques. We now present the idea with examples increasing in complexity, hence along the way the images become more powerful and useful.

Three-way connection

We refer back to the three-way connection from the previous chapter to introduce FLINAview, the prototype tool implemented to support the idea of Flexible Linked Axes. The images, making use of the Flexible Linked Axes idea, created with FLINAview, are called FLINAplots. Using FLINAview we show the capabilities of Flexible Linked Axes to visualize the data for ARGOIs. The problem with original the visualizations techniques was that one axis had to be connected to three other axes, hence leading to the reintroduction of the axis, since it could only be connected to at most two axes. FLINAview removes this restriction, since axes do not have to be parallel or at equidistant space. Figure 31 and Figure 32 show two examples for the data of the ARGOI displayed with Flexible Linked Axes. The first figure shows the ability to connect an axis with more than two axes, the second figure shows a layout for displaying the data from the ARGOI by a more flexible arrangement of the axes. The colored axes are the axes showing functionality that the Flexible Linked Axes idea provides in addition to the original visualization technique.

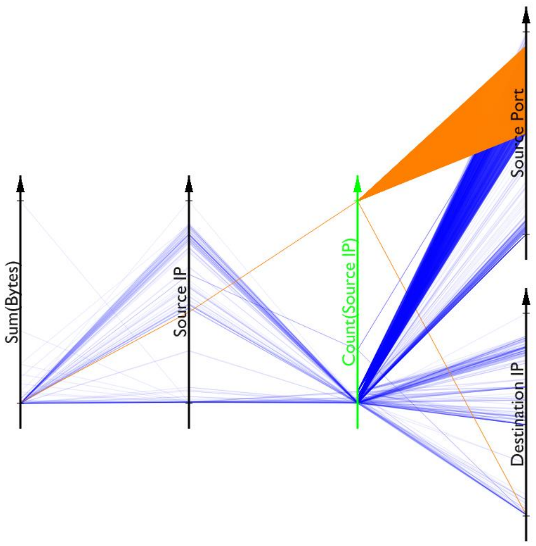

Figure 31: FLINAplot for Three-way connection, showing axes connecting to more than two other axes.

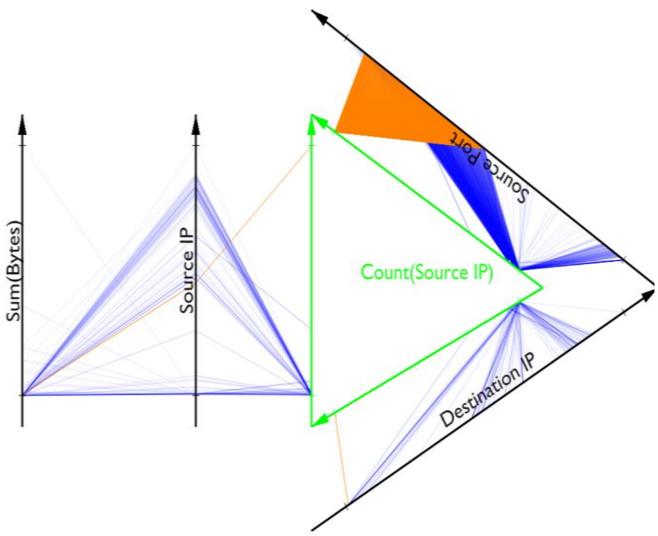

Figure 32: FLINAplot for Three-way connection.

Four-way connection

The previous case with the three-way connection showed that we might have found a host that tries to perform malicious activities. Additional information might make it clearer whether this is actually the case, hence another attribute is added. We want to know when the connections were attempted, hence adding a time attribute: *DateFlowStartTime* (T). The resulting ARGOI is shown in Figure 33 and the resulting FLINAplot in Figure 34. Note that in Figure 34 the 'Destination IP' range is filtered, thus showing a smaller range of data, hence providing more detail.

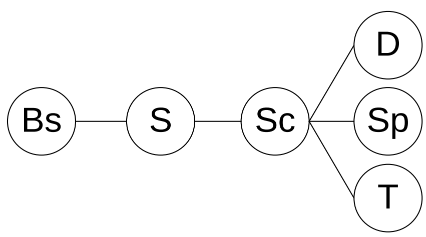

Figure 33: ARGOI for four-way connection.

Figure 34 shows that we have found an external host trying to perform malicious activities within the monitored network. It shows the host performing a port sweep, trying to connect to a default port on a range of hosts. From here on the network administrators are able to perform several actions. They can retrieve details about the attacker is, having IP address: 98.125.132.210 in this dataset, and block the attacker from accessing their network. They can investigate which computers within their network are compromised, these are the computers to which the hacker has been able to send data other than the default connection information data, see Figure 35. The administrators are enabled to further investigate these probably compromised hosts. This can be done by investigating which other external hosts made connection attempts to compromise them, or find the hosts these compromised computers connected to after the attack, to investigate whether they might have compromised other hosts. FLINAview is capable of visualizing all these cases, hence aiding the network specialists with securing their network.

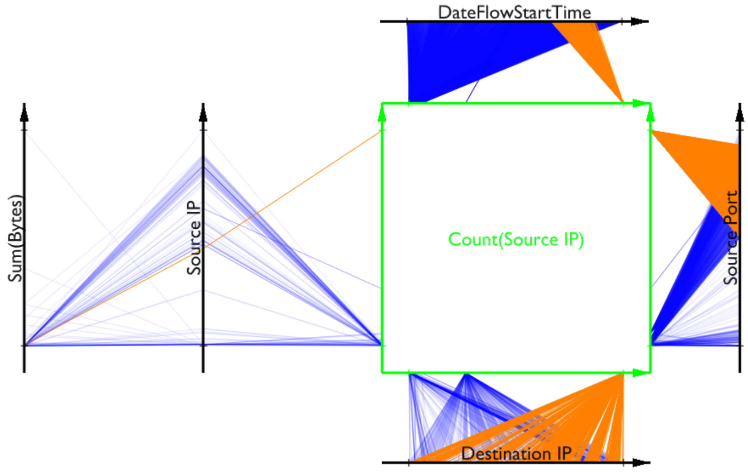

Figure 34: FLINAplot for Four-way connection, showing a malicious host performing a port sweep.

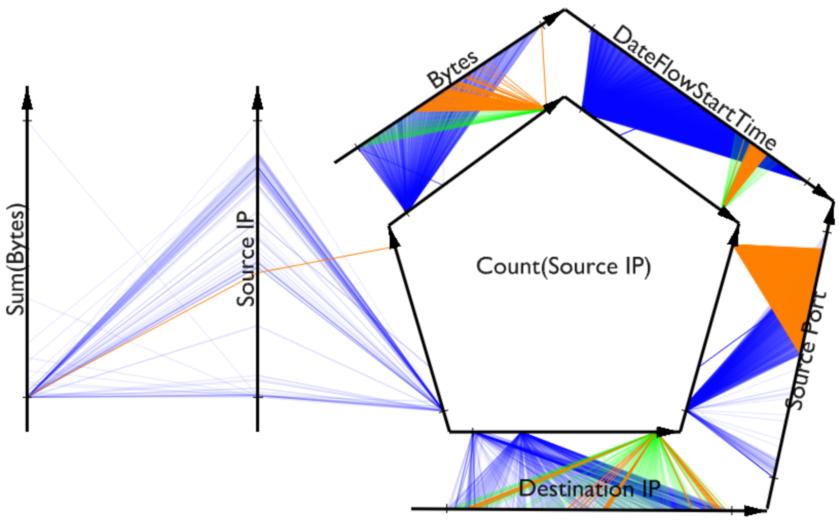

Figure 35: FLINAplot showing a malicious host performing a port sweep. The additional attribute 'Bytes' gives information about the success of the connection attempts. The green livo's denote unsuccessful connection attempts, the orange livo's show successful connection attempts. The attacker has made 28,163 connections within 19 minutes.

Fan out

Another, completely different view that the user might be interested in is the relation from one attribute to all other attributes. This is a so-called fan out. Figure 36 shows an example of an ARGOI for a fan out, linking one attribute to twelve attributes. Figure 37 shows an example of a FLINAplot showing the fan out. A fan out can be useful for investigation of one attribute in more detail, to focus on that attribute, and see the relation with the other attributes. Chapter 7 gives another example where a 'Fan out' can be used for.

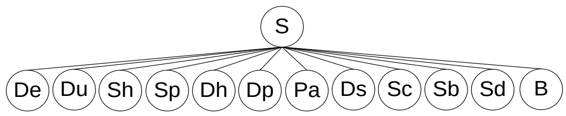

Figure 36: ARGOI for fan out.

Combination

So far the idea of Flexible Linked Axes was introduced by a step-by-step investigation of a dataset. The step-by-step approach gives an idea of how the idea of Flexible Linked Axes can be used, but does not show more complex layouts the user might be interested in.

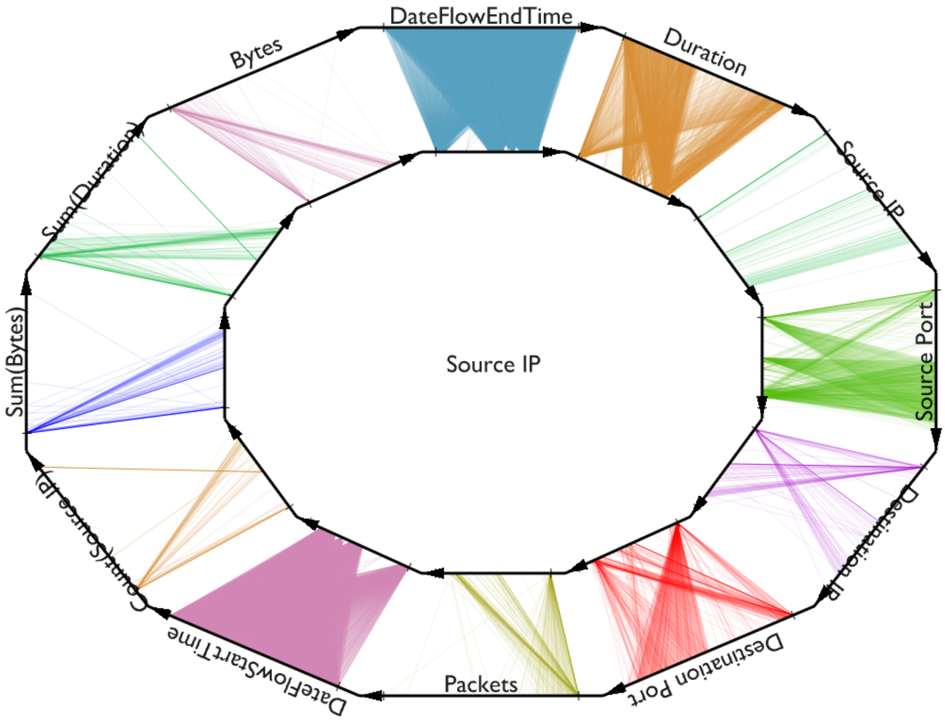

Figure 37: FLINAPlot of fan out, linking one attribute to twelve attributes.

Here several combined ARGOI's are given with a corresponding FLINAplot, as examples how Flexible Linked Axes can be used to visualize more complex user interests in datasets.

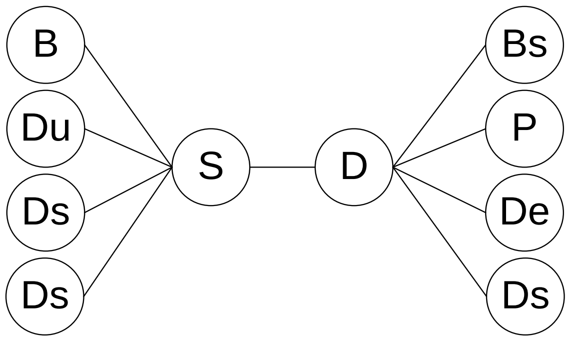

Figure 38: ARGOI for combination.

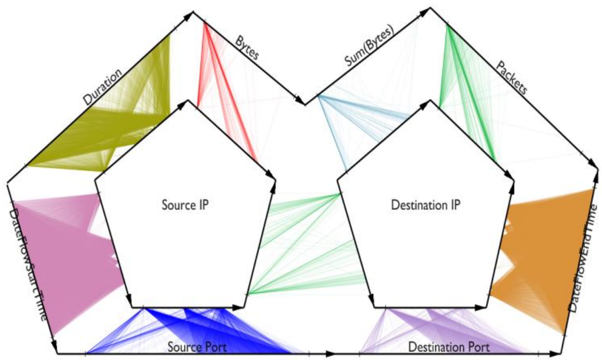

Figure 39: FLINAplot for combination ARGOI.

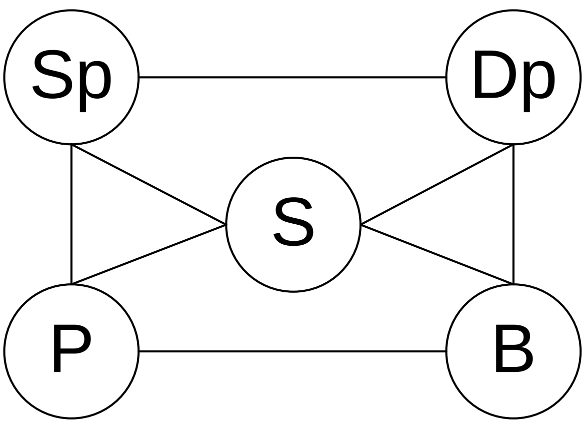

Figure 40: ARGOI for focus.

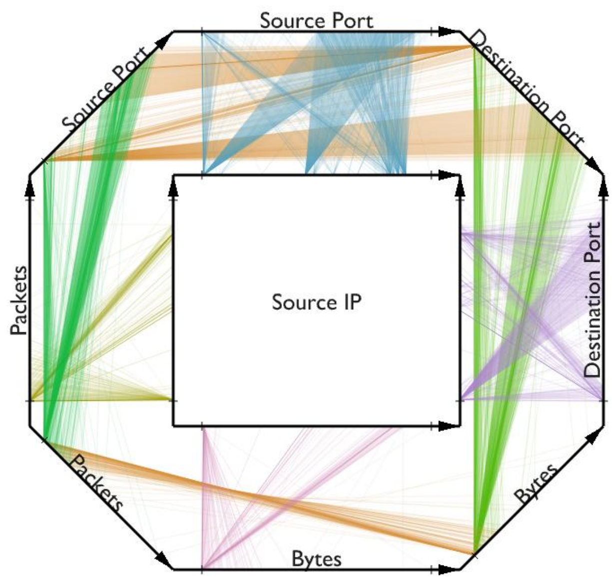

Figure 41: FLINAplot for focus ARGOI.

4.3 Scatter plot

Until now the Flexible Linked Axes concept has been introduced by examples of PCPs, whereas it was argued in Section 3.1 that scatter plots also have merits and should be supported, as they prove to be easier to understand and more clear in certain cases. So far we connected pairs of axes with lines to show data values, leading to generalized PCPs. We can also interpret pairs of axes as definitions of, possibly rotated and skewed, scatter plots, and show pairs of data values as points. Figure 42 gives an example where a scatter plot configuration provides more clear information in comparison to a PCP configuration.

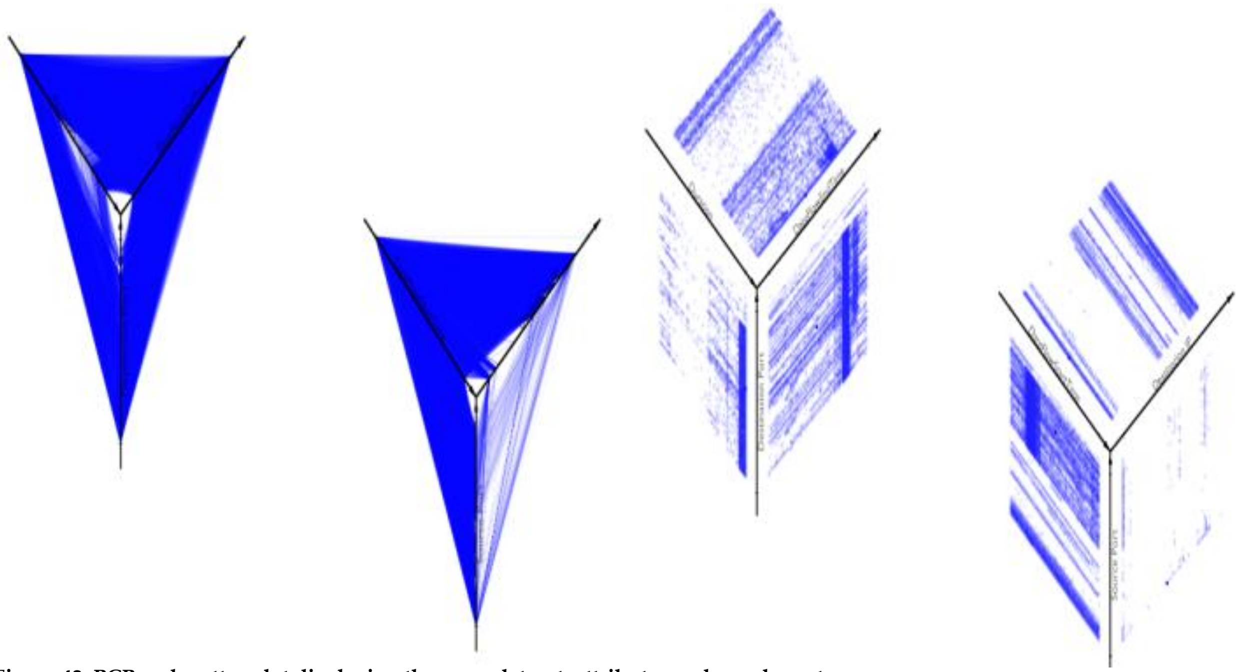

Figure 42: PCP and scatter plot displaying the same dataset, attributes and axes layout.

Chapter 5

Visualization interface

The previous chapters have shown several visualizations using Flexible Linked Axes. Note that an ARGOI can be visualized by multiple different FLINAplots. In this chapter the tool that was developed based on the idea of Flexible Linked Axes, FLINAview, is discussed.

FLINAview has several operations, concepts and metaphors to support the user to define and edit FLINAplots, for which the important ones are discussed here. A schematic overview of the system is displayed in Figure 43, showing the concepts and their relations. The numbers, next to the lines connecting the concepts, give information about the relation and the number of occurrences for a given concept within the relation. *N* denotes any possible number, whereas *P* and *R* denote the number of records and attributes respectively in the dataset. The relation between *Polygon* and *Point* (*1 - 2..N*) represents that each *Point* belongs to exactly one *Polygon*, but every *Polygon* consists of at least two points.

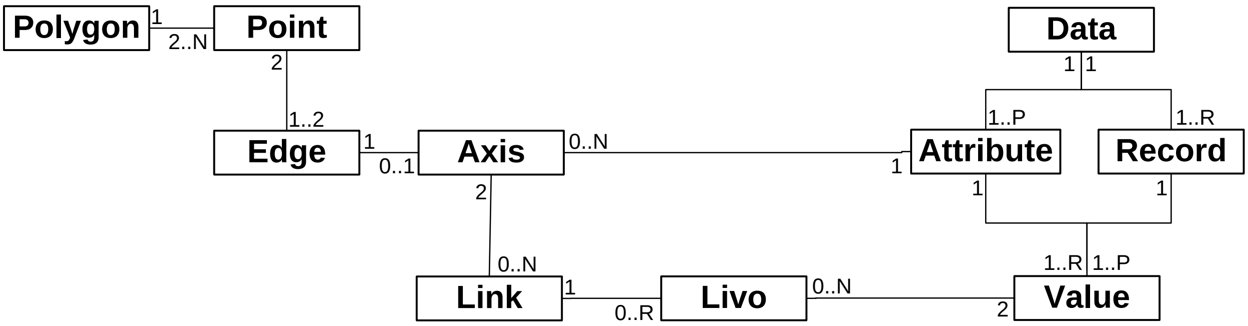

Figure 43: Schematic overview of the system structure and concepts.

Figure 44 shows an annotated interface for FLINAview, displaying the concepts and metaphors.

Sketch

To provide fast attribute relation investigation, the metaphor of a sketch interface is chosen. The interface supports quick creation and manipulation of axes, such that the user can sketch axes and easily specify which axes are linked. The sketch metaphor allows for real-time interaction with the visualization.

Canvas

The tool provides the ability to quickly create and manipulate the visualization, such that it enables the user to produce visualizations addressing the problem. The canvas is the area available for the image, hence the result from any manipulation operation is visualized on the canvas. The canvas provides a grid, gray points on top of the canvas where objects snap to, to aid the user in defining the position for objects drawn on the canvas.

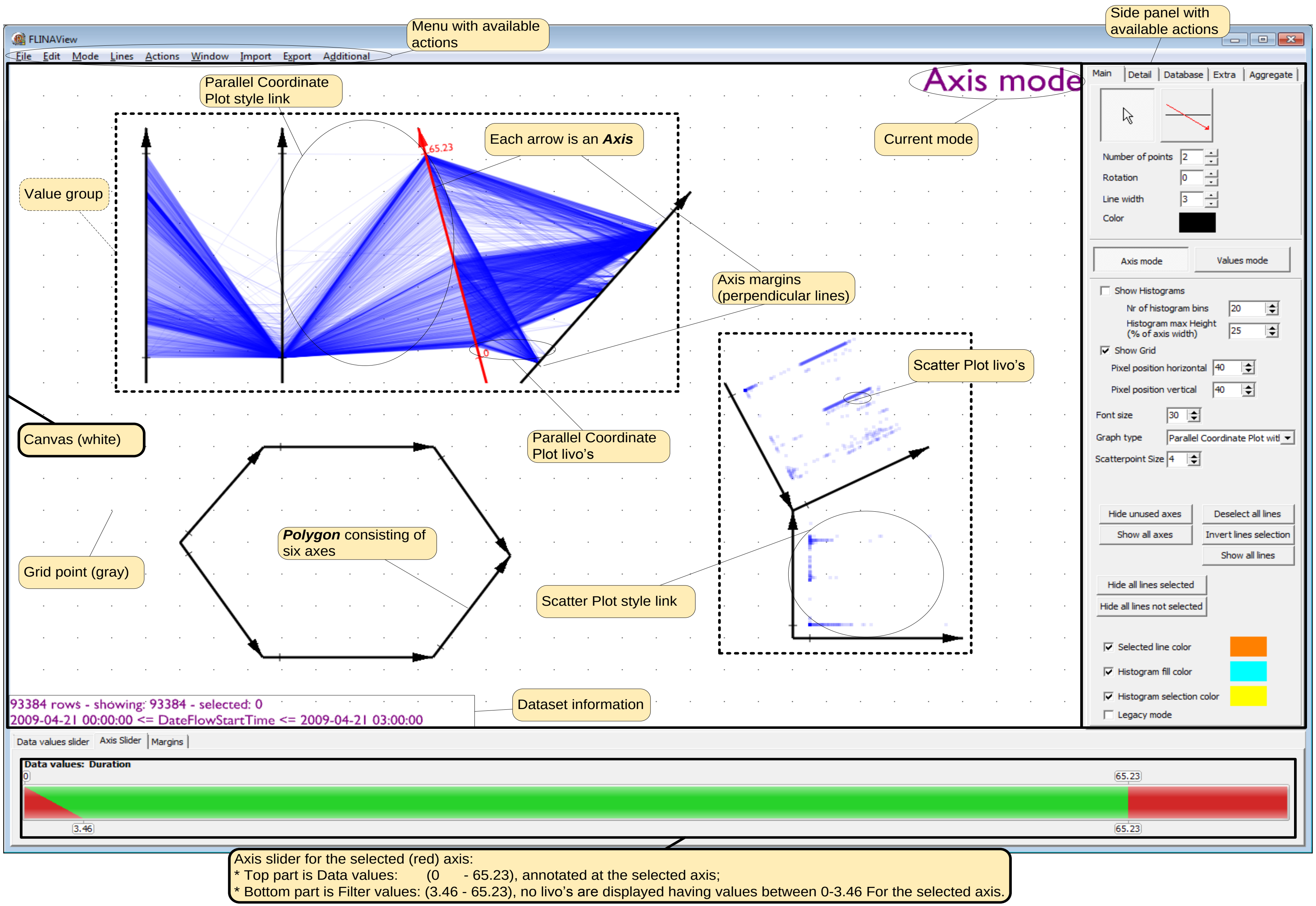

Figure 44: FLINAview overview showing the interface and the important concepts.

Axis

An *axis* is used for displaying one particular data attribute. The nodes from the ARGOI can be directly mapped onto an axis. Axes are defined by two points in the two-dimensional coordinate plane. The line between the points, the edge, denotes the target range for the specific data attribute. Axis *margins* define the part of an edge being used for data representation. The *data range* defines the lower and upper bound of the attribute values and can altered to allow for using sub ranges. Apart from the data range there is a *filter range*, being a sub range of the data range. Only records with attribute values within the filter range for the particular axis are shown. This allows for filtering within a particular data range, providing information about the part of the data range that is currently displayed. An axis by default is shown with a *label*, displaying the attribute name along the axis, for which the text can be modified or removed by the user.

Polygons

Polygons are combinations of axes. They can consist of an arbitrary number of axes. Polygons allow for quick selection and manipulation of multiple axes. For instance, the user is enabled to draw polygons with multiple sides, allowing for quick generation of standard layouts of axes. Polygons group axes, but each of the axes can be manipulated individually. The shape of the polygon is by default regular, to fit in the rectangle outlined by the user. Individual points from a polygon can be displaced, hence changing two axes simultaneously. If all axes of a polygon have the same attribute, one label is shown centered in the polygon.

Link

A *link* is used to display the correlation between the attributes of axes. A link can be defined between any two axes. An attribute relation from an ARGOI is mapped onto a link. Each link consists out of so called *link value objects*.

Link Value Object

The link value object (livo) represents the actual values and belongs to a link. Each individual livo represents the values from two attributes for one specific record. The values are encoded by a position along the edges from the axes that represents the value for the particular record and attribute. A livo can be a line between the values on the edges, or a point at the intersection of the perpendicular lines emerging from the points representing the attribute values for that record.

5.1 Operations

To support interactive inspection, with a dynamic ARGOI, the user is provided with a set of operations to adapt the visualization. For most operations the user is offered multiple ways to perform them: via a button, a pop-up menu or a keyboard shortcut. Undo and redo functionality is supported for most operations, to enable and invite the user to experiment.

Axis manipulation operations

The user is enabled to real-time manipulate the current visualization. Axes are important objects within the visualization and define most of the visualization, hence users should be enabled to alter an axis to their needs. Moving and resizing an axis are available operations for placing the axis at the appropriate position. The start and end point for an axis can individually be manipulated, leading to a flexible way of axis positioning.

Creating and removing an axis gives the option to adjust the visualization to show required or remove unnecessary attributes. Filter and zoom operations on the axis provide the ability to show the required subset of data values for a particular axis. Several other operations supported for axis manipulation are: coloring, naming, rotating, flipping and changing of the attribute.

Histograms

Histograms can be shown along an axis to show the value distribution. Histograms pack the value distribution into a number of bins, where for each bin a rectangle is shown which height represents the number of values within the range of the bin. The user is enabled to change the number of bins as well as the maximum height. Histograms representations are shown per axis, where the bins are linearly scaled accordingly to the bin with the largest amount of values. In Section 5.3 the histograms and their functionality are discussed in more detail.

Link manipulation operations

When two axes are selected, a link can be defined between them. If there already exists a link, that link can be removed. When a link has been defined several operations are available to manipulate the link and the livo's belonging to that link. The type of visualization technique can be changed, to display a scatter plot or PCP with flexible linked axes functionality. The link, and therefore all livo's belonging to that link, can be color coded. Since the flexibility enables the user to define overlapping livo's, link coloring enables to user to identify which livo's belong to what link.

Livo manipulation

The user is enabled to make selections of livo's. Whenever there is an active selection the user can perform operations on that selection. A selection can be created by drawing a box around, or a line intersecting the livo's the user wants to select, also known as brushing [34, 35]. Selections can be extended or reduced by selecting or unselecting livo's. The selection is a global selection, hence not the attribute values, but the records belonging to the attribute values are selected. Global record selection allows for emphasizing all visualized livo's for the selected records within the visualization. The active selection is highlighted in three ways; setting the line opacity to the maximum level, see Section 5.3; showing it in a different color; and drawing it on top of all other livo's. Removing and hiding are operations that can be performed on selected livo's. The data values are separated from the canvas objects, only links from the canvas objects are defined pointing to attributes, hence removing or hiding livo's removes or hides them from all defined links. When it is required to remove or hide livo's from individual axes, the data range and filter range should be used. Also the exact data values of the selected records can be shown in a separate table.

Global operations

The tool supports zooming and panning of the total visualization, where zooming is performed towards the selected two-dimensional coordinate the user is pointing at on the canvas.

5.2 Functionality

For quick interaction and additional flexibility additional functionality is available aiding the user or the system to define more clearly what is visualized or to which objects the operations apply.

Modes

Two separate modes have been defined within the interface; 'Axis' and 'Values'. The mode refers to the actions that can be performed. Users must be enabled to quickly select and manipulate parts of the image to their needs. If the visualization becomes dense, it should remain clear what the user wants to manipulate. When selecting a particular point or area it might be unclear what the user wants to select; a polygon, axis, livo or histogram bin. To overcome this problem a mode has been introduced. The mode defines what the user is currently working with. There are two modes defined: '*axis mode*' or '*values mode*', for selecting axes and polygons, or for livo's and histograms respectively.

Axes value group

The user should be enabled to investigate the attributes and their values in *isolation* or in a *joined* fashion. This difference becomes important when the user for instance draws a number of disjoint scatter plots. Now, if for instance records are filtered out from one scatter plot, it should be up to the user to decide if these records should also be removed from other scatter plots (joined mode) or not (isolation mode). The isolation mode enables the user to study details of one aspect, without changing the overall picture; joined mode corresponds to brushing and linking, a well-known concept in information visualization. In order to support both modes the concept of *axes value groups* is introduced.

All axes that are linked belong to an axes value group. An axes value group is a group of axes all showing the same set of data records. The data records that are shown within an axes value group are the records for which the attributes values can be mapped on all the axes belonging to the axes value group. An attribute value can be mapped on an axis the value it within the lower and upper filter range for that axis. By default, all axes within an axes value group denoting the same attribute, have the same data and filter range. The axes belonging to a polygon are implicitly linked; meaning they will not show the link and their appropriate livo's, but are part of the same value group. Axes without links are not part of a value group, since they are not showing livo's it is irrelevant to make them member of an axes value group. There is no explicit visual feedback for an axes value group, hence axis, link and livo coloring can be done regardless of axes value groups.

The user is enabled to change the isolation mode per link, although the axes value grouping is automatically performed by the system. In isolation mode, all shown livo's have data values within the filter range of the axes belonging to the link. In joined mode, all shown livo's have data values within the filter range of the particular attributes for all given axes belonging to the value group, hence possibly checking more than two attributes. The following pseudo-code shows how the algorithm decides which livo's to display.

Function Link_Visualization

Pre: - ***axis1*** **and** ***axis2*** **denote the axes belonging to the current link. Those variables have a property called** ***value_group*** **denoting the axes value group they belong to, hence denoting the same value. They also have a property called** ***attribute*** **denoting their appropriate attribute column number, and a** ***filterlow*** **and** ***filterhigh*****, denoting the appropriate filter values for each axis.**
- ***link*** **denotes the current link being visualized having a property** ***InIsolation*** **denoting a boolean value whether the link is in isolation mode.**
Post: **- The link, hence the corresponding livo's are shown**

```
If InIsolation then
   for each r as record in dataset                      //A record being an array of values, one for each attribute
      if (axis1.filerlow ≤ r[axis1.attribute] ≤ axis1.filerhigh) ∧ (axis2.filerlow ≤ r[axis2.attribute] ≤ axis2.filerhigh) then
         ShowLivo r
ElseIf not InIsolation then
   for each r as record in dataset                      //A record being an array of values, one for each attribute
     if (for all q as axis in axis1.value_group=> (q.filerlow ≤ r[q.attribute] ≤ q.filerhigh))
         ShowLivo r
```

5.3 Visualization problems

Several problems emerged while implementing the prototype tool. Visualizing large multivariate datasets gives several challenges. In this chapter solutions are presented to overcome some of the most interesting problems.

Overplotting

The problem with straightforward drawing of the livo's is that there can be too much information to display, since the resolution is limited. This is called '*overplotting*', which can lead to highly misleading images. To cope with this problem two options have been implemented: *opacity* and *histograms*.

Opacity

Johansson et al. [7] provide a way of reducing the number of drawn object without omitting, but rather adding, information into the visualization, known as opacity. Opacity reduces the number of objects by precalculating the number of livo occurrences per link. The opacity reflects the number of occurrences, while making it possible to emphasize the number of occurrences. When discussing the parallel coordinate plot in Section 3.3 line opacity were silently introduced, although not being standard PCP functionality.

Livo's are transformed to display coordinates by calculating the start and end point in pixel coordinates. By precalculation of the number of livo's starting and ending at a particular pixel position, the *weight* of the livo is determined. After the precalculations the weight of the livo is used to denote the livo opacity, using a linear scale. All livo's have a minimum opacity to make sure that they are always visible. This opacity makes

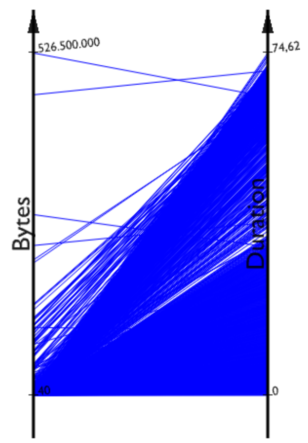

Figure 45: PCP without opacity.

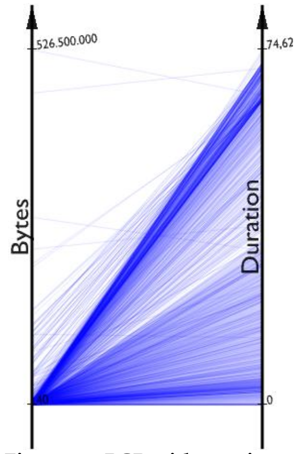

Figure 46: PCP with opacity.

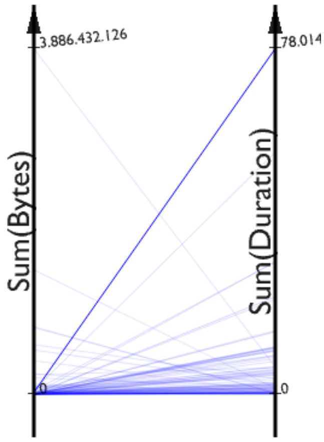

Figure 47: PCP with opacity.

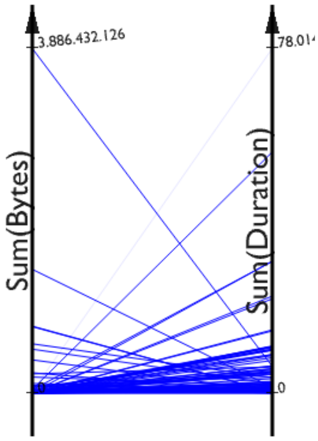

Figure 48: PCP with inverse opacity

the lines with the highest weights stand out. Figure 45 and Figure 46 show the difference between two PCP's, the first without and the latter with opacity. Inverse opacity is available to make the livo's with the lowest weights stand out, in order to find anomalies. Figure 47 and Figure 48 show an example where the first figure denotes a PCP with normal opacity and the latter denoting inverse opacity. The livo opacity is calculated per link, relatively to the maximum value within the link.

Histograms

The second option to compensate for overplotting is by making use of histograms. Histograms accumulate the number of livo's within a particular axis range and are scaled relative accordingly to the largest bin per axis. The 'base case' from Section 3.1 is used to illustrate the overplotting problem and how histograms aid in interpreting the actual image.

The first two figures, Figure 49 and Figure 50 are the previously given images for showing the base case with a scatter plot and a parallel coordinate plot. From these two images it seems that there are more occurrences for the attribute *Source IP* in the higher part of the range, the black dotted rectangle, than in the middle part of the range, the red dotted rectangle. When histograms are displayed it becomes clear that this is a wrong interpretation. The histograms show clearly that the middle part has far more occurrences than the sum of the upper part of the range.

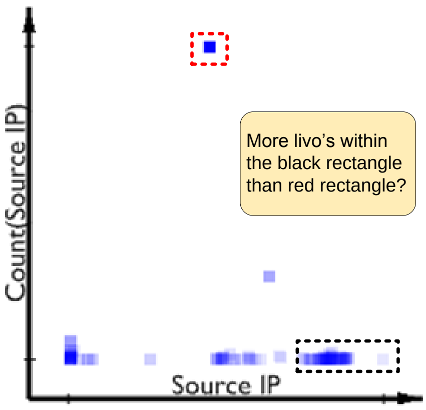

Figure 49: Base case as scatter plot.

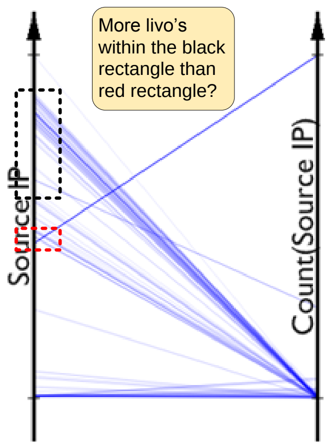

Figure 50: Base case as PCP.

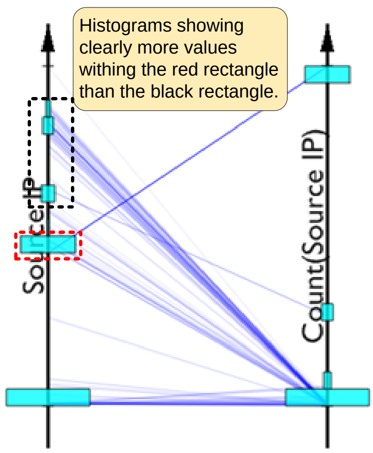

Figure 51: Base case as PCP showing histograms.

Livo coloring

Users might be interested in subsets of records and want to see the differences between them. Data values selection, visualized as highlighted livo's, does not distinct between several subsets. Only one selection can exist, so in order to provide the ability of data subset comparison, data coloring is introduced. Data coloring is an extension to data highlighting, enabling the user to give a subset of records a specific color.

Since the user is enabled to color encode a subset of records, this might lead to problems while drawing the livo's. It is possible that one particular livo encodes multiple records having the same attribute values for one particular link, but different attribute values for other attributes. If the user color encodes the records based on the attribute for which they have different values, the system has several options to color encode the livo. An example of such a problem is for instance a dataset with three attributes where there exist two records having the first two attributes exactly the same value, but the third attribute different. If the user defines color encoding on the third attribute, the two records are color encoded differently. If the user displays the attribute relation between the first two attributes, both of the records are displayed as the same livo, but they should be colored coded differently. We provide the following solution for this problem: for a scatter plot type livo one color is used, being the color most frequently used. For a PCP type livo the two most frequently used colors for the records are used. The livo start point denotes the value occurring most, whereas the point on the second axis denotes the second most frequently used color for that livo.

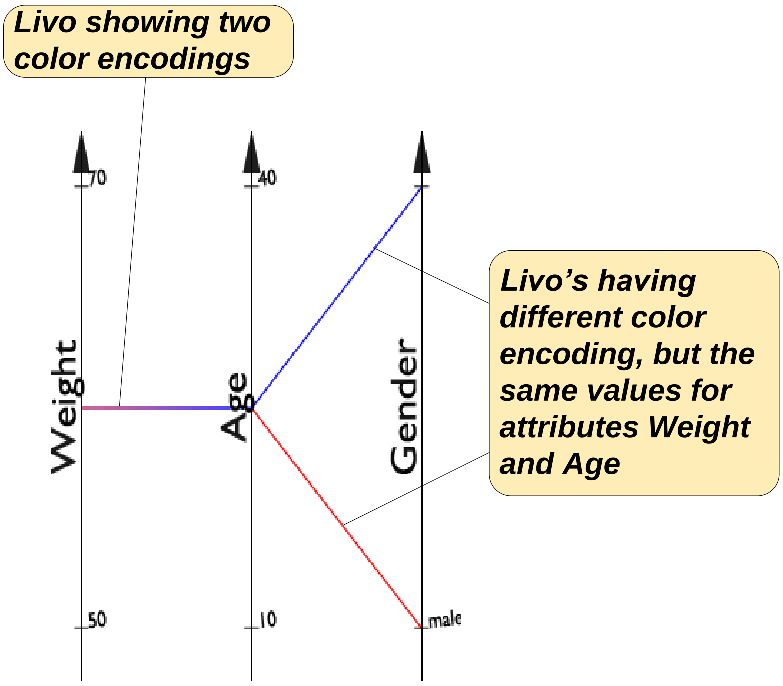

Figure 52: FLINAplot showing a livo with color blending.

Record	Weight	Age	Gender	Color encoding (based on gender)
1	60	25	Male	Blue
2	60	25	Female	Red

Table 3: Example dataset for livo coloring.

A linear blending function from OpenGL, also known as a gradient, colors the other pixels on the edge connecting the start and end point, hence making it clear that the livo encodes multiple colors. An example of a dataset with such a property is given in Table 3 and visualized by Figure 52. In Chapter 8 alternative options are discussed to show the colors belonging to a certain livo, when several color encodings are possible.

5.4 Aggregation

The user is often interested in certain frequently used functions on, possible combinations of, attributes. For instance, the user might be interested in investigating the sum of all values that oblige to certain characteristics. If a dataset is investigated with geographical data, the user might want to quickly identify the maximum, minimum, average or sum for a particular attribute, all belonging to the same geographical space. It might be even the case that the user wants to see certain characteristics for attributes having to fulfill multiple characteristics. This is a challenging task in the current situation, as well as visualizing the dataset using other visualization methods and tools. To cope with this, we introduce the idea of data aggregation. The user has an interface available to select which attribute to aggregate over, using which standard function, while grouping it by which available attributes. After creating the selection with an interface, a new attribute is calculated displaying the result from the aggregation. This way the user is enabled to see more information for the dataset in a straightforward fashion.

Table 4 gives an example of data aggregation, for which the FLINAplot is shown in Figure 53. Giving an identifiable example for this table might make it clearer. Assume having a dataset with the following attributes:

- **A** = continent, where the number stands for the continent *(1=Asia,2=Africa,3=North America,4=Europe);*
- **B** = gender, where the number represents *male* or *female* in a certain region for the continent *(1=male, 2=female);*
- **E** = population, where the number represents a population amount for the given continent and gender in a certain region.

A	B	C	D	E	F = Sum(D), grouped by A	G = Sum(D), group by B	H = Sum(E), grouped by A,B	I = Maximum(C), grouped by A
1	1	10	42	400	55 *(D1+D2)*	1076 *(D1+D2+D3+D5+D6+D7)*	490 *(E1 + E2)*	10 *(C1)*
1	1	8	13	90	55 *(D1+D2)*	1076 *(D1+D2+D3+D5+D6+D7)*	490 *(E1 + E2)*	10 *(C1)*
2	1	4	423	756	598 *(D3+D4+D5)*	1076 *(D1+D2+D3+D5+D6+D7)*	1699 *(E3 + E5)*	213 *(C4)*
2	2	213	52	454	598 *(D3+D4+D5)*	52 *(D4)*	454 *(E4)*	213 *(C4)*
2	1	23	123	943	598 *(D3+D4+D5)*	1076 *(D1+D2+D3+D5+D6+D7)*	1699 *(E3 + E5)*	213 *(C4)*
3	1	1	52	135	52 *(D6)*	1076 *(D1+D2+D3+D5+D6+D7)*	135 *(E6)*	1 *(C6)*
4	1	4	423	110	423 *(D7)*	1076 *(D1+D2+D3+D5+D6+D7)*	110 *(E7)*	4 *(C7)*

Table 4: Example of data aggregation, where the last three columns are aggregated. With the cell the formula explains which column cells are summed or what the maximum value is for the specified column for the selected grouping.

When interested in visualizing the amount of male and females in a certain continent, the visualization is unable to represent this in a distinctive manner. Retrieving the values might be accomplished by filtering and visualizing multiple separate links to show each of them individually. The user is still required to make an estimation of the sum for certain continents, as they are not given as a single value. Therefore we argue that it is useful to enable the user to define aggregated fields. Column E, being an aggregated field for the given example, denotes the required values. By means of transforming the user's interest in an aggregated field, this newly created attribute can be used within the application like any other attribute.

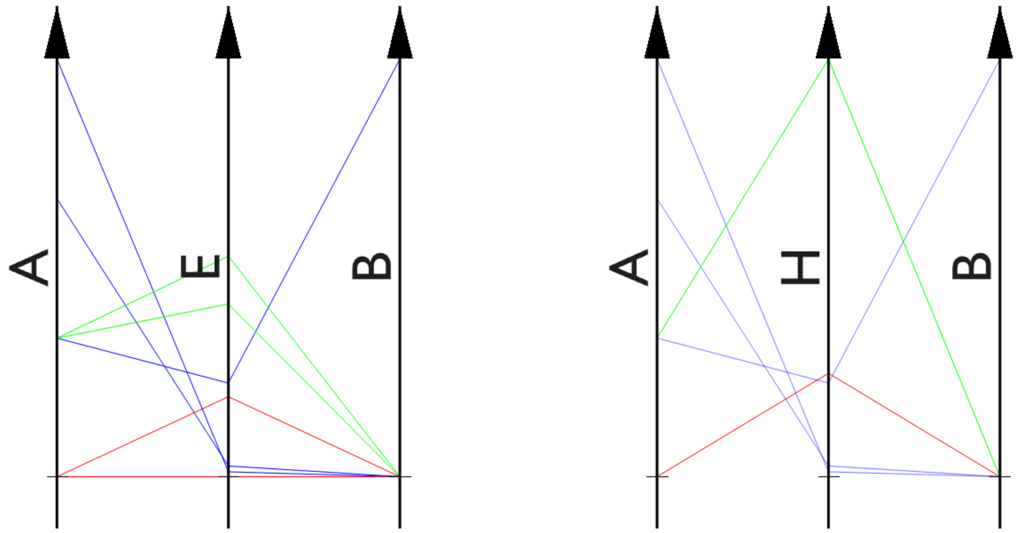

Figure 53: Figure showing data aggregation, where the right PCP is with data aggregation. The axes are showing the same value range. The colored livo's show the difference between normal and aggregated records, providing information otherwise harder to perceive.

5.5 Implementation

FLINAview is written in programming language Pascal, using the CodeGear Delphi 2009 IDE environment . The application is platform dependant and is only supported on Windows or Windows emulated operating systems. The source code consists of 32,173 lines, excluding lines of coded from included libraries. FLINAview uses a MySQL database for loading and saving datasets, and creation of aggregated fields. OpenGL is used for displaying the visualization graphics, whereas a third party open source library (GL2PS), was included to allow for exports of FLINAplots to commonly used graphic formats.

The application was developed on two different computers, having the following specifications:

Development system at home

Processor	AMD Athlon 64 X2 Dual Core Processor 5200+ 2.7 Ghz
Memory	2,00 GB
Videocard	ATI Radeon HD 2600 Pro
Operating System	Microsoft Vista Home Premium Service Pack 2

Development system at Eindhoven University of Technology

Processor	Intel Core 2 Quad CPU Q6600 2.4 Ghz
Memory	4,00 GB (3,25 GB usable)
Videocard	ATI Radeon HD 2400 Pro
Operating System	Microsoft Windows 7 Enterprise

The development system from the Eindhoven University of Technology was the system used during for the user study, as described in Chapter 7.

Chapter 6

Examples

In previous chapters the idea of Flexible Linked Axes was described as well as the interface for a tool supporting the idea. This chapter gives an impression of FLINAplots manually created with the tool, to give an overview of the possibilities of FLINAview. Most of these examples were shown to the users participating in the user study, prior to investigating their own dataset, see Chapter 7 for the user study.

Figure 54 to Figure 58 show the iris dataset[2]. The iris dataset consists of five attributes: *sepallength*, *sepalwidth*, *petallength*, *petalwidth* and *class*. The challenge of this dataset is to classify flowers into one of the three classes from the attribute c*lass*, color coded in the images, according to the values of the other attributes. The color of the axes and the label denote the attribute, with

- A (black) = sepallength;
- B (pink) = sepalwidth;
- C (orange) = petallength;
- D (blue) = petalwidth;
- E (gray) = class.

The images depict an *all-attributes-link* or *fully connected* view, being all possible unique links for a number of attributes, ranging from two to five attributes.

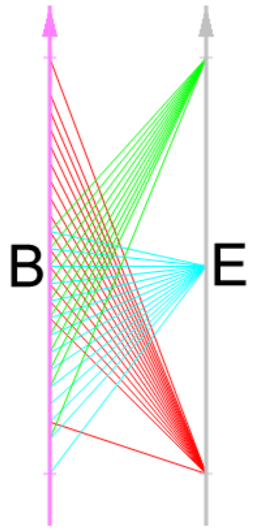

Figure 54: FLINAplot showing two attributes fully connected.

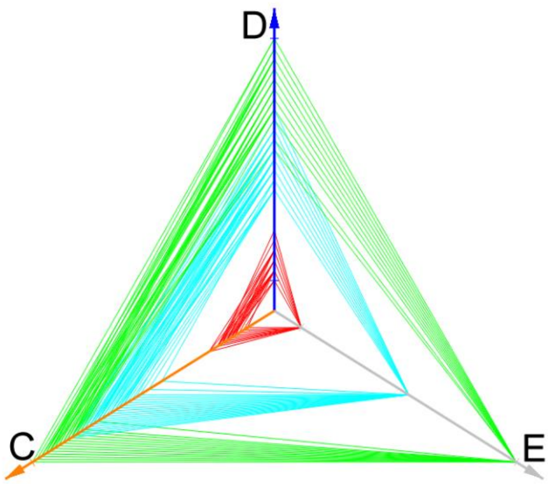

Figure 55: FLINAplot showing three attributes fully connected.

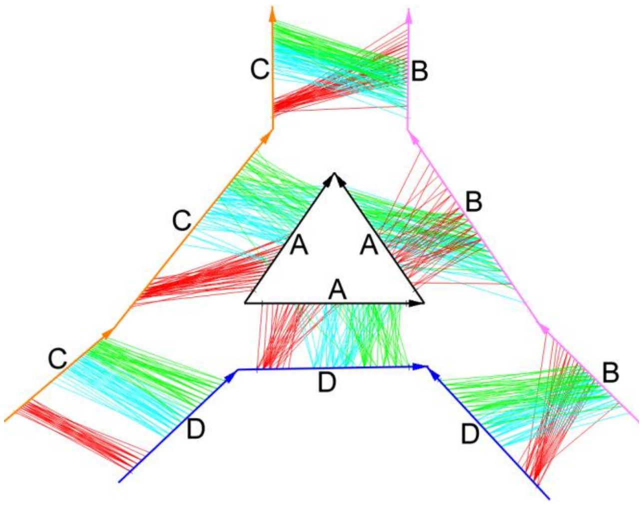

Figure 56: FLINAplot showing four attributes fully connected.

[2] http://archive.ics.uci.edu/ml/datasets/Iris

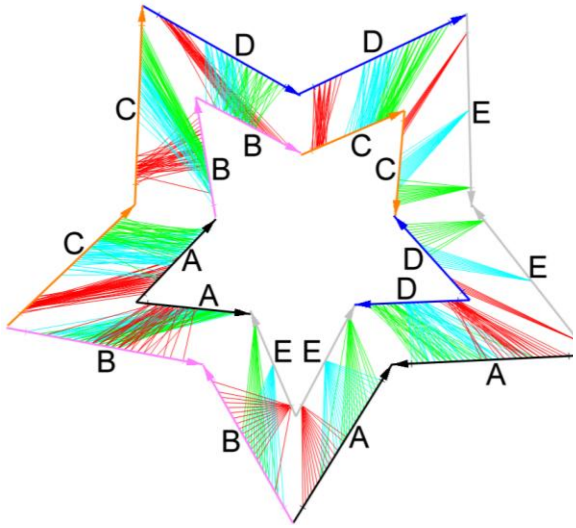

Figure 57: FLINAplot showing five attributes fully connected .

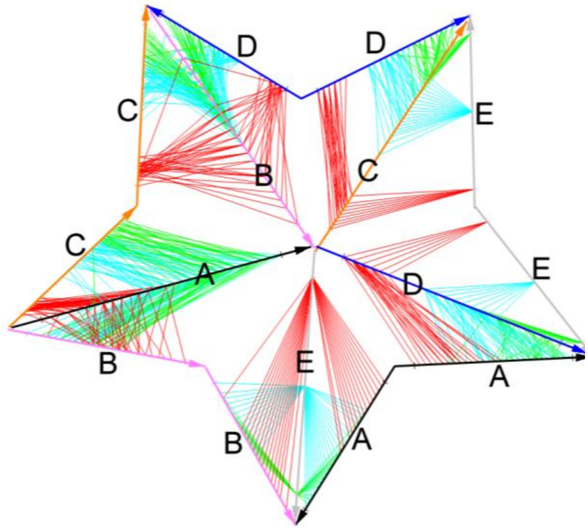

Figure 58: Alternative FLINAplot showing five attributes fully connected .

The images provide a general overview on the dataset which the user can use to create other visualizations, by focusing on the important aspects from the dataset. This done for the given dataset and the result is shown in Figure 59.

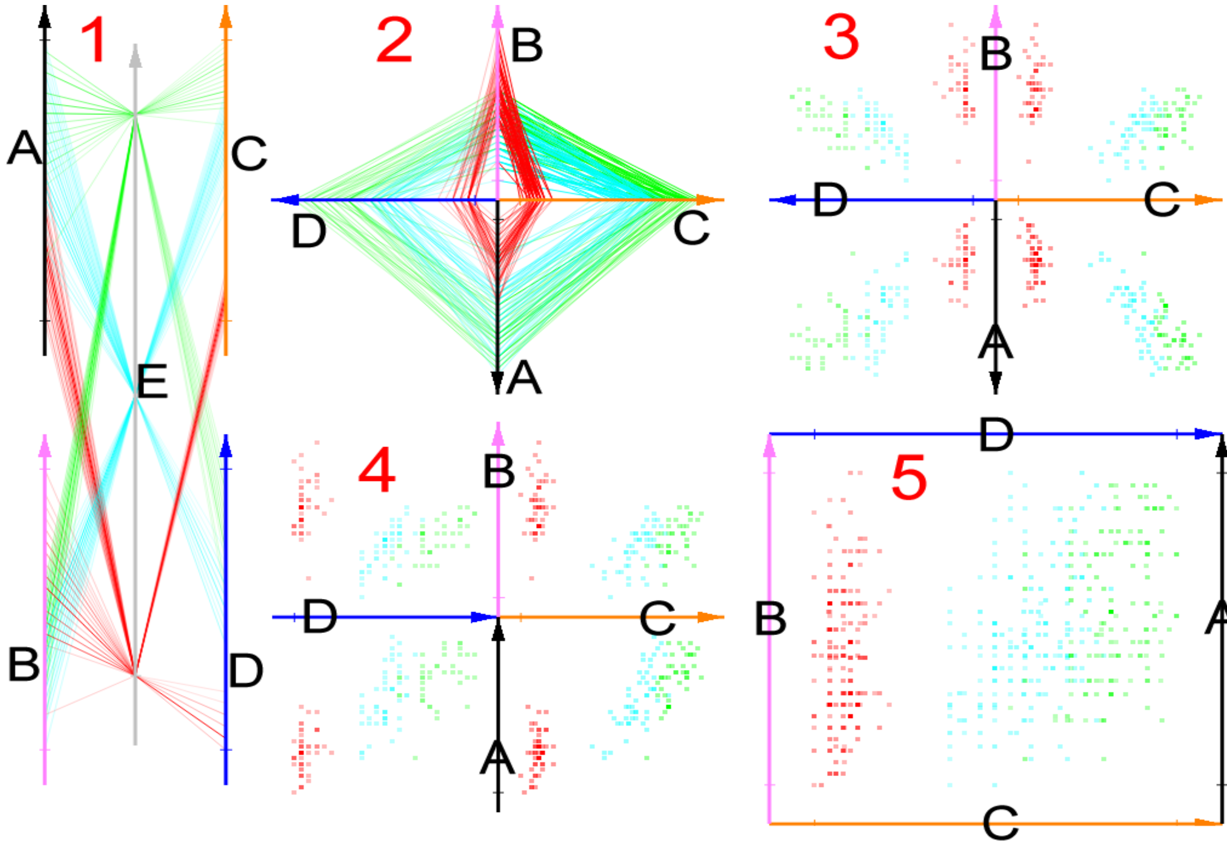

Figure 59: Iris dataset examples showing different ways to visualize characteristics from the Iris dataset.

The left example, with number 1, shows a layout where all attributes are linked to the *class* attribute. It shows that direct classification seems possible on the attributes with label *C* and *D*. This information was used to define a radar chart like layout (number 2). Here it becomes clearer that in general attribute *B* has the most unique value distribution, when looking at value distribution per class per attribute. The link types where transformed into scatter plot links (number 3). Here it shows class grouping clearly per scatter plot. When changing the axes orientation (number 4), it seems the scatter plots are reasonably similar in horizontal direction. This characteristic was used to merge the 4 scatter plots into one larger scatter plot as shown at number 5. This scatter plot still shows grouping in horizontal direction, hence showing attribute classification in a completely different way than shown at number 1.

6.1 Cars dataset

The following images are for the cars dataset[3]. The cars dataset consists of eight attributes; *acceleration*, *cylinders*, *miles per gallon*, *origin*, *model year*, *engine displacement*, *horsepower* and *weight*. These images also give an all-attributes-link view for the dataset, for an increasing number of attributes. The axes attribute encoding is as follows:

- A (black) = miles per gallon;
- B (yellow) = acceleration;
- C (pink) = cylinders;
- D (gray) = horsepower;
- E (blue) = weight;
- F (orange) = engine displacement;
- G (green) = origin;
- H (cyan) = price (added attribute with random values, with formula: *Rand() * 40000 + 10000*);
- I (Olive) = model year

The livo's are color encoded on number of cylinders; 3=blue, 4=purple, 5=orange, 6=cyan, 8=green.

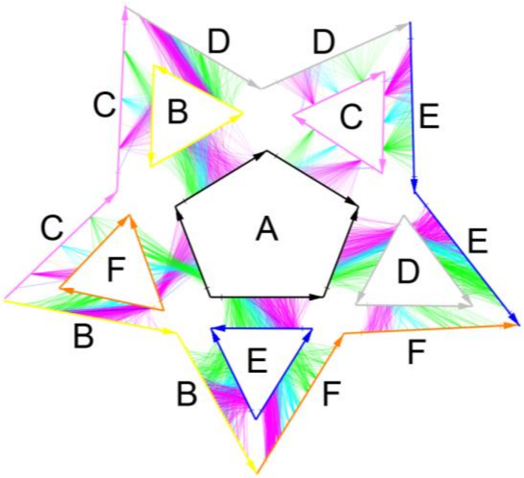

Figure 60: FLINAplot showing six attributes fully connected.

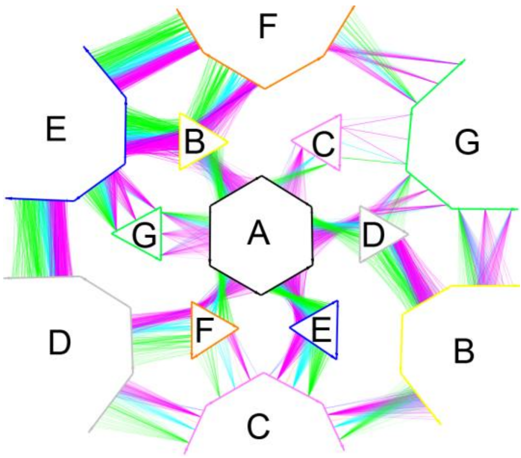

Figure 61: FLINAplot showing seven attributes fully connected with Many-to-Many relational Parallel Coordinate Display layout.

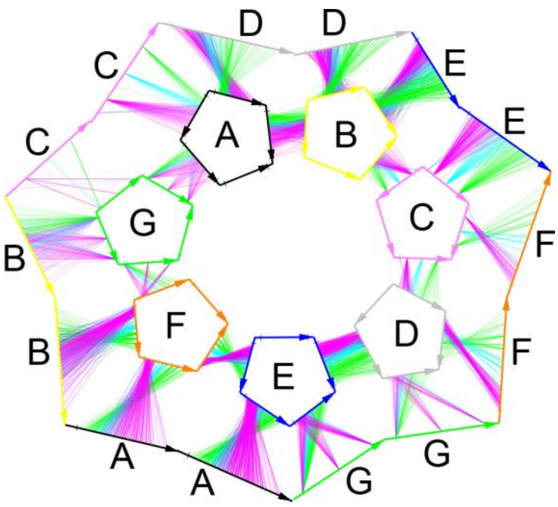

Figure 62: Alternative FLINAplot showing seven attributes fully connected.

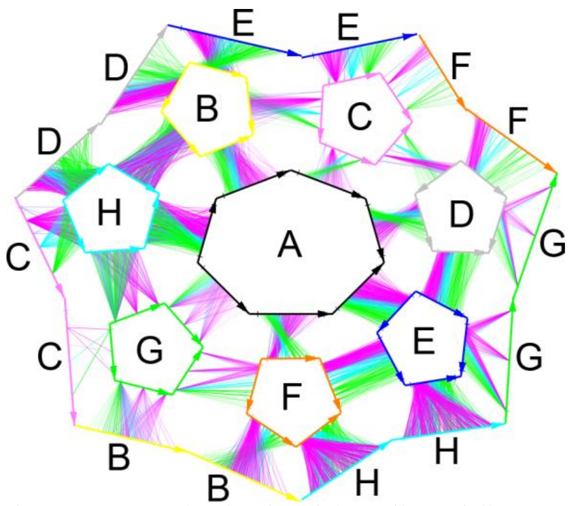

Figure 63: FLINAplot showing eight attributes fully connected, one additional attribute was added.

[3] http://lib.stat.cmu.edu/datasets/

The given FLINAplot layouts show all the attributes and their correlations without constantly changing the point of view. All relations for any given attribute can be seen by only changing the point of view once, or in the case of nine attributes (Figure 64), changing the point of view at most two times. Figure 64 shows another interesting property of FLINAplots, the ability to define stand-alone layouts and combining them into a larger layout. The middle part from the figure is independently defined. The outer and inner parts of the given layout do not share any axis, hence being two separate value groups.

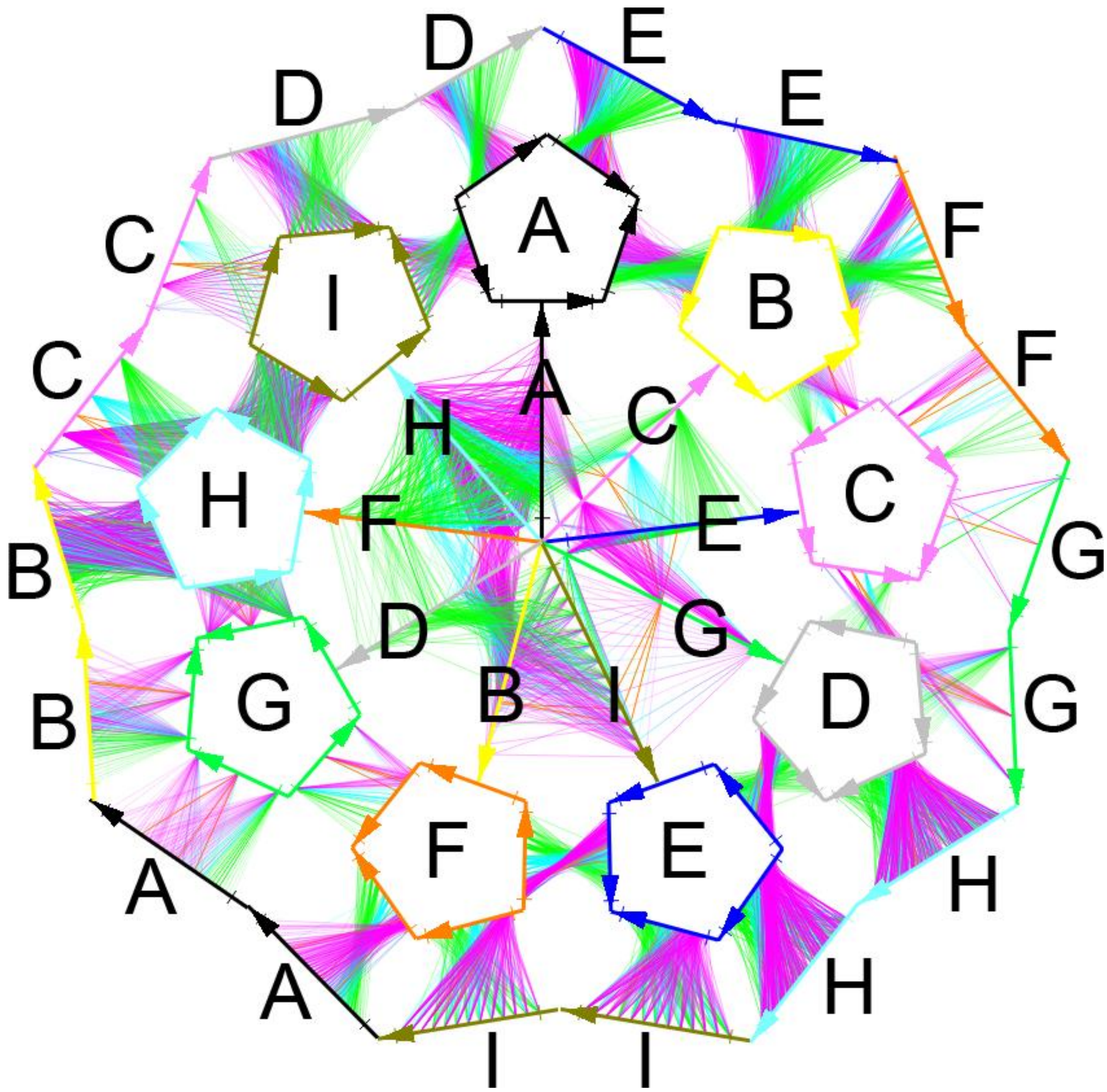

Figure 64: All-attributes-link view for Cars dataset with nine attributes, where one additional attribute was added.

6.2 All-attribute-links

The presented all-attribute-link examples are created manually, although it was tried to define a generic approach for the creation of an all-attribute link for an arbitrary number of attributes. A generic approach could not be defined, although the creation of the all-attribute-link layouts for up to nine attributes, showed interesting properties. When defining the layouts it showed it was only required to define a layout for the even numbers of attributes configurations *N*, since they could be transformed to an all-attribute-link layout for *N-1* attributes, by merely removing the centered polygon. Lind et al. [3] defined their seven attributes fully connected layout by derivation from a four attribute fully connected layout. It might be possible to derive layouts for larger number of attributes from existing layouts showing less attributes. Furthermore it showed that when the total number of attributes for the layout increased, the number of axes for the inner, not centered, polygons also increased, in order to keep the number of viewpoint changes minimized.

6.3 Changing point of view

The given examples allow for a comparison between the given visualization techniques, shown in Table 5, by means of two definitions: the *minimum number of axes* and *minimum number of required viewpoint changes* to see all links for an attribute. At most three adjacent attributes are considered as one viewpoint change. With the Flexible Linked Axes example the value between parentheses denotes the number of linked axes. Unlinked axes could be omitted from the layouts, since their only task is improving from a user point of view, hence not used for displaying attribute relations. For each comparison the highest and lowest values are shown in bold.

Five attributes	Minimum number of axes	Min. nr of viewpoint changes to see all links for an attribute
Scatter Plot Matric	**25**	3
Parallel Coordinate Plot	**10**	**2**
Flexible Linked Axes Plot (Figure 63)	20	**2**
Eight attributes		
Scatter Plot Matrices	64	**4**
Parallel Coordinate Plot	**28**	**4**
Flexible Linked Axes (Figure 64)	**66** (59)	**2**
Nine attributes		
Scatter Plot Matrices	**81**	**4**
Parallel Coordinate Plot	**36**	**4**
Flexible Linked Axes (Figure 57)	77 (68)	**3**

Table 5: Evaluation of visualization techniques with respect to the minimum number of axes and viewpoint changes.

One thing should be noted with this comparison, being the fact that Flexible Linked Axes and Scatter Plot Matrices are not optimized. Both techniques may have unused axes, not fully connected axes or duplicated links.

Chapter 7

User study and evaluation

A user study was conducted to investigate the usability of Flexible Linked Axes. Ten users, nine men and one woman between the ages 23 and 39, participated in the user study. The users are from three different areas; visualization, network monitoring and logistics. All participants are experienced computer users, and one of them was color blind. Prior to the user study two of the participants were unfamiliar with the PCP visualization technique.

The user study took one hour per person. The study consisted of the following steps:

1. *Introduction*, to explain the idea that would be tested;
2. *Demonstration of the tool*, to explain some basic functionality;
3. *User test with the Iris dataset answering seven questions*, to make the user familiar with the tool, to see if the explanation was clear and to stimulate the user to make use of the provided flexibility;
4. *Viewing examples,* showing alternative methods to visualize the Iris dataset, showing more capabilities and ideas for the usage of the added flexibility option;
5. *User study with custom dataset,* the user works with his or her own dataset to try to find new insights, find out whether flexibility is useful, and where possible come up with new ideas and visualizations;
6. *Questionnaire,* to retrieve feedback about the tool and the idea;

The users were all able to answer the questions about the Iris dataset correctly. That means that the users were able to correctly interpret the visualizations created using the tool, hence fulfilling requirement: R10: correct interpretation.

During the analysis of their custom dataset the ‘thinking out loud’ method was used. Most users started creating standard PCP’s and scatter plots for initial exploration of the dataset. After a while around 70 percent of the participants started using the flexibility by placing the axes in a more flexible configuration, whereas the remaining group played a little with the flexibility but mainly returned to the standard layouts.

Exploration methods

The user study showed that there are three exploration modes the system can be used for:

- *top-down*, starting with an overview of all the attributes and removing uninteresting information (see Chapter 6);
- *bottom-up*, build up the image step by step by adding more information subsequently (see Section 3.1) ;
- *in-between*, starting with a set of attributes and links, from there on add or remove information.

The bottom-up and top-down approach were the most commonly used techniques during the user study, and only a few used the in-between approach.

One of the users was studying decision trees, and had the idea that the tool could be used for the creation of a decision tree for the given dataset. After a while it became clear how the idea of Flexible Linked Axes could be used to build and visualize the decision tree, and led to an interesting result. A decision tree is a top-down, divide-and-conquer based approach, resulting in a tree, which partitions the dataset into smaller subsets with the growth of the tree. The smaller subsets provide information about the classification of the data [36]. The construction of a decision tree is difficult. FLINAview, however, provides functionality making it easier for the user to construct a decision tree. Figure 65 shows an example of a decision tree visualized in a tool focusing on the creation of decision trees. Figure 66 shows the decision tree in FLINAview. The image consists of two parts; the decision tree is depicted to the left and the '*decision circle*' to the right. The decision circle shows the links between one attribute, for which the decision tree had to be made, and all other attributes. Each time the decision tree was extended the decision circle was used to find the next attribute most appropriate to make a decision on. The attribute was then added to the tree, and the attribute values removed from the decision circle. The decision circle immediately showed all the values for which no decision had been made, hence aiding in finding the next best decision attribute. FLINAview initially performed well for growing the decision tree, but after two levels of branches certain values reappeared which were previously removed from the particular tree branch. Axes value groups are the reason for the reappearance of certain data values.

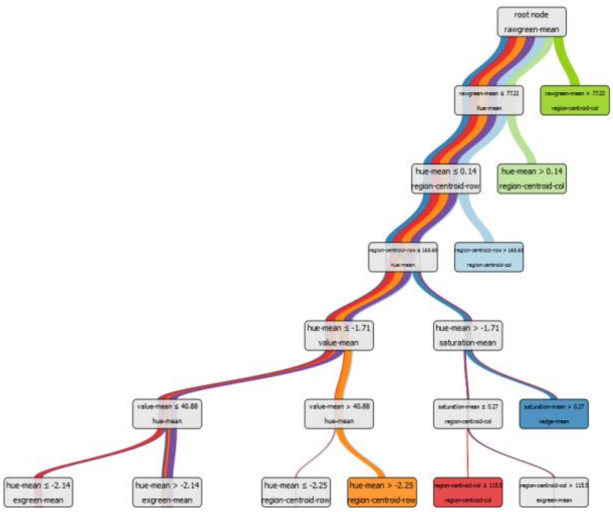

Figure 65: Decision tree example from an application developed to support decision trees.

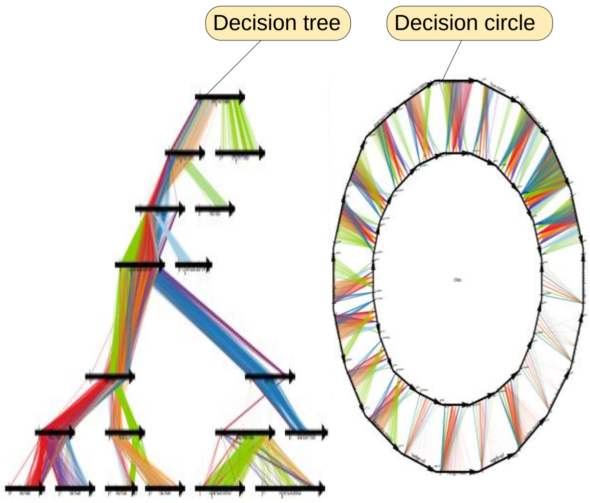

Figure 66: FLINAplot example for growing the same decision tree as given in Figure 65.

The result from this particular user test was that FLINAview is capable of showing small or simple decision trees with a limited amount of attributes. The decision circle on the other hand, is able to visualize the required information for making the consecutive decision for all decision trees.

Questionnaire

The questionnaire was introduced to receive more insight into the opinion of the users. The questionnaire was created to retrieve answers from the user on two types of questions:

1. What do you think of the idea of using flexible linked axes?
2. Does the tool support the functionality of flexible linked axes?

Although these questions were not directly asked to the users, the questions could be categorized to answer those two questions. The results from the questionnaires are shown in the following table:

FLINAview is	**Strongly Agree**	**Slightly Agree**	**Neutral**	**Slightly Disagree**	**Strongly Disagree**
easy to use	30 %	60 %	10%		
easy to understand	30 %	60 %		10 %	
useful	60 %	30 %	10 %		
The system provides a good base for					
creating and displaying Parallel Coordinate Plots	70 %	30 %			
creating and displaying Parallel Coordinate Plots	60 %	10 %	20 %	10 %	

visualizing multivariate data	60 %	30 %	10 %		
investigating dataset characteristics	40 %	30 %	30 %		

The concept of Flexible Linked Axes

has added value over Parallel Coordinate Plots	70 %	30 %			
has added value over scatter plots	10 %	70 %	10 %	10 %	

Table 6: Results from the questionnaire

Strong points

At the question: '*What do you consider to be the strong point of the system*', the subjects responded:

- Quickly visualizing correlations;
- The possibility to correlate many attributes in a single plot;
- Data selection amongst different axes, because it helps to discover new relations that are not obvious to see;
- Having two different visualization techniques;
- One sub-visualization per question, while the other visualizations remain visible. The exploration steps remain visible, hence having some history;
- Linked sub visualizations;
- Flexibility in exploring the data;
- The ability to filter data.

Weak points

At the question: '*What do you consider to be the weak points of the system*', the subjects responded:

- The two modes system, responded by 50% of the participants;
- Stability of the tool: the bugs that were still in the tool when doing the test sometimes gave minor problems, but most of them could be corrected by performing several actions to correct the flaw;
- It requires some training because it is a new way of thinking about a problem;
- The opacity representation;
- Only linear value scaling;
- Hard to see data with Parallel Coordinate Plots in general.

The question, whether the tool was powerful and user friendly enough to support the idea, was merely answered with 'Slightly Agree'. The tool provides basic functionality and several additional features, but most users argued that functionality was missing. Features that were missing or could be improved are:

- Color map instead of opacity representation;
- Polygon axes orientation in a clockwise direction;
- Global livo opacity;
- Lasso tool for selecting livo's;
- Text rendering along the axes;
- Creation of a legend to define what the colors represent;
- Defining new axes, changing the axes attribute and changing the axes attribute values
- Edge bundling, for showing hierarchies.

Having these options available in the tool would lead to increased usability, although it does not affect the core idea that is presented in this thesis. Also certain missing features might degrade existing functionality, for instance a color map, might degrade the option of global data subset coloring. Other techniques might not be applicable to generic datasets or might lead to misinterpretations, such as edge bundling. Having two separate modes overall considered to be the weakest feature of the implementation of the tool.

Questionnaire comments

Some of the results the subjects wrote down at the "Do you have additional comments" question from the questionnaire are:

- Very nice system, having high potential!
  Some time is needed though, to know the potential of the tool;
- Nice to play with;
- It took me a while to get the hang of the system and how I could use it in my particular case, but once I knew how to use it, many interesting possibilities arose.

Overall the test subjects were enthusiastic about the idea and tool, and afterwards provided useful enhancements for the tool to make it more powerful and ideas where the tool could be used for. 50 percent of the subjects responded being interested to use the tool for other dataset, 40 percent responded maybe, whereas 10 percent responded they were not interested in using the tool with for other datasets. It should be noted this was reasoned not being due to the fact that the tool not considered interesting, but rather having no dataset available to investigate.

7.1 FLINAview usage

The user study gave more insight into the best practices for FLINAview. The discussions with the participants showed that the datasets most appropriate to use FLINAview for, or more precisely the Flexible Linked Axes idea, have to fulfill certain properties.

Amount of attributes
Small amounts of attributes are considered to be visualized easier by general standard applications and business graphs. FLINAview is considered to be too complex to visualize datasets with a limited amount of attributes. The minimum number of attributes to consider using FLINAview on is four attributes. On the other hand it was argued that when the number of attributes drastically increased, first some prior dataset investigation might come in handy. Displaying all attributes when the number of attributes is large, around twenty or more, proved to be hard. This might be due to the reason that the system did not provide an automated way to generate such layouts.

Domain expert
Another criterion for using the system is having domain knowledge. Corporate management most likely would not be interested in using the system to visualize information. It was argued to be complicated for management to use. The tool seems to be most appropriate for data investigation and less for final data representation. It is merely a research visualization technique, hence the added value of the flexibility.

The conclusion of the user test was that, although the tool was considered incomplete, the idea was considered useful in many cases of multivariate data.

7.2 Evaluation

We evaluate the idea of 'Flexible Linked Axes' according to the requirements as given in Chapter 3.

	R1: Single view	R2: Attributes	R3: Generic	R4: Value distribution	R5: Correlation	R6: Unambiguous value	R7: Unambiguous occurrences	R8: Easy to understand	R9: Fast usability	R10: Correct interpretation	Total Score
Flexible Linked Axes	+	+	+	+	+	+	+	+	□	+	9

Table 7: Flexible Linked Axes analysed accoring to the given requirements

When comparing Flexible Linked Axes to the previously given visualization techniques by means of the given requirements, it shows that the total score for Flexible Linked Axes is higher. When comparing the results to the results of the other visualization techniques, only the '*Fast Usability*' requirement got degraded. This is primarily due to the feedback from the user study, for which the subjects responded it took some time to get familiar with the idea, and that the tool was not production stable. The overall result seems reasonable, since it is an extension on the combination of the two best performing techniques as previously evaluated.

Chapter 8

Future work and conclusion

FLINAview provides a solid base to visualize multivariate data in a generic way, although there are possibilities to extend the framework. However, when adding certain functionality to the system, these might violate some of the requirements given in Chapter 3.

FLINAview is capable of showing standard PCP visualizations, by simply placing a sequence of vertical axes, at equidistant space, in a row. Also, scatter plots, radar plots and a form of iconic displays can be visualized with FLINAview. Histograms provide an option for showing a one dimensional business graph, as depicted in Figure 67.

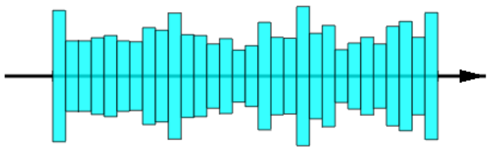

Figure 67: Histogram showing business graph like functionality in FLINAview.

In Chapter 7, several ideas have been proposed by the user test participants to enhance the FLINAview tool and Flexible Linked Axes concept, here other enhancements are discussed.

Axes mapping

Currently the axes in the FLINAview are only capable of showing the value distribution in a linear way. There are cases where linear scaling is not the best solution, for instance in cases where the value distribution is dense around a certain range and sparse for the rest of the ranges. Using more space for the dense part and less space for the sparse parts makes it possible to visualize more information, although one must keep in mind that the user should be notified of non-linear scaling. If the user is not aware of the non-linear scaling, it leads to misinterpretation.

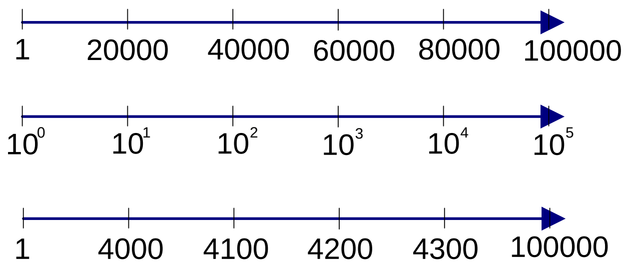

Figure 68: Example of axes mapping
Upper = linear, middle = logarithmic, bottom = zoomed

Color encoding

The current color coding, as implemented, was argued to be the functionality where the tool is best capable of improving the dataset visualizations. The current implementation of the system is only capable of encoding at most two colors per livo. The larger the dataset becomes, the more information gets discarded by this simple approach. Improving the color encoding can be done in two ways.

Color map

Livo opacity and histograms are used techniques to show which livo's occur most, although it showed that even the combination of these two techniques could be misleading. Assume a dataset having one value sub range for an attribute exceeding all others, assume a factor hundred compared to the sum of the other sub ranges. This sub range will be clearly visualized compared to the other sub ranges. The livo's and histogram bins for the other sub ranges will not be clearly distinguishable although they might encode important information. Color maps can be used to overcome certain problems. Color maps proved to be easier perceivable by users in when compared to opacity. They can be used to encode the number of occurrences for the livo's, providing clearer distinction between the numbers of occurring livo's. One disadvantage of using a color map is that it becomes more difficult to use color coding for data selections, and as such global record information most likely are harder noticeable.

Color mixing

Color mixing is another technique that could be used to encode more information in a FLINAplot. Using a color mixing technique, all appropriate colors could be encoded into a single livo. When the livo color encoding is done proportional to the amount of occurrences per color, all information will be visualized, while the most important ones are more distinct. A disadvantage is that whenever livo's overlap, this technique also has the problem of not being able to show all information. Using color grouping this problem could be avoided, by means of ordering the livo coloring in such a way that color encodings for overlapping livo's are placed as adjacent as possible, although this might require excessive calculations. Another possibility to improve the presented problem is by using color interpolation, although that could also lead to misinterpretation. Interpolated colors might be interpreted as non interpolated colors, hence representing a different value.

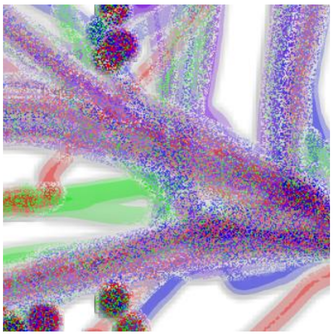

Figure 69: Color mixing example for vessel trajectories [39]

Both color encoding techniques have their problems, although they could prove to be more useful in certain cases. It should be investigated whether one of these techniques is more appropriate in comparison to the currently implemented color encoding technique.

Line curving

Krzywinski et al. [21] argue that start and endpoints are easier to spot when using Bézier curves, especially when the curves are made to impinge on the axes perpendicularly, regardless of where they are coming in from. When the number of livo's within one link gets larger, identification for the start and end point becomes increasingly more difficult. Krzywinski et al. do not provide results from a study showing that line curving performs better than straight lines, hence being an important research task prior to considering the functionality. Graham and Kennedy [37] also argue that curves improve the readability and have observed subjects acting with the visualization, but also have no test results supporting the claim.

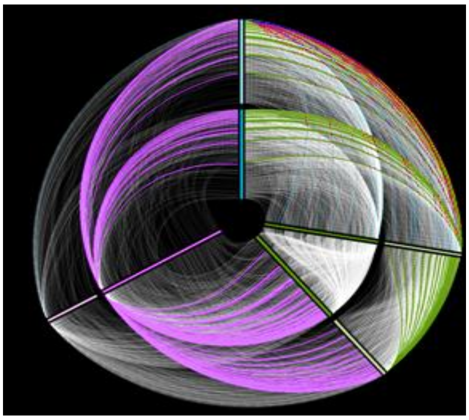

Figure 70: Example showing curved lines. http://mkweb.bcgsc.ca/linnet/

Clutter reduction

When the number of records increases, this in general leads to more clutter within the visualization, hence decreasing the interpretability. Ellis and Dix [38] compared existing clutter reduction techniques for parallel coordinate plots and their result showed that a random sampling algorithm work best for clutter reduction, with respect to speed, efficiency and accuracy. Random sampling showed to be very useful for exploratory tasks, hence the tasks argued Flexible Linked Axes are most appropriate for.

Templates or automated layouts

Having predefined templates makes data exploration easier. Several participants, during the user study, wanted to try how the previously shown layouts would visualized their dataset, but the tool did not support this option by means of available templates. It was even argued that the user should be enabled to select a number of attributes, and the system automatically generates a layout for the selected set attributes. Manually defining templates is a challenging task, especially with increasing amounts of attributes. Having an automated mechanism capable of defining layouts for a set of attributes might prove useful.

Hybrid technique

Viau et al. [25] present a way to combine PCP and scatter plots into a hybrid visualization technique called a P-SPLOM (Parallel Scatter Plot Matrix). P-SPLOMS are scatter plot matrices which are rotated along the vertical and horizontal axes. The scatter plot data values are connected, resulting in a three-dimensional parallel coordinate plot combined with scatter plots. Hybrid visualization techniques combine both visualization techniques in a straightforward way although it leads to more clutter.

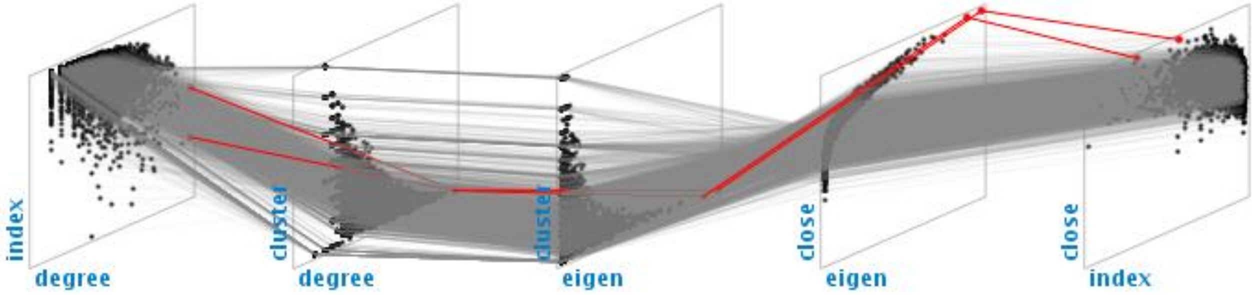

Figure 71: P-SPLOM example, showing the combination of scatter plots and parallel coordinate plots [25]

8.1 Evolution

The idea of Flexible Linked Axes was initially created for a specific domain, but generalized to support all types of datasets. The application still has a several options available for evolution. We define three possible evolution paths, whereas combinations of those paths are also feasible:

- Domain specific;
- Visualization extension;
- General functionality extension.

Domain specific

The feedback from the user test showed that the participants were looking for functionality that would aid them in investigating their particular type of dataset. Network monitoring dataset explorers argued that adding more functionality based on IP addresses, for instance selecting them by means of a subnet, would be useful. Other, non network related datasets do not have an IP address types of attributes. The number of domain specific requirements from the users study was large, hence implementing them would decrease the overall usability. The tool would provide much functionality not required by most of the users, only distracting them during their dataset investigation.

General functionality extension

In contrast to domain specific functionality general functionality was also suggested. The additional functionality, for which some of it is explained in this chapter, would make the system more powerful as a generic dataset research tool.

Visualization extension

It showed that although commonly used visualization techniques are used within the tool, not all users were familiar with these techniques. It required explanation for certain users to interpret the resulting images. The users sometimes tried to translate visualization back to more business graph kinds of visualization techniques. Having more different visualization techniques, especially more familiar ones, available in the system might lead to better understanding of the other visualization techniques and reduces the required time to get familiar with the visualizations.

Combinations

Combinations for the given evolution paths are possible, e.g. combining visualization and generic functionality extensions lead to a larger system with more options, hence more likely that the user has the functionality he is interested in. However, such evolution for the system might violate or degrade previously given requirements.

8.2 Conclusion

This thesis has shown a flexible technique capable of visualizing multivariate data. The research started with network monitoring data visualization, but was generalized to an idea applicable to all types of multivariate data. The user study showed that the idea was considered valuable by the participants. After the first presentation a user described FLINAview as: "Visual data mining". Although the result was not as the network monitoring experts initially expected, they were pleased with the final result. It gave them a completely new view and exploration methods for their datasets, capable of showing relations that previously could not have been investigated, leading to new dataset insights. The result from this research is a generic idea together with a flexible tool. Additional research might enhance the idea and tool, whereas more real life cases of dataset investigation of the tool emphasize the usefulness.

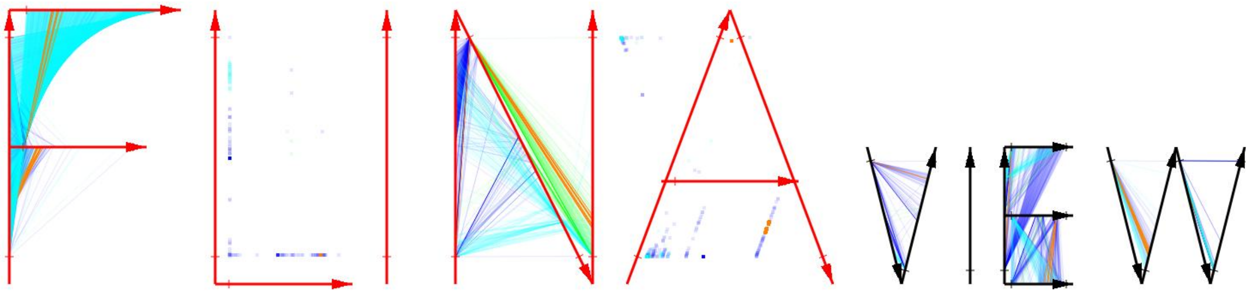

Figure 72: FLINAplot showing some of the possibilities of Flexible Linked Axes, implemented in FLINAview.

References

[1] Alfred Inselberg
"The Plane with Parallel Coordinates"
The Visual Computer, Vol.1, 1985, pages 69–91

[2] Xiaoxin Yin, William Yurcik, Adam Slagell
“VisFlowConnect-IP: An animated Link Analysis Tool For Visualizing NetFlows”
Proceedings of the ACM Workshop on Visualization and Data Mining for Computer Security, 2004, pages 26-34

[3] Mats Lind, Jimmy Johansson, Matthew Cooper
“Many-to-Many Relational Parallel Coordinates Displays”
13th international Conference on Information Visualization, 2009, pages 25-31

[4] Ji Soo Yi, Rachel Melton, John Stasko, Julie A. Jacko
“Dust & Magnet: multivariate information visualization using a magnet metaphor”
Information visualization 2005, vol. 4, pages 239-256

[5] Rida E.A. Moustafa and Edward J Wegman
“On Some Generalizations of Parallel Coordinate Plots”
Seeing a million, A Data Visualization Workshop, 2002

[6] D.A. Keim
“Information Visualization and visual data mining”
IEEE Transactions on Visualization and Computer Graphics, 8, 2002, pages 1-8.

[7] Jimmy Johansson, Patric Ljung and Matthew Cooper
Depth Cues and Density in Temporal Parallel Coordinates
Symposium on Visualization 2007, May 2007, pages 35-42

[8] Helwig Hauser, Florian Ledermann and Helmut Doleisch
Angular Brushing of Extended Parallel Coordinates
Proceedings on IEEE Symposium on Information Visualization, October 202, pages 127-130

[9] Jarry Claessen
Visualization of Network Monitoring data
Eindhoven University of Technology, Literature study, 2010

[10] Maurice d’Ocagne
“Coordonnées Parallèles et Axiales: Méthode de transformation géométrique et procédé nouveau de calcul graphique déduits de la consideration des coordonnées parallèlles"
January 1885, Paris, Gauthier-Villars

[11] Robert Amar, James Eagan and John Stasko
“Low-Level Components of Analytic Activity in Information Visualization”
Proceeding of the Symposium on Information Visualization (InfoVis), 2005, pages 111-117

[12] Michel Beaudouin-Lafon
“Designing interaction, not interfaces”
Conference on Advanced visual interfaces, Italy, 2004, pages 15-22

[13] Eser Kandogan
“Star Coordinates: A Multi-dimensional Visualization Technique with Uniform Treatment of Dimensions”
Proceedings of IEEE Information Visualization, Hot Topics, 2000, pages. 4-8

[14] Georges Grinstein, Marjan Trutschl, Urška Cvek
“High-Dimensional Visualizations”
Proceedings of the Visual Data Mining workshop, 2001, pages 7-19

[15] Alfred Inselberg, Bernard Dimsdale
Parallel coordinates : A Tool for Visualizing Multi-Dimensional Geometry
Proceedings of the 1st conference on Visualization '90, 1990, pages 361-378

[16] Wen-Yuan Liu, Bao-Wen Wang, Jia-Xin Yu et al.
Visualization classification method of multi-dimensional data based on radar chart mapping
Proceeding of the 7th International Conference on Machine Leaning and Cybernetics, Kumming, 12-15 July 2008, pages 857 – 862

[17] Ying-Huey Fua, Matthew .O. Ward, E.A. Rundensteiner
“Hierarchical parallel coordinates for exploration of large dataset”
Proceeding of visualization ’99, October 1999, pages 43-50

[18] Matthew O. Ward
“Handbook of Data Visualization, Springer Handbooks of Computational Statistics, 2008, II, 179-198,
Multivariate Data Glyphs: Principles and Practice

[19] Matthew O. Ward
“XmdvTool: integrating multiple methods for visualizing multivariate data”
Proceedings of the conference on Visualization '94, 1994, pages 326 - 333

[20] Daniel A. Keim, Hans-Peter Kriegel
“Visualization techniques for Mining Large Databases: A comparison”
IEEE Transactions on Knowledge and Data Engineering, volume 8 issue 6, Dec 1996, pages 923-938

[21] Martin Krzywinski, Jacqueline Schein, Inanç Birol, Joseph Connors, Randy Gascoyne, Doug Horsman, Steven J Jones, Marco A Marra
“Circos: An information aestethic for comparative genomics”
Genome Research, Vol. 19, No. 9, 1 September 2009, pp. 1639-1645

[22] Ben Shneiderman
“The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations”
Proceedings of the IEEE Symposium on Visual Languages, 1996, pages 336-343

[23] Danny Holten
“Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data”
IEEE Transactions on Visualization and Computer Graphics, 2006, Vol. 12, No.5, Pages 741-748

[24] Jorik Blaas, Charl P. Botha,Frits H. Post
“Extension of Parallel Coordinates for Interactive Exploration of Large Multi-Timepoint Data Sets”
IEEE Transactions on Visualization and Computer Graphics, Vol. 14, No. 6, December 2008, pages 1436-1451

[25] Christophe Viau, Michael J McGuffin, Yves Chiricota, Igor Jurisica
“The FlowVizMenu and Parallel Scatterplot Matrix: Hybrid Multidimensional Visualizations for Network Exploration”
IEEE Transactions on Visualization and Computer Graphics, 2010, Vol 16, No.1, January 2010, Pages 1100-1108

[26] S. Feiner and C. Beshers
“Visualizing n-dimensional virtual world with n-vision”
Computer Graphics, Vol. 24, No.2, 1990, Pages 37-38

[27] G.G. Robertson, J.D.Mackinlay, S.K. Card
“Cone trees: Animated 3D visualizations of hierarchical information”
Proceedings of the SIGHI conference on Human Factors in Computing Systems, 1991, pages 189-194

[28] Ben Schneiderman
“Tree Visualization with Tree-Maps: 2-d Space Filling Approach”
Transaction on Graphics, Volume 11, No. 1, January 1992, pages 92-99

[29] Daniel A. Keim
“Designing Pixel-Oriented Visualization Techniques: Theory and Applications”
Transactions on Visualization and Computer Graphics, Volume 6, No. 1, March 2000, pages 59-78

[30] Ronald M. Pickett, Georges G. Grinstein
“Iconographic Displays for Visualizing Multidimensional Data”
Proceedings on Systemens, Man and Cybernetics, August 1988, pages 514-519

[31] Tom Goldring
“Scatter (and other) Plots for Visualizing User Profiling Data and Network Traffic”
Proceedings of workshop on Visualization and data mining for computer security, 2004, pages 119-123

[32] Andrada Tatu, Peter Bak, Enrico Bertini, Daniel Keim, Joern Schneidewind
“Visual quality metrics and human perception: an intial study on 2D projections of large multidimensional data”
Proceedings of the International Conference on Advanced Visual Interfaces, May 2010, Pages 49-56

[33] Xiping Dai, Frank Hardisy
“Conditioned and Manipulable Matrix For Visual Exploration”
Proceedings of the conference on Digital Government Research, 2002, pages 1-4

[34] Helwig Hauser, Florian Ledermann, Helmut Doleisch
“Angular Brushing of Extended Parallel Coordinates”
IEEE Symposium on Information Visualization, 2002, pages 127-130

[35] R Becker, W. Cleveland
“Brushing scatterplots”
Technometric, Volume 29, No. 2, 1987, pages 127-142

[36] J. R. Quinlan
“Decision Trees and Decisionmaking”
IEEE Transactions on Systems, Man and Cybernetics, Volume 20, No. 2, March 1990, pages 339-346

[37] Martin Graham, Jessie Kennedy
“Using curves to Enhance Parallel Coordinate Visualizations”
Proceeding on the Seventh International Conference on Information Visualization, July 2003, pages 10-16

[38] Geoffrey Ellis, Alan Dix
“Enabling Automatic Clutter Reduction in Parallel Coordinate Plots”
IEEE Transactions on Visualization and Computer Graphics, Volume 12, No. 5, September 2006, pages717-724

[39] Roeland Scheepens, Niels Willems, Huub van de Wetering, Jarke J. van Wijk
Interactive Visualization of Multivariate Trajectory Data with Density Maps.
IEEE proceedings Pacific Visualization Symposium, March 2011