Flexible Linked Axes for Multivariate Data Visualization

Jarry H.T. Claessen and Jarke J. van Wijk, *Member, IEEE*

Abstract—Multivariate data visualization is a classic topic, for which many solutions have been proposed, each with its own strengths and weaknesses. In standard solutions the structure of the visualization is fixed, we explore how to give the user more freedom to define visualizations. Our new approach is based on the usage of Flexible Linked Axes: The user is enabled to define a visualization by drawing and linking axes on a canvas. Each axis has an associated attribute and range, which can be adapted. Links between pairs of axes are used to show data in either scatterplot- or Parallel Coordinates Plot-style. Flexible Linked Axes enable users to define a wide variety of different visualizations. These include standard methods, such as scatterplot matrices, radar charts, and PCPs [11]; less well known approaches, such as Hyperboxes [1], TimeWheels [17], and many-to-many relational parallel coordinate displays [14]; and also custom visualizations, consisting of combinations of scatterplots and PCPs. Furthermore, our method allows users to define composite visualizations that automatically support brushing and linking. We have discussed our approach with ten prospective users, who found the concept easy to understand and highly promising.

Index Terms—Multivariate data, visualization, scatterplot, Parallel Coordinates Plot.

1 Introduction

Multivariate data are a ubiquitous datatype, which describe homogenous sets of items by values of their attributes. An item can denote for instance a person, with attributes like gender, height, weight, and age; a sale, with attributes such as time, value, and product type; an internet packet, with attributes sender, receiver, time stamp, and size; etc. Many methods have been developed to provide insight into multivariate data using interactive visualizations, with the scatterplot, the Parallel Coordinates Plot (PCP)[11], and the radar chart as important representatives.

In this article we propose yet another approach: Flexible Linked Axes. The method is based on a simple idea. In all methods so far the structure of the visualization is more or less fixed, and the user can only change some properties of the representations provided. We propose to enable users to define and position coordinate axes freely, and specify suitable visualizations by linking these axes. This approach enables users to define scatterplots, PCPs, and radar charts, but also to develop highly customized visual representations.

In Section 2 we describe the background for our work and discuss limitations of existing approaches as well as some inspiring other work. The concept of Flexible Linked Axes is described in Section 3, followed by a number of examples of their use in Section 4. We have discussed our prototype with 10 prospective users, and report on the results in Section 5. Finally, we give conclusions and suggestions for future work in Section 6.

2 Background

A multivariate dataset can be described as a table T with cells indexed with row number i and column number j, with $i = 1, \ldots M, j = 1, \ldots, N$. Each row T_{i*} denotes an item, each column T_{*j} an attribute A_j, and the value T_{ij} stored in a cell denotes the value of attribute j for item i. Each attribute has an associated domain, such as natural numbers, real numbers, or strings.

A variety of approaches can be used to analyze such data sets. Many methods are available for multivariate data analysis, such as the use of regression analysis, multidimensional scaling, and cluster analysis. Visualizations of their results are typically generated by considering items as points, which are projected on a derived low-dimensional space. Here we limit ourselves to purely visual methods, such as shown in Figure 1, where visualizations are generated based on the values for attributes of items and insight has to be achieved by looking at and interacting with images.

Before discussing related work, we first consider the interest of the user during multivariate data visualization. We argue that users are mainly interested in three aspects of the data: individual items and their values, the distribution of values of items for a single attribute, and the relation between values for two attributes. When these aspects are displayed properly, many of the low-level tasks identified by Amar *et al.* [2], including Retrieve Value, Determine Range, Characterize Distribution, Find Anomalies, Cluster, and Correlate can be performed.

The first aspect can be dealt with by showing items in a table, or by providing detail on item selection. We focus on the distribution of a single attribute and the relations between pairs of attributes. During exploration, users are interested in a subset of all attributes, and a subset of pairs of attributes. We can model this as a graph, where vertices represent attributes and edges relations between pairs of attributes, and denote the current interest of a user as the Attribute Relation Graph of Interest, or ARGOI for short. An ARGOI is dynamic: during exploration users will discard attributes that do not provide additional insight or pairs that show no correlation at all, and add new attributes and relations that are potentially interesting. An ARGOI is not necessarily connected: it can well be that users may want to view simultaneously a pair (A, B) and a pair (C, D). A simple ARGOI is a path, where a sequence of attributes A_j and the relations between attributes A_j and A_{j+1} are of interest, but we argue that many other configurations are potentially useful and interesting as well (Figure 2). One example is a star, where relations between one central attribute and several others are studied; if there are two or more of such central attributes, more complex configurations can be imagined.

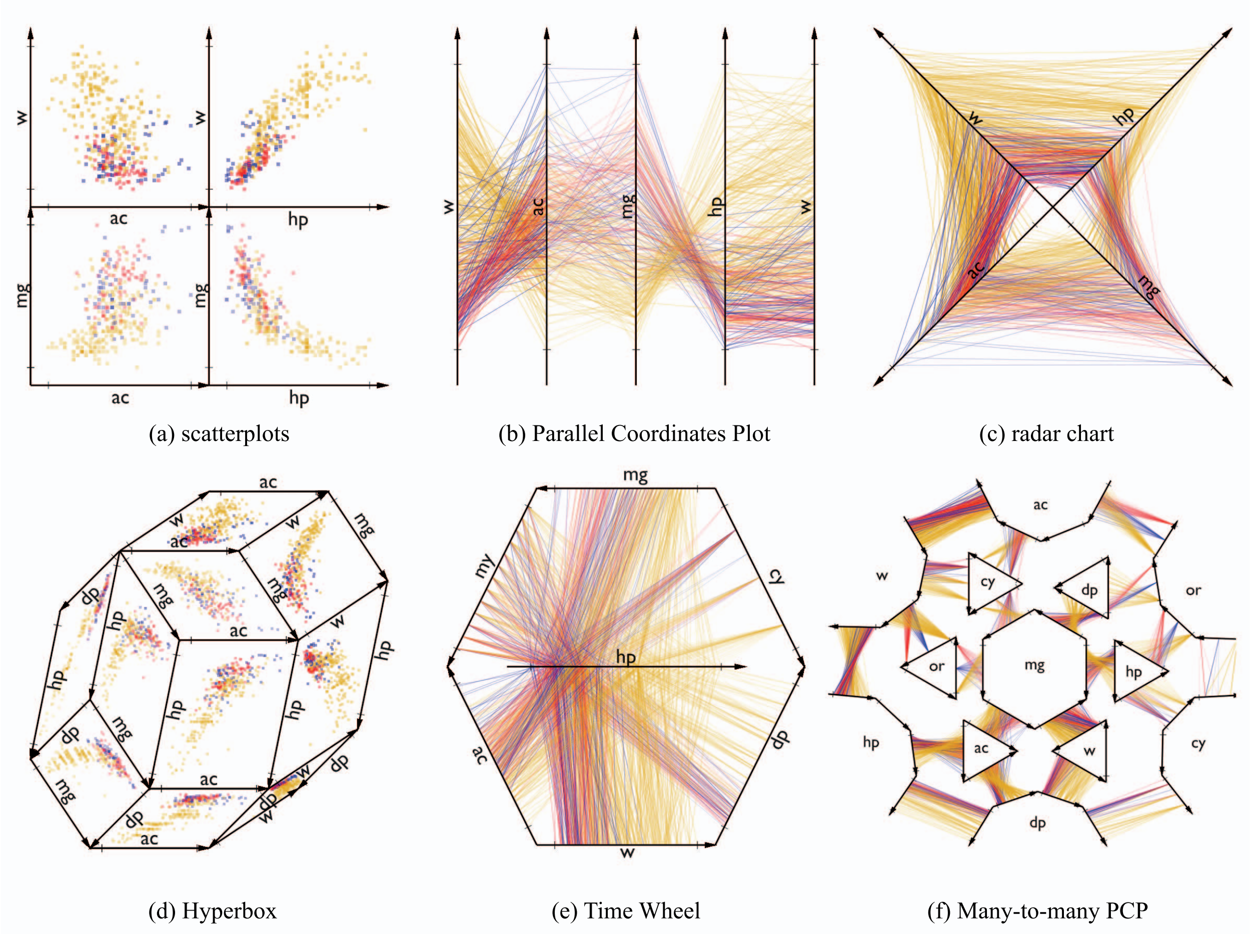
Next, we consider a number of mainstream methods for multivariate data visualization. The literature on multivariate data visualization is rich and extensive, overviews can be found elsewhere [13, 20, 8, 12]. Here we focus on axis-based methods and consider how well they support arbitrary types of ARGOIs. Figure 1 shows six of such axis-based methods, all showing the cars dataset [16].

The main method for multivariate data visualization is the scatterplot. Here the relation between two attributes is shown, in terms of ARGOIs just two nodes and a single edge. However, by relating properties of the icons (such as shape, color and size) to values of attributes, more attributes can be shown simultaneously, though the number of different colors and sizes that can be distinguished is limited. By changing the attributes of the axes, other edges of the ARGOI can be inspected, but this puts a burden on the memory of the user and makes it difficult to make comparisons. A generalization of the scatterplot is the well-known scatterplot matrix; a matrix with (a subset of all) attributes

• *Jarry H.T. Claessen and Jarke J. van Wijk are with Eindhoven University of Technology, E-mail: vanwijk@win.tue.nl.*

Manuscript received 31 March 2011; accepted 1 August 2011; posted online 23 October 2011; mailed on 14 October 2011.
For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

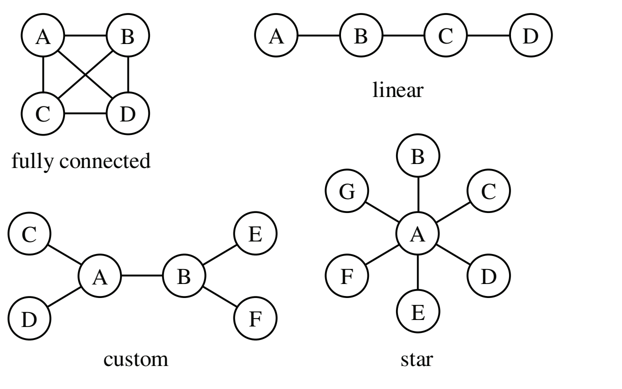
1077-2626/11/$26.00 © 2011 IEEE Published by the IEEE Computer Society



(a) scatterplots (b) Parallel Coordinates Plot (c) radar chart

(d) Hyperbox (e) Time Wheel (f) Many-to-many PCP

Fig. 1. Examples of methods for multivariate data visualization, showing the cars dataset [16], with ac = acceleration, mg = miles per gallon, w = weight, hp = horse power, yr = model year, or = origin, cy = cylinders, and dp = displacement. Colors denote the origin: yellow for USA; blue for Europe; and red for Japan.



fully connected linear custom star

Fig. 2. Examples Attribute Relation Graphs of Interest (ARGOIs). Each node denotes an attribute, edges denote relations of attributes.

along rows and columns, where the cells are filled with scatterplots. In ARGOI terms, such a scatterplot matrix shows a complete graph, as all pairs are shown. A scatterplot matrix can be considered as showing an ARGOI as incidence matrix, and it inherits its features and limitations. The complete graph is shown without clutter, but in many cases only a small subset will be of interest, and valuable screen space is used for showing less relevant pairs. Also, individual edges are depicted clearly, but following a path is difficult. For instance, one could be interested in tracing a possible causal path from attribute A to D to P to Q, which requires quite some visual navigation and breaks the flow of the analysis. Elmqvist *et al.* [6] provide with ScatterDice an elegant and effective solution for interactive exploration of scatterplot matrices.

A Parallel Coordinates Plot (PCP) [11] consists of a sequence of vertical axes, and items are displayed as polylines by connecting the positions of values at succeeding axes by straight lines. In ARGOI terms, a PCP shows a path. By reordering, adding, duplicating, and removing axes the user has freedom to define this path and here a PCP has a clear benefit over the scatterplot matrix. However, arbitrary ARGOIs are not supported. As an example, one could be interested in the relation between A and B, but also whether A is influenced by C and D, and B by E and F (Figure 2, custom ARGOI). PCPs are less familiar and at first sight less intuitive than scatterplots, but their growing popularity across many different application domains indicate their resilient value. In a radar chart (known also under many other names, such as spider chart, star chart, and Kiviat diagram) a set of axes in a star pattern is used, such that ARGOIs consisting of closed paths can be inspected.

Several authors have presented combinations of scatterplots and PCPs. Viau *et al.* [18] show how transitions from sequences of scatterplots and scatterplot matrices to PCPs and sequences of PCPs can be obtained. They connect the dots in neighboring scatterplots, and subsequently rotate along the axes to obtain PCPs. Holten *et al.* [10] found that the use of small scatterplots on top of PCPs helps to find clusters more quickly. Collins *et al.* [4] show the relations between different 2D visualizations by positioning these in 3D space and connecting elements with polylines.

Also, various authors have proposed variations on scatterplots and PCPs with alternative configurations of the axes (Figure 1(d)-(f)). The Hyperbox of Alpern and Bowers [1] is a two-dimensional projection of an N-dimensional box, which looks like a projection of a multi-facetted polygonal surface (Figure 1(d)). Starting from a set of vectors, parallelograms are defined repeatedly from neigboring vectors. The parallelograms are used to show scatterplots. Some of these parallelograms are highly skewed, leading to distorted scatterplots that are not easy to read. In our examples in the following sections we therefore constrain ourselves to standard scatterplots with orthogonal axes.

The TimeWheel of Tominski *et al.* [17] supports a star configuration in ARGOI terms (Figure 1(e)). A number of axes are radially laid out, tangent to a circle; one central axis, typically with time as attribute, is aligned with the horizontal axis. Next, relations between an attribute of a radial axis and the central axis are depicted in PCP-style, by drawing lines between corresponding values of items at these axes. The focus can be changed by rotating the radial axes and the attributes shown for the axes.

An intriguing variant has been given by Lind *et al.* [14]: many-to-many relational parallel coordinate displays, for short many-to-many PCPs. They present two PCP-style displays, where all relations between four, respectively seven (Figure 1(f)) attributes are shown. In the latter case, the axes are configured in polygons (one central hexagon, six triangles, and six half octagons), where the axes belonging to a polygon denote the same attribute. This leads to a compact display, showing the complete ARGOI. They conducted a user study in which they compared their method with a standard PCP, where the users had to spot negative correlations. The amount of errors was similar, but the subjects performed 20% faster using the new layout.

The latter cases are promising. By varying the configurations of axes interesting results can be achieved and structurally different ARGOIs can be presented to the user. In all cases, the configurations were defined by the authors. But, can't we let the user decide what he considers to be the best configuration? This is the main idea of Flexible Linked Axes.

3 Flexible Linked Axes

In this section we present how we enable users to position axes and to visualize multivariate data. The key concept is the use of Flexible LINked Axes, FLINA for short. We have developed a prototype application, called FLINAView, that allows users to define FLINAPlots: visualizations using Flexible LINked Axes. FLINAPlots can be considered as visualizations that present data according to their (abstract) ARGOIs. As a note aside, we found that the word *flina* is Swedish for *to grin*. This was coincidental, but we found this a nice tribute to our Swedish inspiration [14] and to the emotional state we aim to achieve with prospective users.

The metaphor we lean on is that of a drawing tool: the user is enabled to draw and edit graphic elements using standard conventions. We first discuss the main elements (Figure 3), followed by the interaction provided. The central element of a FLINAPlot is the *axis*. Each axis is defined along a line segment between two *point*s. An axis has an associated data attribute; a direction (from point A to B or vice versa); a margin, to control which part of the line is used; a range $[d_{\min}, d_{\max}]$ to define which part of the domain of the attribute has to be mapped on the start and end of the axis. Furthermore, the color and the label of the axis can be defined by the user, and a range $[f_{\min}, f_{\max}]$ to filter items on attribute value can be used.

A point can be shared by multiple axes. For convenience of the user, sets of axes can be defined as *polygon*s, with a user-defined number of sides. Shape and position of such polygons are defined by sweeping out a rectangle, where optionally the user can request a regular polygon. A polygon has an associated attribute and a range, which is inherited by its constituent axes, but which can be overruled per axis.

The next type of element is the *link*, defined between pairs of axes. These links define relations between pairs of axes, and hence define the edges of the ARGOI. For each link the user can define if a PCP- or a scatterplot-style has to be used to visualize the items. Given an item i, and an axis a with attribute A_j, let us define a point P_{ia} as the position on axis a corresponding to T_{ij}, *i.e.*, the projection of item i on axis a. To obtain a PCP-style visualization for two axes a and b, an item is simply shown as a line that connects P_{ia} and P_{ib}. For scatterplot-style, a dot is shown per item i at the intersection of two lines: one through P_{ia} with the direction of axis b, the other through P_{ib} with the direction of axis a. Optionally, items are only shown for a set of transitively linked axes if all projections are within the specified ranges of the axes. This enables the user to focus on subsets of items of interest, and to filter out less relevant parts of the data set.

We provide a number of features to enable users to edit their FLINAPlots efficiently and effectively. Points can be moved, a grid can be used, elements can be copied and pasted, an undo operation is provided, and properties of (multiple) elements can be changed by selecting them and entering desired values in widgets. To enable the user to keep focussing on the FLINAPlot, context-dependent pop-up menus are used where possible, for instance to select an alternative attribute for an axis or to flip its direction. When the user selects two polygons and requests a link to be created, the two best matching axes of the two polygons are determined and linked.

For the rendering of PCP-style visualizations, we use semi-transparent lines and binning of lines [15] to address overplotting issues and to speed up the rendering for large sets of items. Histograms that show the distribution of items over an axis can optionally be shown. We opted for histograms in violin-plot style, *i.e.*, symmetrically over the axis. The histograms provide valuable information in themselves, and visualize the nodes of the ARGOI. Also, they are useful for a correct interpretation of scatterplots and PCPs, as just using semi-transparency does not remedy all overplotting issues.

The user can select sets of data items by sweeping a box (for scatterplots) or by drawing a line that intersects the lines of a PCP [9]. Item selections are highlighted (using a different color, setting opacity to the maximum level, and by drawing them on top of all other graphic elements). Item selections are global, hence selected items are highlighted in all visualizations, thereby supporting linking and brushing [3]. Furthermore, to each item a color is assigned, which is used for non-selected items. The user can change the color for a set of selected items, which is useful to distinguish different classes of items.

For the cases shown in the next section, with datasets of up to 100,000 items and up to 15 links, all images could be redrawn in real-

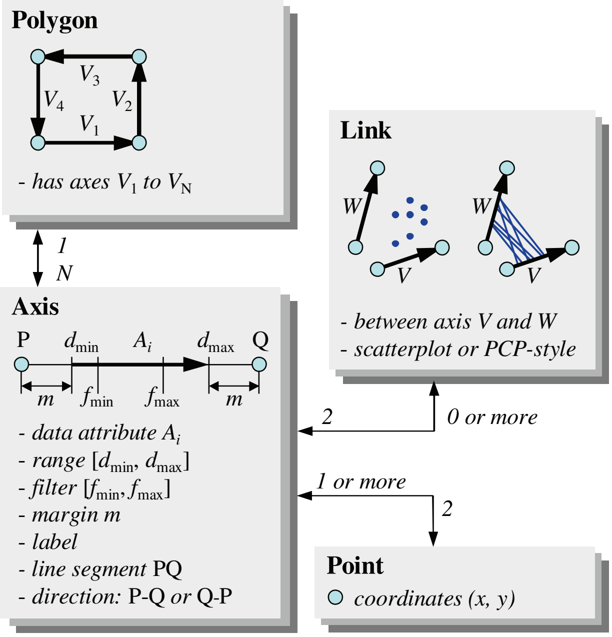


Fig. 3. Main elements defining a FLINAPlot.

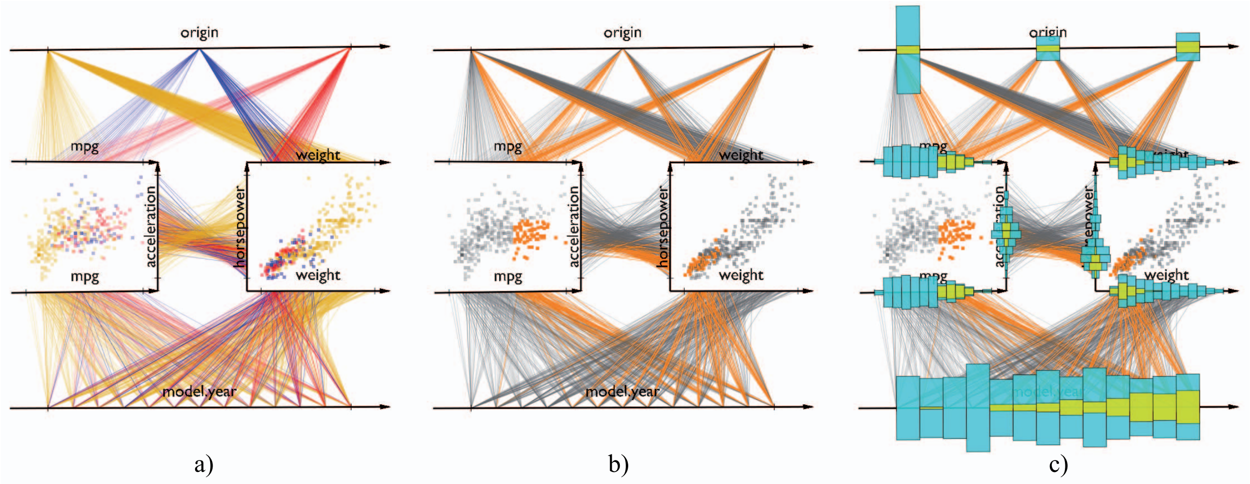


Fig. 4. FLINAPlots for the cars dataset: a) origin highlighted (yellow: USA; blue: Europe; red: Japan); b) selection of cars with fast acceleration (short time to reach 60 mph) and high mpg; c) histograms added.

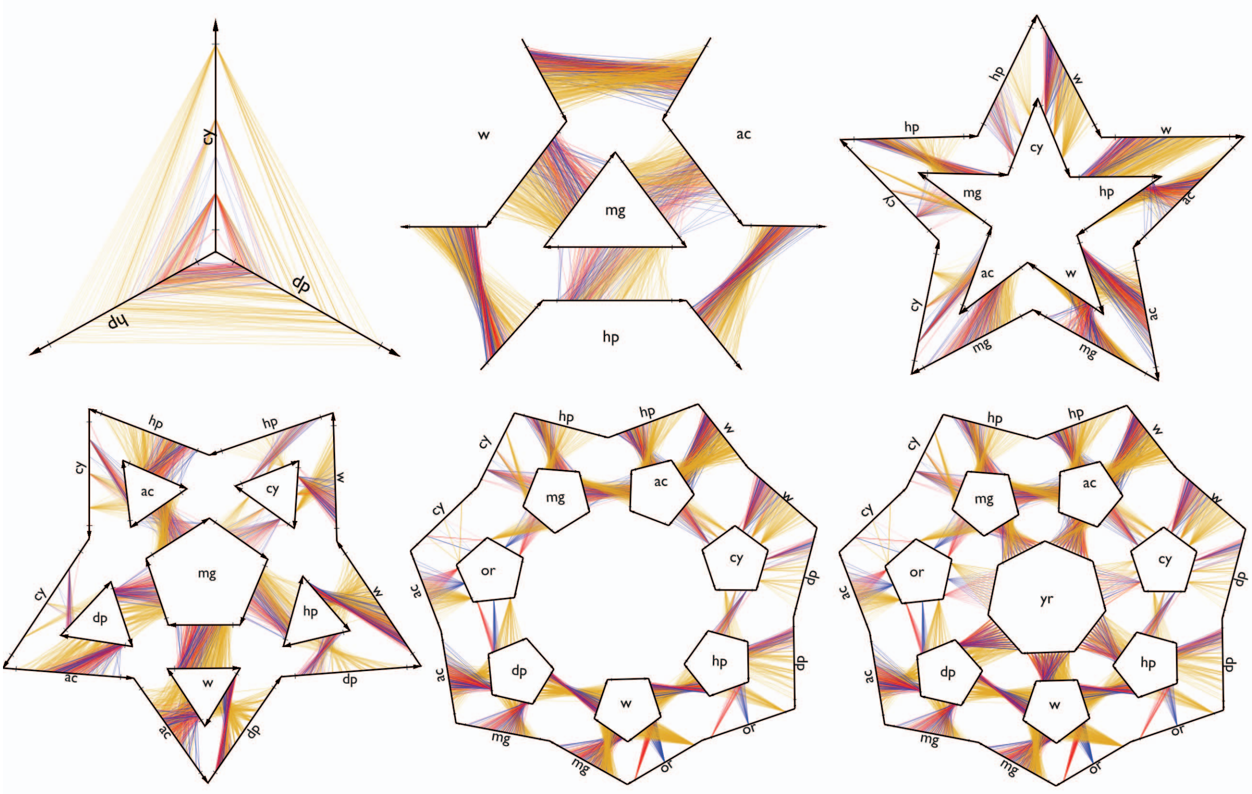


Fig. 5. Many-to-many relational PCPs. Top row: 3, 4, and 5 attributes; bottom row: 6, 7, and 8 attributes.

time on standard laptops. Hence, the user is enabled to perceive immediately the effect of interaction with the configuration and the items.

4 Examples

We have already given examples of FLINAPlots: the images shown in Figure 2 were all generated with our tool, which shows that standard as well as non-standard methods can be mimicked. Tools that have such visualizations built in can exploit particular properties of these methods, such as the fact that in PCPs all axes are aligned and positioned at equal distances. Also, they can provide dedicated interaction, such as the rotation of the TimeWheel. However, it is not that difficult to define FLINAPlots that resemble existing methods, and they provide much additional flexibility. In the following more examples are given.

4.1 Cars

The cars dataset [16] is a classic multivariate dataset. Suppose we are interested in cars with a fast acceleration (*i.e.*, the time to reach 60 mph should be low) that are economic in their use (*i.e.*, have a high value for mpg: miles per gallon). We make a scatterplot of acceleration against mpg, such that we can easily select subsets that meet our requirements. To understand the trade-off in designing a car, we make a second scatterplot, now showing horse power against weight, and add a PCP-style visualization of acceleration against horse power. Finally, for the spatiotemporal context, we also consider origin and year against mpg and weight. Results are shown in Figure 4. Via interaction much additional insight can be obtained, especially by highlighting and coloring subsets of interest. Splitting all cars based on origin

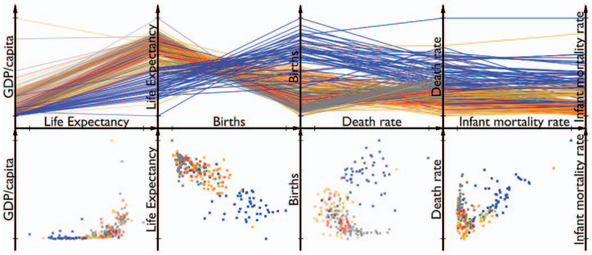


Fig. 6. Demographic data for different countries. Asia: brown; Africa: blue; North America: red; South America: green; Oceania: orange; Europe: gray.

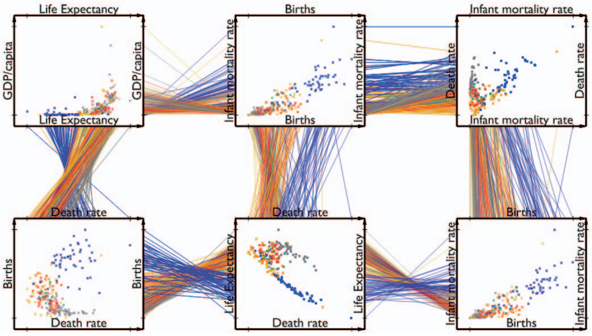


Fig. 7. Alternative lay-out for demographic data

shows for instance that American cars are heavier and have a lower mpg than those from Europe and Japan. Selection of our favorite fast acceleration and high mpg cars shows that these have low horse power and low weight. Showing histograms in addition indicates for instance that the trend is towards economic cars with good performance, and that Japan provides many of these.

4.2 Many-to-many PCPs

As mentioned in Section 2, the many-to-many PCPs of Lind *et al.* [14] were an important inspiration for us. The challenge is to define a minimal configuration of axes and links, such that all pairs of attributes are visualized in PCP-style. We used our tool to find solutions for this, results are shown in Figure 5. The solution for three attributes is trivial, the solution shown for four attributes is the one given by Lind *et al.*. For five to eight attributes we present new solutions. The last three solutions suggest that generic patterns might exist, such that many-to-many PCPs can be generated for arbitrary numbers of attributes.

4.3 Countries

One useful type of application of FLINAPlots is simply to combine scatterplots and PCPs. In Figure 6 we show visualizations of demographic data for different countries, obtained from the World Factbook[1] of the CIA. The PCP shows clusters for different continents across multiple attributes, and the difference between Africa and the other continents is striking. The scatterplots give more insight in the relations between two attributes, and show for instance that infant mortality is not the single explanation for a higher death rate. Figure 7 shows an alternative combination of scatterplots and PCP-style visualizations. The PCPs help to relate the different scatterplots.

4.4 Network monitoring

FLINAView was used to obtain new insights on a network monitoring dataset. An anonymized dataset was provided where the IP addresses were altered while keeping the network ranges, defined by subnets, together. The dataset consisted of three hours of flow data for SSH data, using the TCP protocol, consisting of 93,382 flows. A flow is a unidirectional connection between two network devices (hosts), the main attributes are the Source and the Destination, both described by IP address and port number; the StartTime and Duration; and the number of Bytes and Packages sent. One main challenge in network monitoring is to spot anomalous behavior. For this, it often suffices to consider flow data as multivariate data, and direct visualization of the structure of the network itself is not that relevant. There was no a priori information about the data and therefore the first step was to create a FLINAPlot showing base information on sources and destinations (Figure 8). Prior to actually exchanging data between hosts, TCP employs a handshake protocol where the hosts agree on setting up a connection. The source host requests a connection which the destination host, under normal circumstances, accepts. After a successful connection, two flows are sent, followed by the data to be transported, which is indicated by the higher value of the number of Bytes of the respective flow. Under these normal circumstances one would therefore expect a mirrored image between Source IP-Source Port and Destination IP-Destination Port. Figure 8 shows that the scatterplots of Source and Destination are almost identical. However, in the upper corners (highlighted in orange) it seems that there are many more flows from the higher range of source IP's with a high Source Port value. The distributions of the attributes reveal a related anomaly: one range of Source

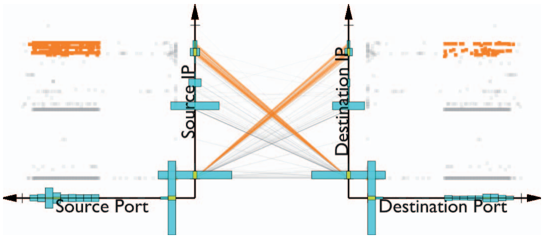


Fig. 8. Network data, showing distributions over attributes.

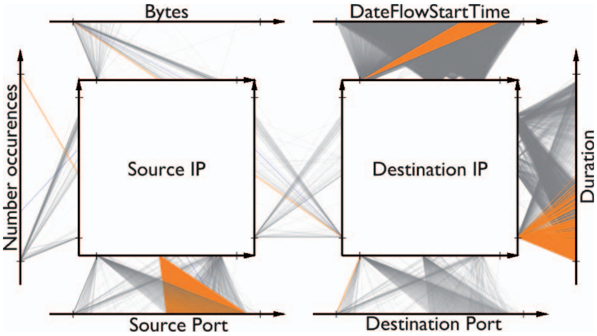


Fig. 9. Network data, showing more information.

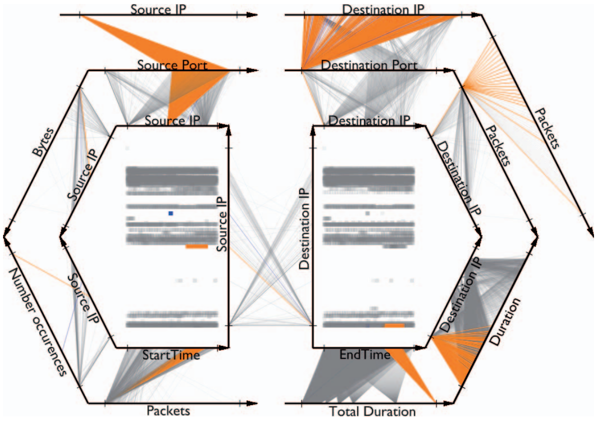


Fig. 10. Network data even more information. Some attributes are shown twice, over the full range and zoomed in (z).

[1] https://www.cia.gov/library/publications/the-world-factbook

IP's occurs often, as indicated by the relatively wide blue bar in the center for the Source IP.

To understand this strange behavior, we defined a new FLINAPlot (Figure 9) , where we showed more attributes. We calculate the number of occurrences per Source IP separately and add these as an additional attribute to the flows. This shows that there is one IP address, or range (highlighted in orange) that occurs far more often than others. We see that there is hardly any data transferred. The StartTime for these flows seems to be evenly distributed over a small range and the same holds for the source IP. The Duration attribute shows that most flows are terminated quickly. All this is peculiar behavior, which could be explained by options such as a keep-alive connection, peer-2-peer network traffic, or a network attack.

To understand which option applies, a third FLINAPlot was created, showing again more attributes (Figure 10). For several attributes we used two axes, where the ranges of secondary axes were tuned to depict all selected values along the axes. This gives much more information; for instance, it first seemed that there was just one Destination IP in the selection, but use of a smaller range for the Destination IP reveals an almost continuous range of Destination IP addresses. For the Source IP, however, there is just one IP address where the selected flows originate from. Also, the same Destination Port (22) is attacked. The total duration seems to show that for the selected flows this is the largest possible, although that is merely due to the amount of occurrences; almost one third of the total amount of flows. Figure 10 clearly shows what is going on: a port sweep. One external host tries to connect to all hosts within the local network to find devices an SSH server is listening. That information can be used to try to gain access to certain hosts within the network. Displays such as shown in Figure 10 can be used to improve the security within the local network. The Source IP trying to perform malicious activities within the network can be denied access, whereas the packets axis with the smaller range can be used to find out which local hosts might have been compromised.

5 USER STUDY

We conducted a user study to evaluate the usability of Flexible Linked Axes. Ten persons, nine male and one female, between the ages 23 and 39, participated. Their background was visualization (6), network monitoring (3), and logistics (1). All participants were experienced computer users. All had normal vision, except for one, who was color blind. Prior to the user study, two participants were unfamiliar with PCPs. All users were invited to bring in their own dataset.

We spend about an hour per person to present and discuss our approach. The study consisted of the following steps:

1. *Introduction*, to explain the reason of the study;
2. *First demonstration*, to show basic functionality;
3. *User exercise*, to make the participant familiar with the tool;
4. *Second demonstration*, showing alternative visualizations of the Iris dataset, showing more options and ideas for the use of the flexibility of the tool;
5. *Exploration user dataset*, participants work with their own dataset, trying to obtain new insights and to find out whether the flexibility offered is useful;
6. *Completion survey*, to get feedback about the tool and concept.

In the second step we showed how to create simple visualizations of the Iris dataset, such as a scatterplot, a PCP, and a radar chart. In the third step, the users were asked to answer seven simple questions about the Iris dataset using the tool. These questions were:

1) What is the range of values of attribute Sepalwidth?
2) Is there a correlation between Sepallength and Sepalwidth?
3) Which value occurs most often for Sepalwidth?
4) What is the average value for Sepalwidth?
5) Is there a correlation between Sepallength and Petallength?
6) Is there a correlation between Sepallength and Petalwidth?
7) Which attribute(s) lend themselves for classification?

All participants were able to answer the questions correctly, which indicated to us that the explanation was clear, that the participants were able to use the tool, and that they could correctly interpret the visualizations.

During the exploration of their own datasets, the 'thinking out loud' method was used. The participants started with standard visualizations: scatterplots and PCPs. After a while seven of them started to place the axes in a more flexible configuration. Interestingly, three different exploration approaches were used, about evenly distributed over the participants. Some worked *top-down*, starting with an overview and removing uninteresting information; some worked *bottom-up*, building up the visualization step by step; and some used an *in-between* approach, dynamically switching between adding and removing axes.

The survey presented a number of statements, where the subjects could indicate their agreement on a 5-point Likert scale: Strongly agree (++), Slightly Agree (+), Neutral (o), Slightly Disagree (−), Strongly Disagree (−−). The results were as follows:

	++	+	o	−	−−
FLINAView is					
- easy to use	3	6	1		
- easy to understand	3	6		1	
- useful	6	3	1		
FLINAView is a good base for					
- creating and displaying scatterplots	7	3			
- creating and displaying PCPs	6	1	2	1	
- visualizing multivariate data	6	3	1		
- investigating dataset characteristics	4	3	3		
The concept of Flexible Linked Axes					
- has added value over PCPs	7	3			
- has added value over scatterplots	1	7	1	1	

Furthermore, the subjects were asked what they considered to be the strong and weak points of the system. Positive features were the ability to quickly visualize correlations, linked among many attributes, in a single plot, having two different visualization techniques available. The main weak points concerned details of the system. We have implemented two modes, an axis-edit mode and a data-view mode, to disambiguate selections, but half of the users found this non-intuitive; extra features like logarithmic axes were requested; and one subject noted that it requires some training because it is a new way of thinking about a problem. We finally asked for additional comments. Reactions of three users were: *”Very nice system, having high potential, but 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 the tool, and appreciated its flexibility and approach.

6 DISCUSSION

We admit that the concept we present in this article is simple and straightforward. We do not introduce new visual encodings, subtle interaction methods, or integrate sophisticated approaches from machine learning. The concept is just to enable users to configure axes in a flexible way, and use these to define visualizations, consisting of a combination of scatterplots and PCPs. However, we argue that this simplicity is a virtue, and that its versatility is surprising.

A simple approach implies that it is easy to implement (though developing a good editor is more involved than developing a good viewer), and, more importantly, that it is easy to use. The drawing metaphor is familiar to most users, accustomed as they are to using drawing tools for other purposes.

Another strong benefit of our approach is that a variety of different visualizations can be used. Users can choose mainstream methods (scatterplots, PCPs, radar charts); experiment with less common methods, such as Hyperboxes, the TimeWheel, and different many-to-many PCPs; as well as invent custom visualizations that match their Attribute Relation Graph of Interest (ARGOI) as closely as possible.

Finally, the drawing metaphor enables the user to use multiple visualizations in an integrated way. Standard visualization tools force the user to view a single visualization, or multiple visualizations split over different windows. With our approach, all visualizations can be laid out on a single canvas. This is advantageous during exploration, where the canvas acts as a scrapbook for jotting down different views, and also for presentation, in order to define a tuned set of views that together shed light on a complex data set. Also, in our approach all visualizations are linked implicitly, and support for linking and brushing is offered in a natural way.

In summary, we think that Flexible Linked Axes provide a versatile, powerful and useful means to create a variety of useful visualizations of multivariate data.

6.1 Comparison

We next compare Flexible Linked Axes with standard approaches, such as scatterplot matrices, PCPs, and radarplots. In principle, all these can be emulated, hence our approach inherits their strengths and weaknesses. If many (say, more than 10-15) unknown attributes have to be explored, these methods fall short and additional means, such as for instance scagnostics [19] are needed. Multidimensional scaling enables better detection of clusters by searching for optimal projections. Such approaches could be added by calculating projected coordinates of points representing items in a separate preprocessing step and adding these as additional attributes.

Compared to standard approaches, Flexible Linked Axes provide more flexibility. This enables the user to define composite and non-standard visualizations, and to use the drawing space optimally for the task at hand. However, this flexibility comes at a price. Compared to standard approaches, where the structure of the visualization is hard coded, the user has to spend more effort; custom, dedicated interaction, such as reordering axes in parallel coordinates is not supported; and the user might create visualizations that do not make sense.

Concerning the effort needed, our experience is that this is acceptable. Indeed, more actions are needed, but these are simple and well-known to most users. Also, the metaphor of a graphics drawing package provides inspiration for more functionality to lighten the task of the user. We already included options to align and evenly distribute axes, to constrain angles to multiples of 15 degrees, to rotate sets of objects, to align endpoints to a grid, and to swap or cycle the attributes of selected axes. We offer polygons to structure axes automatically on a higher level. Similar functionality can be included to quickly generate PCPs and radar charts with multiple axes. More in general, user-defined templates that describe often-occurring patterns would be a very useful extension. The system does currently not detect configurations that do not make sense, such as collinear linked axes. Built-in constraints or rules could be introduced to detect or prevent such situations, however, we found that users can easily detect such cases themselves when they move axes around.

6.2 Future work

Our users found the concept promising and interesting, but also had quite some requests for more functionality. These include the use of logarithmic axes; control over tickmarks, color maps besides opacity to visualize densities, more options to control the dots used in scatterplots, color legends to explain colors used. Concerning interaction and manipulation of axes, Tominski *et al.* [17] provide inspiration on different types of interactive axes.

Many variations on PCPs have been developed in recent years. Techniques such as the use of curved lines [7], bundled lines to depict clusters [21], and random sampling to reduce clutter [5] could be integrated.

In our current approach, we offer scatterplot, PCP-style, and univariate histogram visualizations. More options, such as barcharts, would be useful. Furthermore, we presented a number of new many-to-many PCPs, an open question is if these can be generated and are useful for arbitrary numbers of attributes.

From a more conceptual point of view, an interesting challenge is how to deal with ARGOIs. We introduced these to reason about exploration of multivariate data and the support offered by current visualization methods. In the current version of our tool, they do not show up explicitly: the user defines axes and links them, thereby controlling the visualization of the aspects he is currently interested in. An alternative would be to enable the user to define ARGOIs explicitly, and automatically derive a suitable visualization.

Acknowledgments

We thank Aiko Pras and Anna Sperotto, University of Twente, for giving us a challenging problem, large data sets, and valuable feedback during the project. Furthermore, we wish to thank the participants of the user study for their constructive contributions.

References

[1] B. Alpern and L. Carter. The hyperbox. In *Proceedings 2nd IEEE Conference on Visualization (Vis'91)*, pages 133–139, 1991.

[2] R. A. Amar, J. Eagan, and J. T. Stasko. Low-level components of analytic activity in information visualization. In *Proceedings IEEE Symposium on Information Visualization (InfoVis 2005)*, pages 111–117, 2005.

[3] R. Becker and W. Cleveland. Brushing scatterplots. *Techmometric*, 29(2):127–142, 1987.

[4] C. Collins and M. S. T. Carpendale. Vislink: Revealing relationships amongst visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1192–1199, 2007.

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

[6] N. Elmqvist, P. Dragicevic, and J.-D. Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1141–1148, 2008.

[7] M. Graham and J. Kennedy. Using curves to enhance parallel coordinate visualizations. In *Proceedings Information Visualisation (IV 2003)*, pages 10–16, 2003.

[8] G. Grinstein, M. Trutschl, and U. Cvek. High-dimensional visualizations". In *Proceedings International Workshop on Visual Data Mining 2001*, pages 7–19, 2001.

[9] H. Hauser, F. Ledermann, and H. Doleisch. Angular brushing of extended parallel coordinates. In *Proceedings IEEE Symposium on Information Visualization (InfoVis 2002)*, pages 127–130, 2002.

[10] D. Holten and J. J. van Wijk. Evaluation of cluster identification performance for different PCP variants. *Computer Graphics Forum*, 29(3):793–802, 2010.

[11] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(4):69–97, 1985.

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

[13] D. Keim and H.-P. Kriegel. Visualization techniques for mining large databases: a comparison. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):923–936, 1996.

[14] M. Lind, J. Johansson, and M. Cooper. Many-to-many relational parallel coordinates displays. In *Proceedings Information Visualisation (IV 2009)*, pages 25–31, 2009.

[15] M. Novotny and H. Hauser. Outlier-preserving focus+context visualization in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):893–900, 2006.

[16] E. Ramos and D. Donoho. The 1983 ASA data exposition dataset: Cars. `http://archive.ics.uci.edu/ml/datasets/Auto+MPG`, 1983.

[17] C. Tominski, J. Abello, and H. Schumann. Axes-based visualizations with radial layouts. In *Proceedings of the 2004 ACM symposium on Applied Computing (SAC'04)*, pages 1242–1247, 2004.

[18] C. Viau, M. McGuffin, Y. Chiricota, and I. Jurisica. The flowvizmenu and parallel scatterplot matrix: Hybrid multidimensional visualizations for network exploration. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1100–1108, 2010.

[19] L. Wilkinson, A. Anand, and R. Grossman. Graph-theoretic scagnostics. In *Proceedings IEEE Symposium on Information Visualization (InfoVis 2005)*, pages 157–164, 2005.

[20] P. C. Wong and R. D. Bergeron. 30 years of multidimensional multivariate visualization. In *Scientific Visualization Overviews, Methodologies, and Techniques*, pages 3–33. IEEE Computer Society Press, 1997.

[21] H. Zhou, X. Yuan, H. Qu, W. Cui, and B. Chen. Visual clustering in parallel coordinates. *Computer Graphics Forum*, 27(3):1047–1054, 2008.