THE UNIVERSITY OF CALGARY

Establishing Graphical and Formal Relationships between Visualizations of Multi-Dimensional Data

by

Elena Fanea

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

CALGARY, ALBERTA

April, 2006

THE UNIVERSITY OF CALGARY

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled "Establishing Graphical and Formal Relationships between Visualizations of Multi-Dimensional Data" submitted by Elena Fanea in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE.

Supervisor, Dr. Sheelagh Carpendale
Department of Computer Science

Dr. Rob Kremer
Department of Computer Science

Dr. Elena Braverman
Department of Mathematics and Statistics

Date

Abstract

Multi-dimensional data has been visualized in many ways and many of these visualizations are excellent for a particular set of data or set of tasks. However, we can think of these individual visualizations as point solutions in the problem space of potential visualization for multi-dimensional data. In this research I explore the interrelationships between two multi-dimensional visualizations, namely Parallel Coordinates and Star Glyphs, and investigate the possibilities of describing these visualizations formally. Through both graphical means and my linguistic formalism, I show that these two multi-dimensional visualizations are related and, in fact, belong to the same family of visualizations. This insight lets me develop Parallel Glyphs, a 3D integration of these two 2D visualizations, along with a suite of new interactive exploration techniques. The same linguistic formalism is used to specify Parallel Coordinates and Star Glyphs and is then used to build the morphology and syntax associated to Parallel Glyphs and its available interactions.

Publications

Materials, ideas, and figures from this thesis have appeared previously in the following publication:

Elena Fanea, Sheelagh Carpendale, and Tobias Isenberg. An Interactive 3D Integration of Parallel Coordinates and Star Glyphs. In *Proceedings of the IEEE Symposium on Information Visualization 2005 (InfoVis 2005)*, pages 149–156, Los Alamitos, CA, 2005. IEEE Computer Society Press.

Acknowledgements

I would take this opportunity to thank all the people who have supported me in this endeavour.

First, I would like to thank Dr. Sheelagh Carpendale, my supervisor. This journey was only possible thanks to you. I am grateful that you believed in me even when I had a difficult time adapting to a new life and culture. You have been my mentor, and I will always be indebted to you for all I know and who I am now. I owe your persistent patience and effort the success of this thesis.

I would also like to thank my friends and colleagues in the Interaction Lab. Thank you, Torre, for all your help with OpenGL. If it had not been for your help, I would still be wondering why I could not see the fog. You showed me that I was in the middle of it. Thank you Floh, for such a great collaboration on the InfoVis'05 paper. You made me a LaTeX and Corel Draw fan, and I am grateful for your help with editing my thesis. Petra, I thank you for processing the video for InfoVis in such record time. Thank you for the help with proofreading my thesis, and for the positive energy you always spread around you. Thank you Mike for being there for me whenever I needed any kind of help. I will always treasure the memories I have of you. Charlotte, thank you for being such a great friend. I am happy that we keep in touch even after I left the lab. Nelson, thank you for staying by my side all this time. I enjoyed having my office and then my cubicle next to yours, so I could easily come and ask for help. You made our trip to InfoVis more pleasant, and you had the best advice for my defense, when you anticipated Sheelagh's question. Thank you, Russell, for inviting me and my family to your church, where we felt the blessing of the Lord and the genuine caring of your family. To all the other Smarties from Ilab, thank you for making grad school such a rich, and life changing experience for me.

A special thank you goes to my friend Floriana, who was not afraid to read a computer science thesis. Thank you for spending so many days encouraging me and helping me to find the right words, while I was writing this thesis. You helped me find balance in life and the strength to not give up.

To my "MSc cell mate" Julie, thank you for proofreading my first chapter and for making life at work more bearable.

Lastly, to my parents, I thank you for all the sacrifices you made to raise me. You showed me the path of knowledge, and you encouraged me to walk it. Thank you for making me a tough, disciplined athlete at a young age. This has taught me to fight to the bitter end, and to believe in myself beyond any limitations. Dad, thank you for our long walks and conversations about music, literature and science, thank you for broadening my horizon. Mom, thank you for all the skills you taught me that helped me survive and take care of my family. Thank you for watching me study, late at night, for so many years. I thank both of you for your endless, unconditional love.

Dedication

To Bogdan and Ovi

My dearest son and husband, who have sacrificed everything to support me, and have stood by me for better and for worse. I love you both, more than words can describe.

Table of Contents

List of Figures

Chapter 1

Introduction

We are often exposed to information with more than three attributes that cannot be represented by traditional charts or graphs. This information is defined as multi-dimensional data and can be conveniently stored in a tabular format. An example of multi-dimensional data analysis in everyday life is the familiar task of deciding on which car to purchase. To choose, we have to look at the brand, year of fabrication, price, colour, gas mileage, and number of doors, just to name a few attributes. The complex number of possible combinations makes the selection not an easy task. In order to facilitate multi-dimensional data analysis, many visualization techniques have been developed, each with its strengths and limitations.

This thesis began with looking at two existing visualization techniques for multi-dimensional data: Parallel Coordinates and Star Glyphs. The thesis has evolved into the integration of these two methods, into a new visualization technique named Parallel Glyphs. As a theoretical support, a linguistic formalism has been developed by analogy with natural languages, to provide tools capable of describing visual representations of multi-dimensional data. This capability of the linguistic formalism is illustrated by specifying the morphology and syntax of the visual languages that describe Parallel Coordinates and Star Glyphs, respectively. Furthermore, a unified morphology and syntax has emerged, by applying topological transformations on the previously defined morphological units. This unified visual language provides the formal foundation of Parallel Glyphs.

This chapter first defines the context and the scope of this thesis. The next section introduces Parallel Coordinates and Star Glyphs, along with a motivation for an associated formalism. Then the problems addressed in this thesis are identified. They are followed by the thesis' goals, showing how each problem is addressed. An overview of this thesis is provided at the end of the chapter.

1.1 Problem context and scope

The context and scope of the applied part of this thesis is pictured in Figure 1.1.

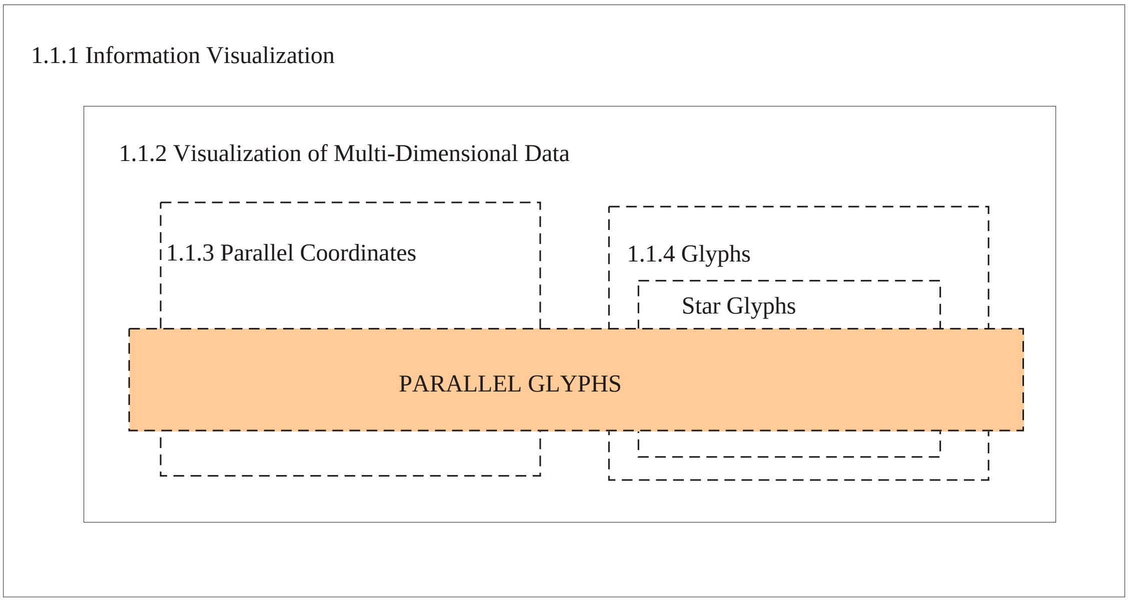


Figure 1.1: Context and scope of the applied aspect of this thesis.

While belonging to the wide discipline of Information Visualization, this research is specifically focused on visualization of multi-dimensional data. The inner rectangles, with a dotted border, represent the visualization techniques of multi-dimensional data that constitute the starting point of this research. Namely, they are Parallel Coordinates and Star

Glyphs, which are a particular case of Glyphs. The middle shaded rectangle represents the new visualization technique developed in this thesis, Parallel Glyphs. This technique employs and, more importantly, extends features of the pre-existing methods.

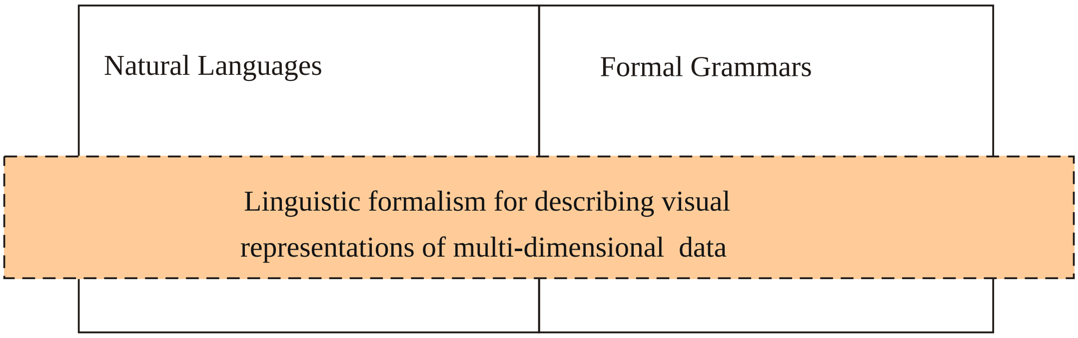


Figure 1.2: Context and scope of the theoretical aspect of this thesis.

Figure 1.2 illustrates the context and scope of the theoretical aspect of this research. The linguistic formalism associated to visual representations of multi-dimensional data developed in this thesis has its foundation in natural languages and formal grammars.

1.1.1 Information Visualization

It is well known that humans started creating visual representations of their surrounding world early in our history. Initially, pictograms were used for communication, and maps were used for orientation. In time, a well defined area of scientific interest based on visual representations has been developed. Horn [33], for example, presents representative stages of the history of visual techniques used by humans.

The science of Information Visualization has evolved out of necessity to better understand the information that we come across with in everyday life. Visual representations of this information are employed to retrieve facts that are not revealed by traditional representations. Chapter 2 will detail Information Visualization and discuss some of its specific techniques.

1.1.2 Visualization of multi-dimensional data

An important sub-area of Information Visualization concerns the study of the development of the visual representations of multi-dimensional data. As already mentioned, the concept of multi-dimensional data represents information with more than three attributes. Time series data, especially financial or biological information, are common examples of multi-dimensional data. They are often involved in decision making processes that require the simultaneous analysis of big sets of items, each item having a considerable number of attributes. Therefore, many researchers have oriented their research toward providing accurate and easy to use means of analysis of multi-dimensional data.

The visual representations of multi-dimensional data make use of shapes, colours, positions or other visual cues to facilitate tasks such as selection, comparison and pattern identification. Generally, multi-dimensional information is stored in a data table, but the tabular format may not always support the user enough to solve these tasks, especially to discover patterns. Therefore, the visual representations of multi-dimensional data extend the traditional representations, as they supply the user with tools for a visual analysis of information.

The research described in this thesis is confined to two of these visual representations, namely Parallel Coordinates and Star Glyphs. This choice is due mostly to the fact that the same dataset can be visualized with both of these techniques. Consequently, different aspects of the data are likely to be emphasized in each case. This raises the question whether it is possible or not to have the best of both worlds simultaneously.

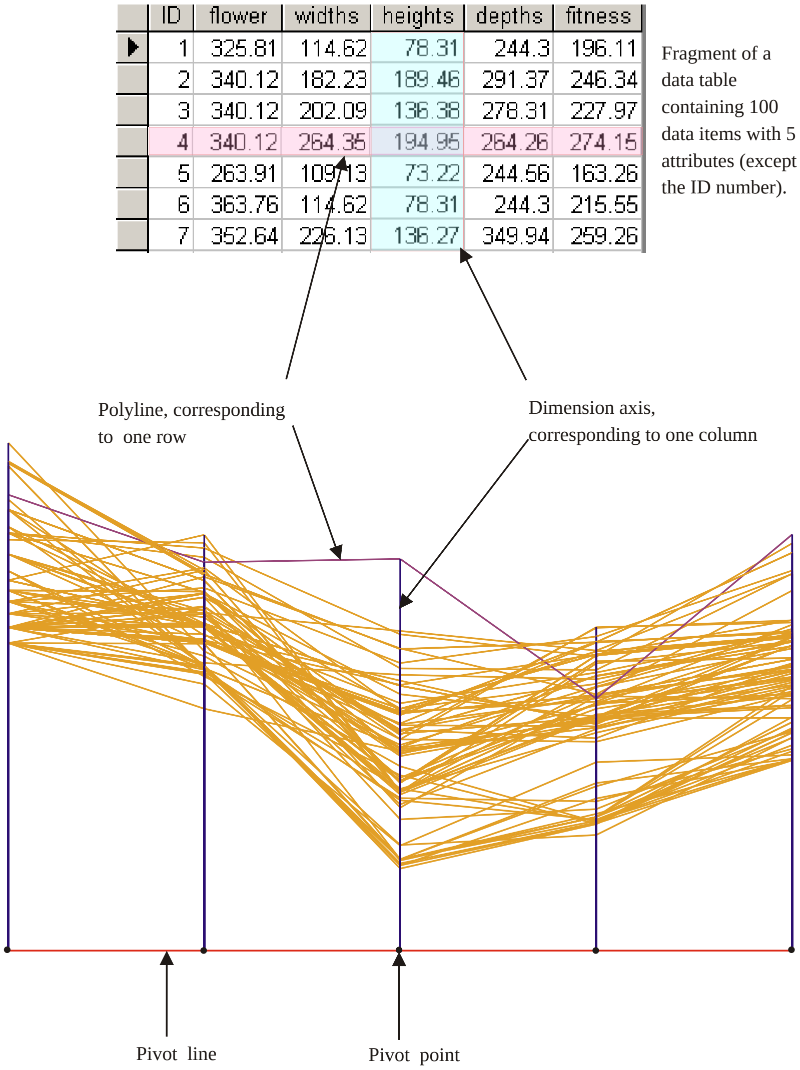
1.1.3 Parallel Coordinates

Parallel Coordinates is a 2D visual representation of multi-dimensional data, initially described by [39] and extended by other researchers. It uses a set of vertical axes for plotting the values of data items that are initially stored in a tabular format (See Figure 1.3). Each axis corresponds to one attribute of the data items, representing one column of the initial data table. These axes are called dimensions. The origin of each dimension is called the pivot point, and is situated on a pivot axis, common to all dimensions. For each data item, a polyline connects a point on each dimension axis that corresponds to the item's value for that dimension.

An advantage of Parallel Coordinates is that they offer a full overview of the dataset, allowing the user to identify trends or patterns. It is possible to plot a very large number of data items, on the order of tens of thousands (20,000 in [81], for example), while the number of dimensions that can be visualized is limited by the available space.

The main issue of Parallel Coordinates is the limited readability due to clutter and overlapping polylines. This fact reduces the advantage of having the entire the dataset displayed in the same graphics window, because some polylines will be fully or partially hidden by other polylines. The various 2D interactions that have been developed to address this problem will be later discussed in Chapter 2 section 2.3.

Displaying a large number of dimensions also reduces the visualization's readability, as the distance between the vertical axes decreases drastically. Because the dimension axes are, by default, equidistant, this distance is computed by dividing the width of the graphical window to the number of dimensions. Therefore, one approach to the problem of reduced readability can be to limit the number of dimensions displayed at one moment.



ID	flower	widths	heights	depths	fitness
1	325.81	114.62	78.31	244.3	196.11
2	340.12	182.23	189.46	291.37	246.34
3	340.12	202.09	136.38	278.31	227.97
4	340.12	264.35	194.95	264.26	274.15
5	263.91	109.13	73.22	244.56	163.26
6	363.76	114.62	78.31	244.3	215.55
7	352.64	226.13	136.27	349.94	259.26

Parallel Coordinates visualization of the above data table.
All the values stored in one column are plotted along one corresponding dimension axis
One polyline connects the points visualizing values of the same data item, stored in one row.

Figure 1.3: Parallel Coordinates.

On the other hand, restraining the number of data items displayed is not necessarily a solution for this problem because, as it will be shown in Chapter 3, even for a data set with fewer than ten data items, ambiguity could be induced by overlapping polylines.

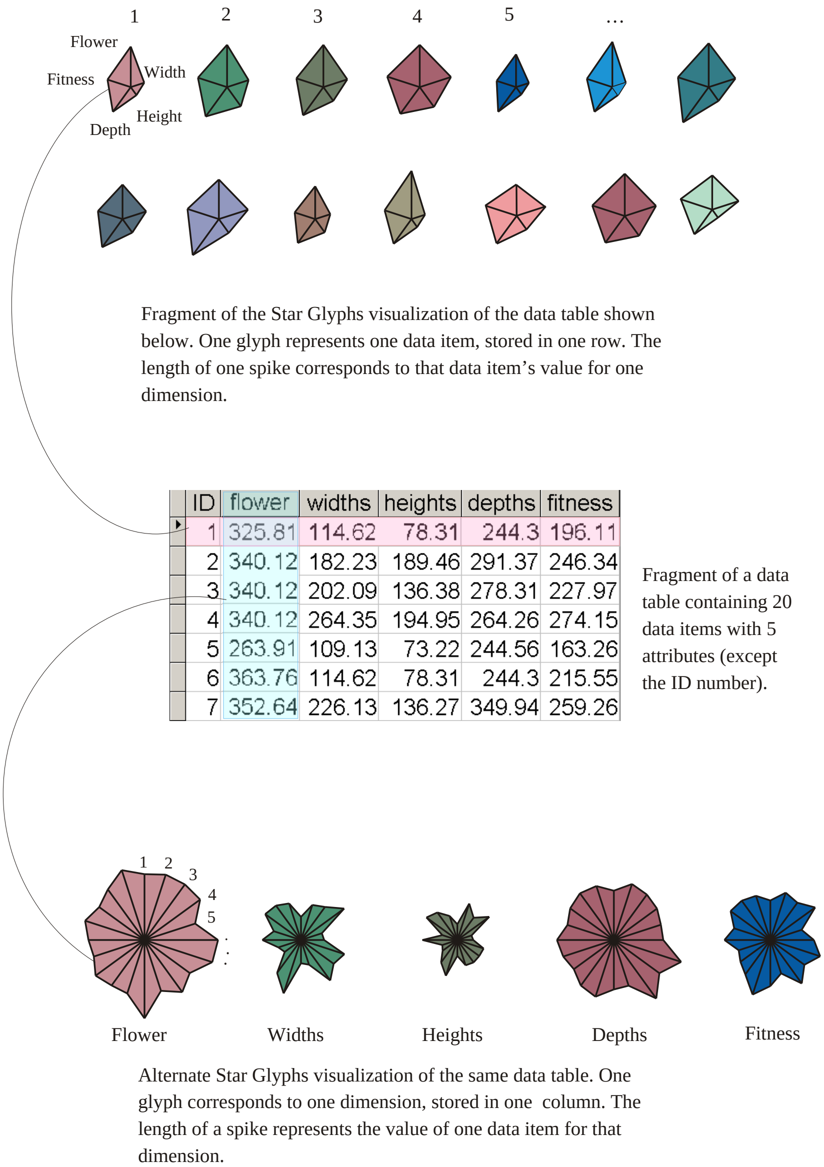
1.1.4 Glyphs

Glyph is a generic name given to a family of visual representations that convey the meaning of the data items through visual cues like shape and colour. Chernoff faces [16] are one example of glyphs. In this case, the values of each attribute of a data item are coded through size and shape of face elements, such as eyes, nose, mouth, or eyebrows.

Star Glyphs are a particular type of glyphs, a 2D visual representation of multi-dimensional data. Typically, one Star Glyph visualizes one data item, stored in a row of the data table. A Star Glyph is a polygon that bounds a set of spikes coming out of a common centre, arranged radially. If there is a Star Glyph per data item, then each spike corresponds to a different dimension. The length of the spike indicates the value of that dimension for the data item (See Figure 1.4).

In this thesis, this approach is extended into an alternative one, in which one glyph visualizes one dimension and each spike corresponds to a particular data item. Therefore, the length of a spike represents the value of the corresponding data item for that dimension (See Figure 1.4).

Star Glyphs are mainly used because, at some extent, the human eye distinguishes their shape accurately. The readability of this visualization can be increased even more by adding colours that convey information such as the identification number of the data item or of dimension, respectively. Unfortunately, even for a relatively small number of star glyphs, the limited display space would make it difficult to maintain an overview of the entire dataset without scaling the glyphs. This is a lose-lose situation, because on one hand not having the overview leads to an impossibility of performing comparison tasks. On the



ID	flower	widths	heights	depths	fitness
1	325.81	114.62	78.31	244.3	196.11
2	340.12	182.23	189.46	291.37	246.34
3	340.12	202.09	136.38	278.31	227.97
4	340.12	264.35	194.95	264.26	274.15
5	263.91	109.13	73.22	244.56	163.26
6	363.76	114.62	78.31	244.3	215.55
7	352.64	226.13	136.27	349.94	259.26

Figure 1.4: Star Glyphs.

other hand, maintaining the overview by reducing the size of the glyphs will decrease the visualization's readability.

1.1.5 Linguistic formalism for specifying visual representations

In Information Visualization, research is usually focused on developing visual representations and interaction techniques without establishing a solid formal foundation. One cause of this fact is the need of fast visual solutions to our increasingly complex life. A problem arising from this lack of formalism associated with visual representations is that it is easy to overlook relationships among these representations. Therefore, it might be worthwhile to build such a formalism even after developing the visual representation, to have a means of relating and comparing it to other existing or future visual representations.

An example and inspiration for a theoretical foundation of specifying visual representations is provided by natural languages. By definition [34], a natural language is ”a human written or spoken language, as opposed to a computer language”. In this thesis the concept of ”natural language” is limited to written languages. This choice is justified by the fact that visual representations are closer to the written languages than to the spoken ones, because the written word can be considered a visualization of the spoken word.

Horn has defined the visual language as the integration of words, images and shapes into a single communication unit [33]. While this may seem to be an informal definition of visual language, in this thesis it has constituted a basis for a linguistic formalism that specifies visual representations of multi-dimensional data. Visual languages describing Parallel Coordinates and Star Glyphs are built by maintaining an analogy of this linguistic formalism to natural languages.

Written languages are based on an ordered alphabet and have a series of morphological and syntactical rules according to which the words and sentences are built. Following these rules, it is possible to compare languages and identify families of languages, to learn a language from the same family, or to translate a text from one language to another. The discussion about how Horn's theory can be applied to build visual languages that specify information visualization techniques is presented in Chapter 4, as space is required to detail the morphology and the syntax of these visual languages.

Another source of inspiration for the linguistic formalism of visual representations is the area of formal grammars, which are also based on an alphabet, and have grammar rules established for their productions. For generative grammars, the rules determine how a new element is obtained by combining existing elements. However, the linguistic formalism defined in this thesis is closer to descriptive grammar, where the rules determine whether or not a particular element is grammatically correct.

1.2 Problem statement

While both Parallel Coordinates and Star Glyphs benefit the visualization of multi-dimensional data, they each have limitations, as just discussed. Also, there is no formalism that can be used to specify and compare them. Therefore this thesis addresses the following problems:

Problem 1. A multi-dimensional dataset can be visualized using either Parallel Coordinates or Star Glyphs, but it is not sufficiently explored whether these visual representations could benefit from each other.

- **a.** Parallel Coordinates are hard to read, while Star Glyphs are, comparatively, easily understood.

b. Vice versa, Star Glyphs have limitations that are well addressed by Parallel Coordinates. For example Star Glyphs do not scale well to large datasets, do not offer an overview of the entire dataset, and do not provide support for comparison of non-adjacent data items.

Whether the two visual representations can support each other while preserving their benefits, and still reducing their individual limitations, is a problem to be investigated.

Problem 2. There is no formal method to specify and assess visual representations of multi-dimensional data. With the proliferation of access to digital media, it is becoming increasingly common for people to present information visually. This has led to a myriad of new types of visual representations that frequently come into existence without an associated formalism. It is often difficult to retroactively fit a formalism to an existing visual representation. To the best of my knowledge, existing visual representations of multi-dimensional data have no formal foundation that supports their theoretical investigation and assessment.

Problem 3. The benefits of such formalism are not assessed. Once a formal method of describing visual representation of multi-dimensional data is defined, a series of questions rise. Will this formalism describe Parallel Coordinates and Star Glyphs? Is it applicable to other visual representations of multi-dimensional data? To what extent can it be generalized?

1.3 Thesis goals

This thesis will address each of the above mentioned problems, with the following goals in mind:

Goal 1. Enhancing visual representation of multi-dimensional data, by addressing both aspects of the problem:

a. Provide Parallel Coordinates with qualities that Star Glyphs have.

b. Provide Star Glyphs with qualities that Parallel Coordinates have.

To achieve this goal, the proposed method is to integrate Parallel Coordinates and Star Glyphs into one new visual representation of multi-dimensional data: Parallel Glyphs. This new visual representation interactively extends 2D Parallel Coordinates into the third dimension, connecting them with Star Glyphs. (Addresses problem 1)

Goal 2. Defining a formalism that provides a notation capable of describing visual representations of multi-dimensional data. Using an analogy to natural languages, I have built an alphabet composed of two types of ordered morphemes. With these morphemes several visual languages can be developed. The grammar foundation is described by associated morphology and syntax. Each visual language thus defined is capable of describing a family of visual representations. This capability has been illustrated by specifying the morphology and syntax necessary to describe Parallel Coordinates and Star Glyphs respectively. (Addresses problem 2)

Goal 3. Exploring the versatility and applicability of the formalism previously defined in Goal 2. A new visual language has been built, starting from the formalism that separately describes Parallel Coordinates and Star Glyphs. The syntax of the unified visual language is constructed by applying topological transformations on the previously defined morphological units. This constitutes the formal foundation of Parallel Glyphs.(Addresses problem 3)

1.4 Thesis overview

This thesis is structured into six chapters, as follows:

Chapter 2 provides a literature survey of existing research in areas related to visual representations of multi-dimensional data and to visual language theory. This will be done in a top-to-bottom approach by looking at information visualization in general and at visualization of multi-dimensional data in particular. A detailed overview of research related to Glyphs and Star Glyphs will be discussed next, followed by approaches related to Parallel Coordinates. The chapter concludes by looking at different approaches to visual languages and formal grammars.

Chapter 3 describes the design and implementation of Parallel Glyphs. This is an interactive 3D integration of Parallel Coordinates and Star Glyphs that utilizes the advantages of both representations to offset the disadvantages they have separately. The role of uniform and stepped colour scales in the visual comparison of non-adjacent items and of Star Glyphs is also being discussed. Finally, existing capabilities for focus-in-context exploration of the data using two types of lenses, and interactions specific to the 3D space are illustrated.

Chapter 4 presents the concepts necessary to build a linguistic formalism associated to visual representations of multi-dimensional data. First I have built an alphabet common to visual languages that describe visual representations of multi-dimensional data. Next, using this alphabet I have defined each morphology and syntax of the visual languages that describe Parallel Coordinates and Star Glyphs respectively.

Chapter 5 shows how via topological transformations these two visual languages can be unified. I have specified the morphology and syntax of this unified language, thus associating a formalism to Parallel Glyphs.

Chapter 6 states the summary of this thesis, reiterating the problems addressed and my contribution toward solving them. I conclude by identifying future research directions related to visual representations of multi-dimensional data and to their associated formalism.

Chapter 2

Related Work

This chapter discusses literature related to visual representations of multi-dimensional data in general, to Parallel Coordinates and Star Glyphs in particular. The objective of this review is to detail the background of the research contained in this thesis, providing the reader with the knowledge necessary to understand the content and motivation of the following chapters. First, the context of the thesis is explained. Next, different visual representations of multi-dimensional data are presented, along with specific interactions. The use of colours in general and colour scales in particular is then discussed. Afterwards, different aspects of visual language theory are presented. The chapter concludes with a summary of the related work based on which the research presented in the upcoming chapters has been developed.

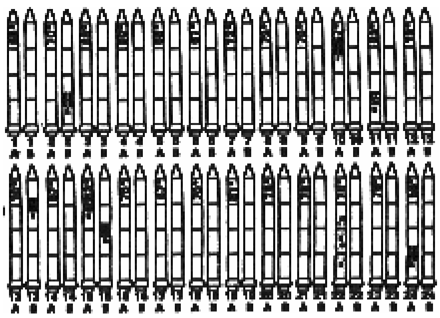
2.1 Context

The research developed in this thesis relates to information visualization and linguistic formalism. In Chapter 1 section 1.1 a succinct description of the context has been made. That will be detailed in the upcoming subsections. However, the discussion concerning the linguistic formalism is deferred until the the end of this chapter, as more space is needed. The first subsection describes Visualization. Next, Scientific Visualization is introduced as one branch of Visualization followed by the other branch, Information Visualization.

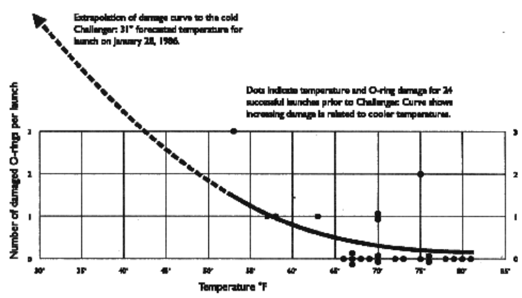
2.1.1 Visualization

Oxford dictionary defines visualization as the formation of mental visual images ([34], [2]). However, Ware [84] points out that this definition has been changed over time. This internal image, he says, has been translated to an external representation, used mostly for decision support. Visualization helps the decision makers to reveal information, relationships, and patterns hidden in the raw data.

A dramatic example of the impact a visualization can have on the decisions made, is the story of the Space Shuttle Challenger [79]. In 1986 all seven members of Challenger died when the shuttle exploded shortly after take off. The explosion was due to a failure of the O-rings. Unfortunately, the engineers who had analyzed the previous failures of these equipments had available a really poor visualization of the data (see Figure 2.1(a)). Therefore, they could not predict that the cool temperature would lead to this tragedy. Later, the same data was visualized by Tufte [79], who emphasized the relation between temperature and numbers of failures (see Figure 2.1(b)).



(a) Visualization of the O-rings failures used by engineers before the tragedy[76].



(b) Visualization of the same data, reflecting the relationship between temperature and O-rings failures [79].

Figure 2.1: Challenger Space Shuttle - O-rings visualizations.

Based on the type of data, visualization has been traditionally divided into two areas: scientific visualization (SciVis) and information visualization (InfoVis) [13]. According to this definition, scientific visualization is applied to scientific data, while information visualization is applied to abstract data. However, both areas have the same goal: to provide the users with tools for a visual data exploration and analysis. Nowadays the "separated but equal" approach to visualization's branches is questioned, whether it should be maintained or not. Arguments pro and counter are now based on more criteria, not only on the structure and type of data [65]. The type of spatialization is another strong decisive factor. This is due to the fact that scientific data has an intrinsic spatial dimension, while for abstract data a spatialization has to be assigned. The type of visualization pipeline also differentiates the two types of visualization: SciVis is focused on optimizing rendering algorithms, while InfoVis is more oriented toward improving perception and user interactions. However, the areas are often overlapped, in order to help the investigators get the most out the visualization of their data.

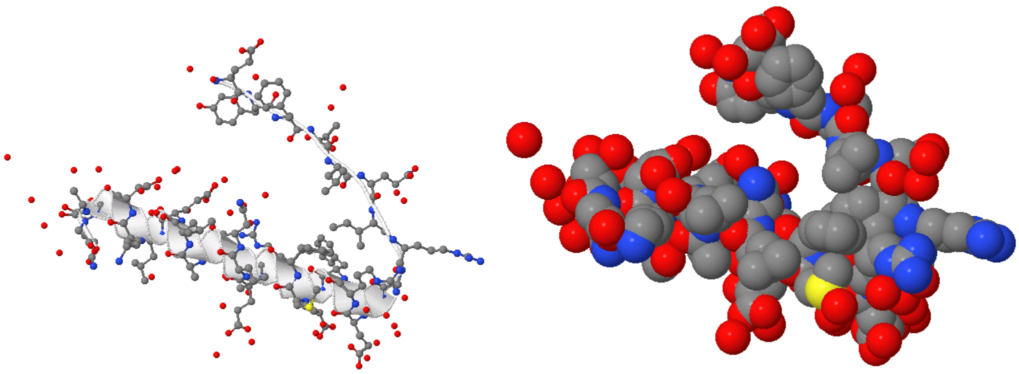
2.1.2 Scientific visualization

As mentioned above, scientific visualization uses data provided by different branches of science to reveal shapes or patterns that are not visible otherwise. Common sources of data are geographical, weather and genomic models. Simulations and experiments from environmental sciences or bioinformatics often provide large-scale data. Therefore, the scientific visualization techniques are often refinements of algorithms, with improved scalability. As many researchers in this community are from the graphics field, rendering alternatives and optimizations are also specific to SciVis.

An example of data with multiple visualizations is the protein p53. This protein is essential for suppressing tumours. Therefore, it is studied for its possible role in the fight against cancer. Various rendering of data describing this protein and its four domains are necessary for a better understanding of phenomena related to cancer [4] [3]. Figure 2.2 shows a sample of the text file containing the structure of the p53 tetramerization domain.

```
ATOM      1  N   GLU   326      14.783  14.947 -11.793  1.00 46.17           N
ATOM      2  CA  GLU   326      15.471  16.220 -11.447  1.00 39.29           C
ATOM      3  C   GLU   326      14.978  16.646 -10.075  1.00 37.04           C
ATOM      4  O   GLU   326      13.774  16.707  -9.841  1.00 37.72           O
ATOM      5  CB  GLU   326      15.133  17.290 -12.489  1.00 45.78           C
ATOM      6  CG  GLU   326      16.102  18.482 -12.553  1.00 71.24           C
```

Figure 2.2: Excerpt from the file describing the P53 Tetramerization domain.



(a) Ball and stick model.

(b) Spacefill model.

Figure 2.3: Examples of visualizations of the P53 Tetramerization domain.

Two models of visualization are shown in Figure 2.3. They illustrate the fact that the protein has an intrinsic spacialization.

2.1.3 Information Visualization

Information visualization is concerned with developing visual representations of abstract data. Looking at information visualization proceedings for the last five years, one can notice that the topics are divided into abstract categories, such as: graphs and trees visualizations, hierarchy navigation, multi-dimensional data, or time series data visualization.

Visual perception is an important objective of information visualization. Spatial position is a strong visual cue for perception. However, abstract data does not have a default position. Therefore, information visualization design principles coordinate how spatial positions are assigned in each visualization. Figure 2.4 presents an example of two different spatializations assigned to the same data.

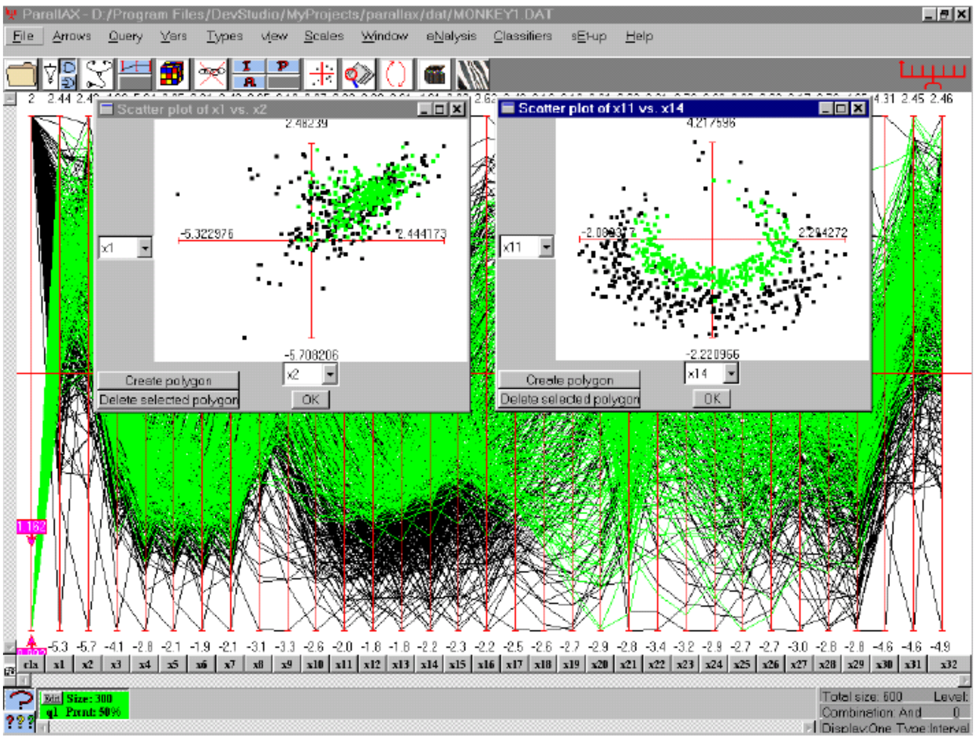


Figure 2.4: Parallel Coordinates and ScatterPlots visualization of abstract data [38].

Visual metaphors are the foundation of information visualization techniques. Researchers in this community strive to find new ways to look at data and new ways to interact with the visualization. Unlike scientific visualization, information visualization methods look mostly at relationships within the data set, not only at individual data values. Evaluations and comparisons of different visualization techniques allow information visualization researchers to determine the benefits and costs of different techniques. The feedback is then used to find new solutions and new directions in developing the visual representations.

One limitation of information visualization techniques is that they do not always scale well for the large data sets used in scientific visualization. This is one reason why information visualization methods are not always easily embraced by researchers outside the community. Avoiding learning a different visual technique is another reason. However, methods such as high dimensional data visualization would successfully support scientific visualization, if accepted.

Although scientific and information visualization seem to be two different areas, they are ultimately parts of the same science, visualization. The ultimate goal of researchers is to integrate the visualization techniques, in order to create tools for analyzing and understanding the data.

2.2 Visual representations of multi-dimensional data

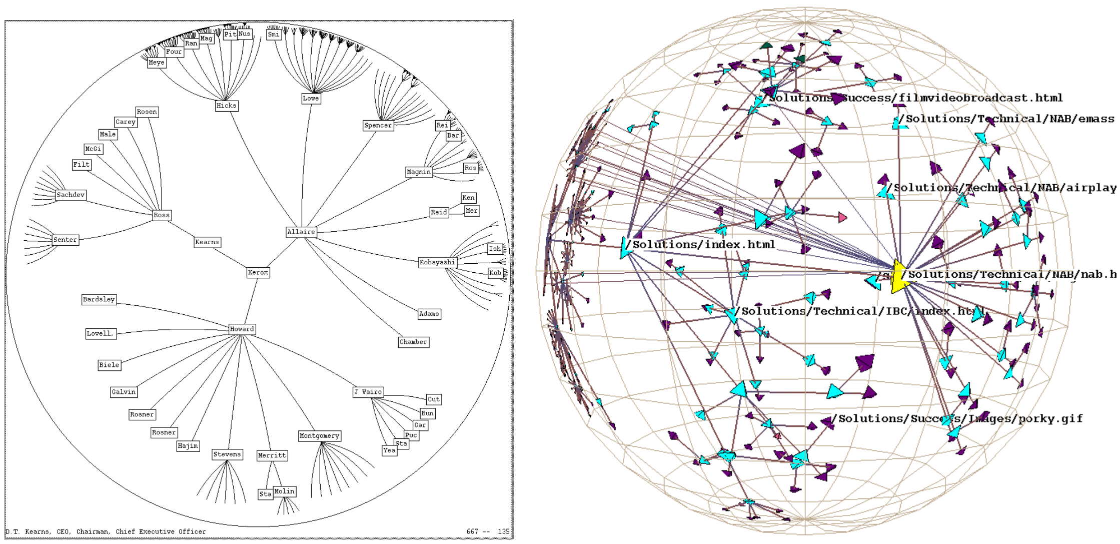
This section is focused on multi-dimensional data, that have been already defined in Chapter 1 section 1.1.2. The first subsection introduces some visualization techniques specific to hierarchical data in general. These techniques are important, because high-dimensional data is often structured in hierarchies, in order to reduce the number of dimensions. The next subsection presents an overview of visualization methods specific to multi-dimensional data. After that, Glyphs and Parallel Coordinates are detailed, as they are of particular interest for this thesis, being the starting point of this research.

2.2.1 Visual representation of hierarchical data

Hierarchical data is usually associated with organizational charts or directory trees, as in Microsoft Windows Explorer. The first goal of visualization techniques in this category is to correctly represent the parent-child relations among the data elements. Moreover, these visual representations aim to optimize the use of screen real estate, while maintaining a reasonable size of elements. In addition, the visualization systems need to provide sufficient tools for the exploration of data, according to Shneiderman's mantra "overview first, zoom and filter, then details on demand" [71]. Therefore, navigation and interaction features are necessary complements of a visualization technique.

The idea of a tree has inspired some visual representations. For example, a hyperbolic tree is used in [50] and [60] to visualize the node-edge relations (see Figure 2.5). The screen space is used by the hyperbolic visualizations more efficiently than by geometric graph visualization. Browsing techniques allow the user of the hyperbolic tree to focus on any particular element. Thus, a detailed analysis of each node is allowed by focusing on it and bringing it to the centre.

Another category of visual representations for hierarchical data is constituted by the space-filling techniques. The spacial arrangement divides these techniques into rectangular, and radial (or circular). Treemaps ([43]) was created in early '90s to visualize the structure of the directory tree. Successive horizontal and vertical rectangular subdivisions



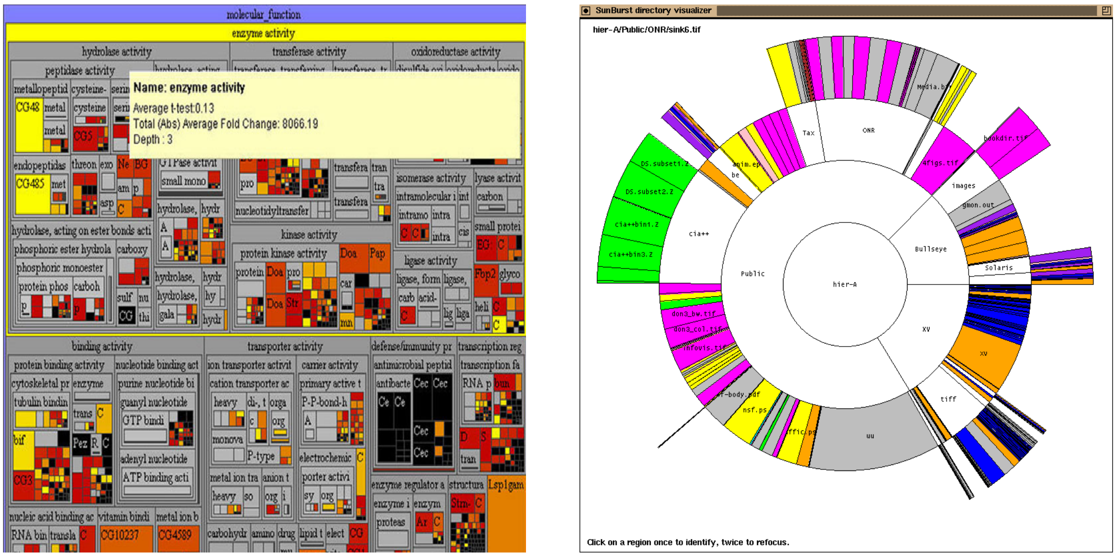
(a) hyperbolic browser [50].

(b) H3: 3D hyperbolic space browser [60].

Figure 2.5: Examples of visualizations of hierarchies in hyperbolic space.

of the screen space correspond to subtrees of the hierarchy. Over the time, treemaps visualization has been refined by adding features that improve the visual perception and the use of screen space [80], [12]. Beside the size of each rectangle, colours, labels and even sounds are mapped to different attributes of the data. Treemaps are now used by scientific visualizations as well, from oil production data to gene ontology (see Figure 2.6).

Early versions of radial visualizations of hierarchical data include information slices. This technique uses cascading, semi-circular discs [6]. The main problem of radial visualizations is that for large hierarchies, some slices become very small. Further improvements of these visual representations employ a full disc and focus + context navigation capabilities. Variations of Sunburst described in [74] address this problem of by adding exploration support (see Figure 2.6). Details of the item that receive the focus can be displayed either inside or outside the circular area.



(a) Treemaps - used to visualize microarray and gene ontology [8].

(b) Sunburst [74].

Figure 2.6: Rectangular and radial space-filling visualizations of hierarchies hyperbolic space.

InterRing [91] extends this approach to multiple foci. It provides the user with features familiar from Microsoft Windows Explorer, such as selection, drill-down and roll-up. In addition, rotation and structure-based colouring are available. Selection is based on the brushing mechanism of XmdvTool [81]. InterRing implements two types of distortions, circular and radial. These distortions are inspired by TableLens [63], and do not require additional space for focus + context display.

The rectangular techniques use all existing space, while the radial ones inherently leave some unused portions of the screen. However, comparisons of these two types of space-filling techniques ([9], [73]) reveal that radial techniques better support the user in exploring the hierarchic structure.

Unlike these 2D techniques, Cone Tree [67] uses the 3D space to visualize hierarchical data (see Figure 2.7). Perception is enhanced by features like animation and transparency of cones.

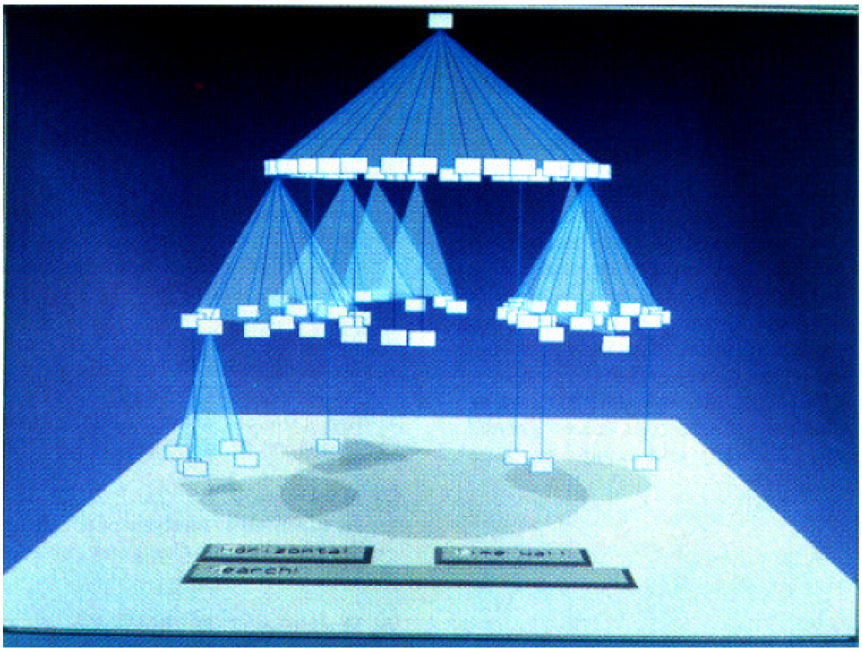


Figure 2.7: Cone Tree [67].

Time series data and the World Wide Web are typical examples of hierarchical data. One interesting technique is built to visualize the web evolution, which includes both categories of aforementioned data. The Time Tube [17] uses one Disk Tree for each period of time (e.g. one week) over which the website's evolution is recorded. From Figure 2.8) one can observe how a Disk Tree looks similar to a Star Glyph (later discussed in Section 2.2.3).

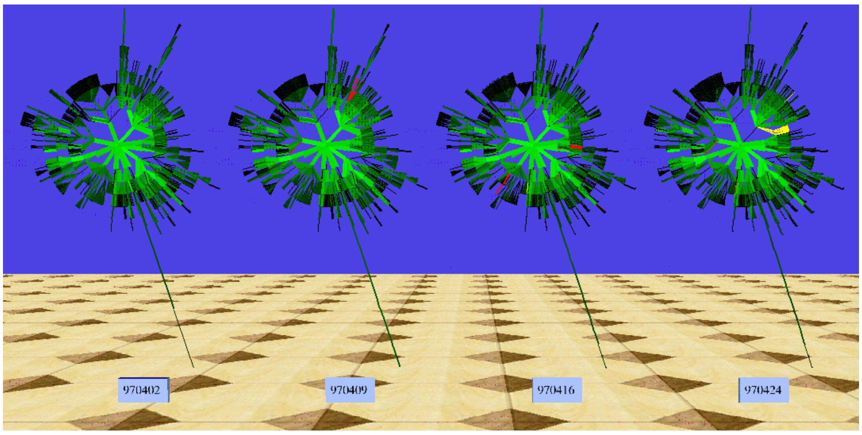
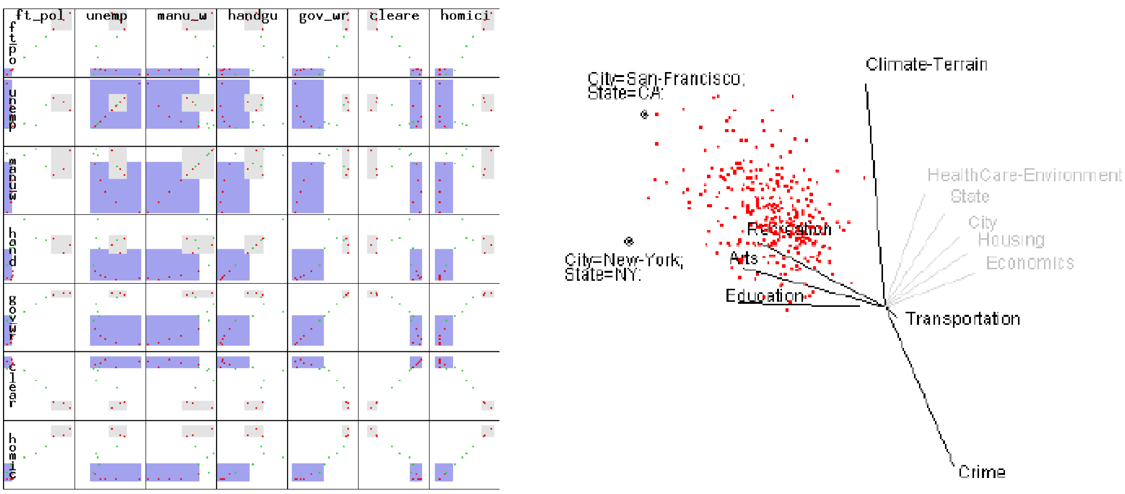


Figure 2.8: Time Tube [17].

2.2.2 Visualization techniques specific to multi-dimensional data

Even if the above visual representations are applicable mostly to hierarchical data, they are still important for multi-dimensional data. This is due to the fact that one common way of reducing the dimensionality is to create hierarchies. When partitioning multi-dimensional data and building the hierarchy, particular care must be shown when choosing the dimensions. Like TreeMaps and ConeTubes, Dimensional Stacking [42] is another example of stacked display. The core idea used by this technique is to embed one system of coordinates into another, starting with the most important dimension. A similar concept of stacked display is used by n-Vision [23], that introduces a system of nested heterogeneous coordinates. By using a physical input device named DataGloves, the user can interact with the visualization and choose points where new sets of three axis are to be nested. Therefore, there is no overview of the dataset, as the visualization would be too cluttered. Only the selected points are rendered deeper, and the last level can contain points, lines or shapes.

One of the most common visual representations of multi-dimensional data that projects the data on a 2D plane is the ScatterPlot (see Figure 2.9(a)). It uses $N*(N-1)/2$ pair-wise parallel projections that convey the relationships between pairs of dimensions. The projections are organized in a grid, for an easy visual association with the corresponding dimensions. Various interactions can improve this visualization, as shown in [26], [58], [61], [81], [82], [90]. Chalmers [14] projects the high-dimensional data on a low dimensional layout. His technique uses a stochastically based algorithm of linear complexity. To illustrate its efficiency, the algorithm is applied on bibliographic and time series data with tens of thousands of dimensions.



(a) ScatterPlot [81].

(b) Star Coordinates [45].

Figure 2.9: Examples of visual representations specific to multi-dimensional data.

Another projection of a multi-dimensional dataset onto the 2D space is Star Coordinates [45] that uses radial coordinate axes emerging from a common origin (see Figure 2.9(b)). In this visualization, each data item is represented by a point, whose location is computed based on all the attributes of that item. In addition, Star Coordinates provide the user with several interactions to support the visual analysis of correlations between dimensions and the identification of clusters, outliers, or patterns.

Johnson [44] proposed a vectorized generalization of Parallel Coordinates (which will be discussed in Section 2.2.4). The SBP (Single-point Broken-line Parallel-coordinates) algorithm uses vector-fusion to visualize the multi-dimensional data stored in a tabular format. The rows of the table are called vectors and the algorithm adds each component vector to the previous, until the entire row is summed up to a single-end-point resultant. This visualization is less intuitive than Parallel Coordinates and needs animation to increase the accuracy of perception.

2.2.3 Glyphs

A glyph is a symbol (as a curved arrow on a road sign) that conveys information nonverbally [2]. In information visualization, it represents a graphical object whose position, orientation, size, shape, colour, or transparency map attributes of a multi-dimensional data item. Glyphs are generally a way of representing many data values and are sometimes called icons. Because of the human eye's pre-attentive ability to distinguish shapes, glyphs are often used to visualize multi-dimensional data.

One of the first examples of glyphs are Chernoff's faces [16]. They map the attributes of one data item on different parts of the face, such as mouth, eyebrows, nose, or eyes (see Figure 2.10). Another common glyph is the arrow, often chosen to represent vector fields. The arrow depicts both speed and direction at a point [48]. Related research has followed two directions. One aiming toward identifying ways of generating glyphs and the other

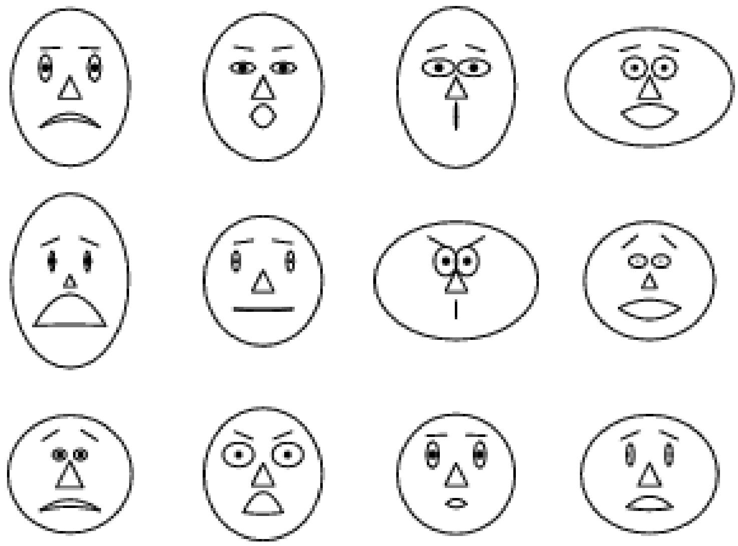


Figure 2.10: Chernoff faces [1].

toward improving the layouts. However, the final goal is still to convey information and relationships between data items.

There have been different approaches to generate glyphs. The solution proposed by Ribarsky et al.[66] is the Glyphmaker. This is an exploratory tool based on the human eye's capability to discern shape differences and spatial relationships. Glyphmaker provides the users with a glyph editor and a point-and-click mechanism that allows them to interactively customize their visualizations. This makes glyphs an effective multi-dimensional visualization.

Shaw et al.[70] use superquadrics to generate glyph shapes and explore their perceptibility. The order of the scalar values visualized is maintained by controlling the glyphs' curvature. Their study reveals that there are 22 distinct superellipsoid shapes. Therefore, changing the glyphs' curvature is a good way to generate glyphs. Forsell et al.[25] built on the same idea of surface curvature when proposing to construct 3D glyphs. In his view, ordinal data is depicted by using the sign of curvature of surface elements. The advantage of this solution is that, according to their study, affine 3D properties can be used in some 3D multi-dimensional data visualizations.

Distribution glyphs proposed by [18] support up to four attributes at a time of multi-dimensional, clustered data. Their main contribution is that they display statistical information such as distribution, variability, and extent information in both 2D and 3D. This solution helps to reduce the number of glyphs displayed, while retaining useful information. Their study reveals that distribution glyphs are as good as raw data, but in addition, they can answer statistical questions.

Beside these methods of generating glyphs there has been more research done exploring pixel-oriented techniques. Keim [46] introduced two such techniques. One was query dependent (e. g.: generalized spiral and circle-segments techniques) and the other, query independent (e. g.: recursive pattern technique). Yang et al.[89] also uses pixel-oriented techniques to condense the display of the data in a new multi-dimensional visualization technique named VaR. Each sub window within VaR, which is a glyph, uses a spiral arrangement of the pixels. These glyphs are positioned in manner that conveys relationships among dimensions.

Glyphs placement strategies are investigated by Ward [83]. The layout of glyphs is important to better convey relationships between data points. He categorizes approaches to glyphs placement in the existing literature into data-driven and structure-driven layouts.

A particular type of glyphs are Star Glyphs [66], [70], [81]. This is one of the two fundamental visual representations of multi-dimensional data that this thesis is based on, next to Parallel Coordinates. Star Glyphs are an axial representation, where the dimension axes are evenly distributed around a common origin. Typically, one star glyph corresponds to one data item. Therefore, the length of each axis represents the value of one attribute of that item (see Figure 2.11). As the way Star Glyphs are generated has not been changed so far, the only improvements concern the available interactions and the layout.

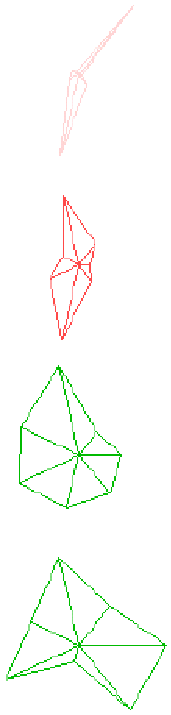


Figure 2.11: Star Glyphs [58].

An interesting set of interactions applicable to Star Glyphs, Parallel Coordinates, Dimensional Stacking, and Scatter Plots have been compared in [26], [58], [61], [81], [82], [90]. These interactions will be further discussed in section 2.3 of this chapter.

As for the improvement of the traditional 2D layout of Star Glyphs, Tominski et al. [77] suggest organizing them on a time based axis. This would provide a linked 3D layout for Star Glyphs with one glyph per data item. However, this idea was only left as a suggestion for future work. Kiviat diagrams [29] are 2D visualizations that resemble Star Glyphs in that radial spikes emerge from a common centre and each spike corresponds to a data item.

Kiviat tubes are obtained by displaying Kiviat diagrams along a time axis and rendering the surrounding surface as suggested by Tominski et al. [77]. The surface that connects all the Kiviat diagrams perhaps emphasizes the shape of the tube and possibly conceals information about individual data items. These ideas are a step forward from the traditional approach in that they link Star Glyphs, but there is no integration with Parallel Coordinates, as this thesis proposes.

2.2.4 Parallel Coordinates

Parallel Coordinates are a well recognized multi-dimensional data visualization whose systematic development began around 1978 [39], [35]. Their pioneer, Alfred Inselberg accentuates that this visual method yields "graphical representations of multi-dimensional relations, rather than just finite point sets" [39]. This fact is due to the rigorous geometry that stands behind Parallel Coordinates. The xy Cartesian plane is extended to the Euclidean N-dimensional space. The y axis of the Cartesian plane is multiplied into a system of N parallel axes, perpendicular on x. One point $C = (c_1, c_2, \ldots, c_N)$ of this space $\mathbb{R}^N$ is represented by a polygonal line that connects the points of coordinates c_i on the x_i axis (see Figure 2.12) [37], [85].

The number of dimensions and data items that can be visualized is theoretically unlimited. However, there is a practical limit defined by the computers's screen size and resolution. In addition, the number of data items is also limited by the degree of overlapping induced to polylines. Beside overdrawing, another limitation of Parallel Coordinates is that for categorical data, there is no inherent order of plotting the values along the vertical axes. This problem is addressed by Rosario et al.[68], whose Distance-Quantification-

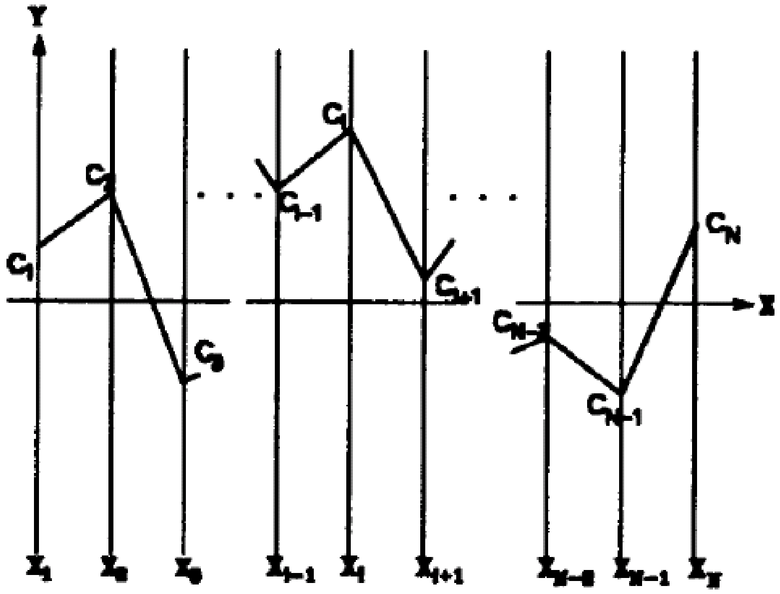


Figure 1: – Parallel axes for R^N.

The polygonal line shown represents the point
$C = (c_1, \ldots, c_{i-1}, c_i, c_{i+1}, \ldots, c_N)$.

Figure 2.12: Parallel Coordinates - definition [37].

Classing (DCQ) algorithm pre-processes the nominal data, before being imported into a visualization tool designed for numerical data, such as Parallel Coordinates.

The main strength of Parallel Coordinates is that they provide the overview of the dataset, thus allowing pattern recognition and relation searching [36]. Examples of how Parallel Coordinates helps data analysis by modelling relations within a data set, range from comparison of different economic sectors [36] to visualization of geographic spatial and temporal data [22].

Refinements of Parallel Coordinates include [51] and [24]. WinViz, described in [51], uses polylines to represent either tuples or clusters, while group bars are placed on the vertical axes instead of attributes values, in order to reduce the clutter. Inselberg, too, has extended Parallel Coordinates by defining convex hypersurfaces in $\mathbb{R}^N$ and studying convexity algorithms [40], [24]. However, this direction has not influenced the research presented in this thesis.

2.3 Interactions and other aspects of Star Glyphs and Parallel Coordinates

There continues to be considerable research extending Parallel Coordinates and Star Glyphs. One common direction is to improve the set of interactions possible with both these techniques [26], [58], [61], [81], [82], [90]. Beside interactions, other features that are not necessarily interactive, such as clustering and colour, can enhance the readability of these visualization. This section presents a series of interactions commonly applied on Star Glyphs and Parallel Coordinates, along with clustering techniques and colour scales. All these have influenced the interactions developed in this thesis which will be described in Chapter 3. These features support the user in accomplishing visual analysis tasks. They are particularly useful when the visual representation of large datasets is too cluttered.

First, clustering methods applied either on dimensions or data items are introduced. Next, brushing, dimension reordering and focus + context techniques are presented. The chapter continues with a discussion about how the use of colour can enhance the understanding of data. Finally, existing combinations of visual representations are mentioned.

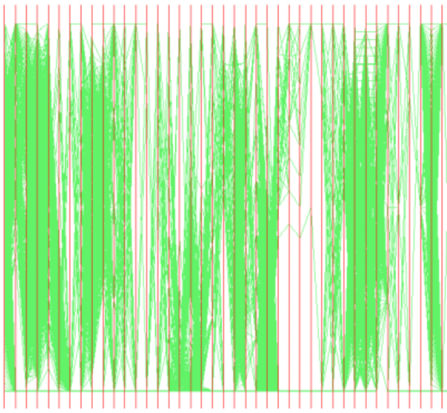
2.3.1 Clustering

Visual clutter is a common problem with both the Star Glyphs and Parallel Coordinates technique. It arises when the visualized data set is very large, having a great number of items or dimensions. If too many items are visualized, a Parallel Coordinates display typically results in visual clutter, since plotted lines overlap extensively. This makes the visualization difficult to read and aspects of the information are not conveyed. On the other hand, Star Glyphs visualizations can become cluttered because a large number of glyphs need to be displayed, as one glyph represents one item. In order to fit all glyphs in the screen space, the size of the glyphs and the distance between them have to be drastically reduced. Therefore, overlapping can occur, as in Parallel Coordinates.

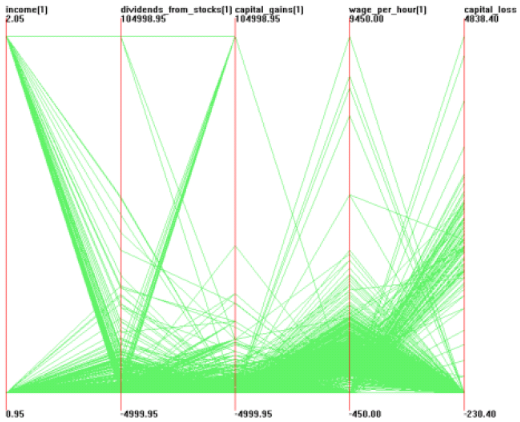
One approach to the mitigation of this problem is to use hierarchical clustering, which repeatedly groups dimensions based on user defined similarity measures (see Figure 2.13). Thus, this representation provides a multi-resolution view of the data that reveals trends at different degrees of summarization [47], [90], [92].

In the case of a high-dimensional data set, the dimensional axes of Parallel Coordinates are too close to each other to be properly read. On the other hand, in Star Glyphs visualizations, each dimension is represented by one spike of each glyph. Therefore, the angle between two adjacent spikes is inversely proportional to the number of dimensions. Consequently, for more than 8-12 dimensions the readability of Star Glyphs is very limited.

The same idea of cluster-based hierarchies can be applied to data items [26], [58]. Another technique, polyline averaging, represents a range of polylines with an averaged polyline. This method can be used for dynamically summarizing a set of polylines that in-



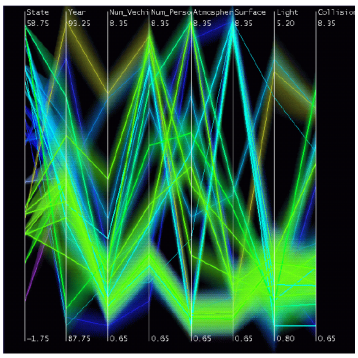
(a) Parallel Coordinates with 42 dimensions.



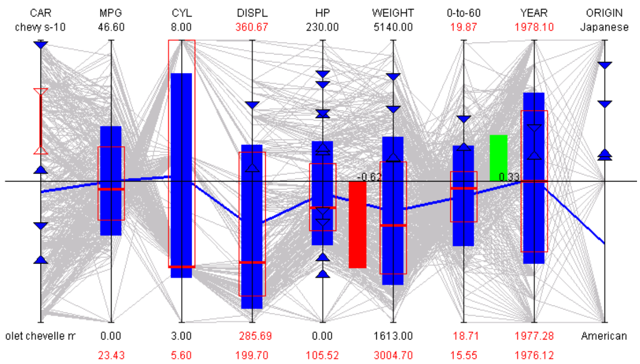
(b) Clustered dimensions in Parallel Coordinates.

Figure 2.13: Clustering of dimensions applied to reduce clutter in high dimensional Parallel Coordinates [92].

tersect the area interactively brushed on the display [72]. It reduces the clutter arising from overlapping polylines and is an alternative to more computationally demanding methods such as hierarchical clustering (see Figure 2.14).



(a) Hierarchical clustering of data items [26].



(b) Polyline averaging [72].

Figure 2.14: Clustering of data items in Parallel Coordinates.

2.3.2 Brushing

Advancements for Parallel Coordinates and Star Glyphs include brushing, to better support the interactive exploration of multi-dimensional data [15], [58], [61], [81], [82]. Brushing is a technique that allows for the interactive selection of a range of values directly from the visual representation via a click-and-drag technique. This is different from filtering, that allows selection via tools external to the visual representation, such as sliders. This selected range is useful for reducing clutter [90] and discovering answers for particular queries [7].

Trends in data can be revealed in Parallel Coordinates using angular brushing [30]. With this interaction technique the user specifies a subset of slopes to mark the corresponding data points as part of the current focus. The subset resulting from this brushing can be used for filtering the initial set. Thus, the user can choose not to display the items or dimensions that are relatively unimportant for a particular task [90].

For hierarchically clustered representations the XMDVTOOL implements *n*-dimensional brushing for interaction at a chosen level in the hierarchy [58, 81]. Angular brushing can also be done automatically or in relation to the hierarchy, if one is established [30].

2.3.3 Dimension reordering

A common interactive technique for reducing clutter and revealing data relationships in Parallel Coordinates is the manual or automatic reordering of the dimensions [61], [81]. While manual reordering may be more tedious than automatic reordering, it is often preferred because it allows for the establishment of a superior reordering [90]. Specific aspects of the data can be taken into consideration that might be overlooked by an algorithm that reorders based on similarity measures. Dimension reordering is also applicable to Star

Glyphs as each spike in a glyph represents a dimension and adjacent spikes can be easier to compare [90].

2.3.4 Focus + context

Focus + context methods are general techniques developed to address problems of visibility of details and have been applied to Parallel Coordinates and Star Glyphs. These detailed views can be provided by expanding or contracting the spaces between dimensions [90]. This can be important when the number of dimensions in the data set leads to a particularly dense display [91]. Multi-focus distortion operations allow the user to enlarge or shrink several regions in the same presentation [92]. The space between dimensions can be increased or decreased with a simple click. If, with high-dimensional data, the space between dimensions is too tight to allow an accurate click, a structure-based distortion called INTERRING can be used [91]. In an automatic application of distortion, the spacing parameter is the distance (for Parallel Coordinates) or angle (for Star Glyphs) [90]. Although the spacing between dimensions in Parallel Coordinates and Star Glyphs is typically uniform, non-uniform variations therein can convey additional information about the dimensions [90].

2.3.5 Use of colour

Maureen Stone considers colour as a key component of information visualization [75]. However, if colour is abused, the result is visual chaos. Therefore, the research related to colour aims to identify rules and guidelines for the best way of conveying information via colour. One important contribution is made by Tufte in [78]. He emphasizes that the first rule in using colour should be “do not harm”. The main functionalities of colour identified by Tufte and later illustrated also by Ware [84] and Stone [75] are: to label, to quantify, to indicate shape and size, to emphasize and to cluster.

In order to effectively communicate the information to everybody, possible colour vision deficiencies must be taken into consideration. Therefore, Stone [75] recommends to maintain sufficient contrast, so, even if reduced to shades of grey, the visualization is still correctly interpreted. Another helpful idea is to reinforce the colour encoding with other visual cues, such as position, shape and size. Ware's suggestion is to avoid colour scales based on the red-green channel, because of the possible users' colour blindness [84]. Beside this deficiency, he also points out that in general the human eye has its limitations. Therefore, he estimates that only 6-12 colours can be rapidly perceived. A study conducted by Healey [32] suggests to use up to seven isoluminant colours simultaneously, to maintain a rapid and accurate identification. However, in a collection of colours, individual colours are not always equally distinguishable. For choosing effective colour for data visualization, he specifies at least three criteria: a minimum colour distance, linear colour separation, and colour category differentiation [31].

The grey scale is often used, for its simplicity and natural sense of order. However, Levkowitz and Herman [53] underline that it has a limited perceived dynamic range of only 6-90 just noticeable differences (JND). Therefore, they built a locally optimal colour scale to maximize the JND between neighbouring pairs of colours. They also identify the desired properties of a colour scale. Order is one of these properties, meaning that the colours composing the scale should be perceived as preserving the order of values. Stone [75] notices that there are no intuitive hue scales, only learned ones. Therefore, colour scales based on variations of hue do not have an intuitive order. However, a rainbow scale, ranging from black to white through all the hues of the rainbow is proposed by Levkowitz and Herman [53]. This scale is improved though, by varying the lightness of its components. Another property of colour scales is uniformity and representative distance. This means that colours should convey the distances between the values represented. Colour scales should also not create perceived boundaries. Therefore they should continuously represent continuous scales. Stepped colour scales used in Chapter 3 do not have these properties, so they are not discussed in [53]. Ware [84] identifies another criterion for the evaluation of colour scales. He notices that colour scales in the yellow-blue channel can also create perception problems if the objects are too small.

2.3.6 Combinations of visual representations

Several systems use Parallel Coordinates along with other visual representations. For example, Wong et al. [88] developed a 2D visualization system that contains Parallel Coordinates and Scatter Plots, each presented in their own regions of the window. The representations are linked so that interactive changes to one are reflected in the other. The two techniques balance each other in data exploration.

The various types of interactions possible with Parallel Coordinates, Star Glyphs, Scatter Plots, and Dimensional Stacking have been applied by several authors simultaneously on each of these four visualization techniques [26], [58], [61], [81], [82], [90]. However, the goal of these combinations is to show how the interactions work on each visualization. There is no comparison of the visualization techniques or discussion about whether or not they can complement each other.

Finally, Wernert et al. [86] make an exploration of alternate multi-dimensional data layouts but these representations are not linked either visually, structurally, or interactively.

2.4 Visual language theory

An early definition, given by Lotman [55], states that language is "any system of communication which uses signs in a particular way". Visual language, in particular, is defined by Horn [33] as "an integration of words, images and shapes into a single communication unit". Although he does not talk about a visual language, McCloud's [59] analysis of comics is an inspiration for Horn. Comics, McCloud says, are a sight-based medium, created with visual iconography and words, which are totally abstract ideas. He suggests to analyze comics by separating the form from content, as cartooning is a form of amplification through simplification.

Horn expands McCloud's idea of analyzing comics. He claims that comic books, scientific diagrams and visualizations, and many other forms of communication involving visual representations converge into a "visual language". The possibility of being analyzed linguistically is one criterion for supporting the idea that the visual language is in fact a language. Horn defines the units of the visual language based on size: icons, concept diagrams, information graphics, and information murals. The problem with choosing these units is that they cannot constitute an alphabet in the sense defined by the English Oxford Dictionary [34], because they are neither finite nor ordered.

The morphology of the visual language, which is the study of the primitive components, has seen different approaches over time. One of them is illustrated by Bertin [10] and Saint-Martin [69] and has a formal type. Saint-Martin [69] defines a taxonomy of visual primitives, making a distinction between open (linear) and closed forms, each having a series of subclasses of elements. Bertin [10] considers graphics a "language for the eye". Therefore, he explores the factors that influence perception and identifies the following visual morphological variables that he calls "retinal variables": 2D position, size, value, texture, colour, orientation, shape. Although they are discussed in the context of cartography, these variables are also valid in terms of information visualization and visual language. Saint-Martin's set of 2D primitives is extended by Biederman [11], who identifies 36 3D primitives he called "geons" (geometric icons).

However, this formalist approach to morphology is not appropriate for Horn's perspective on visual language for different reasons. Bertin, for example, does not take the integration between words and images into consideration and, therefore, does not follow Horn's definition of a visual language. Saint-Martin's primitives, on the other hand, are too abstract and low-level. Hence, they are difficult to use for building meaningful units. For example, a tree defined as a group of lines and curves at specific angles is hard to be identified as a tree. In response, Horn follows a utilitarian approach to defining the morphological units of visual language. His categories of morphological primitives are words, images, and shapes (points, lines, abstract shapes, and the space between shapes). The properties that characterize these primitives extend the set of variables identified by Bertin with 3D position, motion, thickness, and illumination. His approach to morphological primitives also builds on Goldsmith's [27] definition of unities, as areas of a picture that can be recognized as having a separate identity.

The combinations of the morphological units are studied by syntax. The rules for these combinations depend on at least two factors. First, the topological relations established between the primitives existing in the visual field [69]. Second, the Gestalt Principles of proximity, similarity, common region (closed forms), connectedness, good continuation, and closure [69] [33].

A different direction of research related to visual languages is focused on defining formal grammars that can later be used as a basis for developing visual programming languages. Visual languages are defined by Marriott et al. [57] as sets of diagrams that have been defined as valid sentences. They often involve both generative and analytic aspects of formal grammar.

Generative grammars define a set of rules for generating all the elements of the language, starting from an initial element. Chomsky [20] introduced the definition of a formal grammar. Furthermore, the language of a formal grammar $G = (N, \Sigma, P, S)$ is defined as all those strings over Σ that can be generated by starting with the start symbol S and then applying the production rules in P until no more non-terminal symbols belonging to N are present. A grammar is context-sensitive if all the production rules determine if a string β can be correctly generated from another string α, based on the context of α. This definition derives from the notion of context-sensitive grammar introduced by Chomsky [19] to describe the syntax of natural languages. This term reflected the fact that using a particular word was grammatically correct or not in that natural language, depending on the context. If a language is not context-sensitive, it is called context-free.

Analytic grammars assume that the language has been already generated and then analyze whether an arbitrary input string is grammatically correct. They formally describe a parser for a language.

It has been suggested that there are three main approaches to the specification of visual languages: grammatical, logical and algebraic [21] [57].

The grammatical approaches are based on string rewrite mechanisms. They have an initial structure, an alphabet, and a set of rewrite rules. L-systems [54] [62] are an example of these approaches that can generate complex structures based on rewriting rules. An alphabet, and a set of productions are defined. Productions are the rewriting rules for the individual modules over an interval of time. An L-system development has an initial structure or 'axiom' and is mainly used for describing recursive structures. Rekers and Sch*ü*rr [64] underline the need to complement the spatial relation graph with an abstract syntax graph. They not only use the graph grammar as syntax definition for formalism for visual languages, but also provide a graphical parsing algorithm for this grammar.

The logical approach uses logic formalisms from mathematics or artificial intelligence. Haarslev [28] is an example that uses artificial intelligence description logic theory to combine topology and spatial relations.

A high-level framework for the definition of visual programming languages is presented in [56]. The layout perspective of the spatial relationships in that formalism is extended to a spatial graph grammar that introduces spatial constraints to the abstract syntax in [49] using algebraic specifications of composing functions to define and compare graphs.

The linguistic formalism described in this thesis does not follow this direction of formal languages. It only relates to the grammatical approaches in that it has an alphabet. However, it differs in that it has no initial axiom and rewrite rules. Also, while algebraic formalism is used, the linguistic formalism described here does not rely on the composition of multiple functions to define a grammar. Instead, it follows a closer analogy to natural language and to Bertin's [10] and Saint-Martin's [69] approach for defining morphological units. Saint-Martin defines a visual language that communicates through painting, having then coloremes as primitives. Similarly, the visual languages defined in this thesis use different information visualization techniques as means of communication. Therefore, the morphemes and morphological units defined here as primitives are specific to these techniques, with the addition that they comply with Horn's [33] requirement to integrate shapes and text.

2.5 Conclusion

This chapter has presented a summary of relevant literature concerning existing methods of visualization of multi-dimensional data and formal grammars. Two such visualizations, Parallel Coordinates and Star Glyphs, have been analyzed in detail. They will be the basis for the new visual representation that will be introduced in Chapter 3. Along with these visualizations, some of their associated interactions, such as clustering and brushing have also been described. The role of colours in conveying information has been discussed next. The chapter has concluded with an overview of the visual language theory.

Chapter 3

Parallel Glyphs

This chapter presents Parallel Glyphs, an interactive integration of the visual representations of Parallel Coordinates and Star Glyphs that utilizes the advantages of both representations to offset the disadvantages they have separately. Parallel Coordinates are a powerful method for visualizing multi-dimensional data, but when applied to large data sets, they become cluttered and difficult to read. Star Glyphs, on the other hand, can be used to display either the attributes of a data item or the values across all items for a single attribute. Star Glyphs may readily provide a quick impression; however, since the full data set will require multiple glyphs overall readings are more difficult. Parallel Glyphs extend 2D Parallel Coordinates into the third dimension and naturally connects them with Star Glyphs. This shows that, in fact, Parallel Coordinates and Star Glyphs are orthogonal to each other and that they belong to a continuum of visualization techniques. This new technique can be enriched by applying colour scales to the 3D Parallel Glyphs to support comparison and selection tasks in 3D. In addition, new interaction methods that are possible in 3D, such as a ring ruler, enhance comparison of data values and can lead users to new insights into the data.

The remainder of this chapter is organized as follows: Section 3.1 explains Parallel Coordinates and Star Glyphs in detail, along with their advantages and disadvantages. Next, the concept of Parallel Glyphs is introduced and the way they are constructed is shown. Afterwards, I discuss different features of Parallel Glyphs that improve their readability, such as use of colour scales, new interaction techniques and two focus + context methods using lens. Finally, a short case study is presented to illustrate the application of this new technique on a specific data set. The chapter concludes with a summary of Parallel Glyphs and their capabilities to support the user in visual analysis of multi-dimensional data.

3.1 Introduction

As already presented in Chapter 2, the Information Visualization community shows a wide interest in developing and improving multi-dimensional data visualization techniques. Researchers are motivated in their endeavours by the benefits of these techniques that provide bridges between the original data set format and representations of the same data that are easier to perceive. As mentioned in Chapter 2 Section 2.3, several interactive methods have been developed with the focus on facilitating rapid, qualitative understanding of multi-dimensional data.

Multi-dimensional data is composed of many multi-dimensional data items that each have values over a series of more than three dimensions. A multi-dimensional data item is a tuple consisting of n ordered elements $(a_1,\ a_2, \ldots,\ a_k, \ldots,\ a_n)$, where each element a_k is the value of the data item for the k^{th} dimension, and where $k \in \{1, \ldots, n\}$. The set of all these data items can be organized in a tabular format. One data item a^i is the i^{th} row $(a^i_1,$ $a^i_2, \ldots,\ a^i_k,\ \ldots,\ a^i_n)$, where $i \in \{1, \ldots, NrTuples\}$ and $NrTuples$ is the number of tuples. Similarly, the k^{th} column is the set of the k^{th} components of all data items. Therefore, the k^{th} column corresponds to the k^{th} dimension, where $k \in \{1, \ldots, n\}$.

$$\begin{pmatrix} a_1^1 & a_2^1 & \dots & a_k^1 & \dots & a_n^1 \\ a_1^2 & a_2^2 & \dots & a_k^2 & \dots & a_n^2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_1^i & a_2^i & \dots & a_k^i & \dots & a_n^i \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_1^m & a_2^m & \dots & a_k^m & \dots & a_n^m \end{pmatrix}$$

The core of this section further explains why visual representations are powerful alternatives to tabular representations of multi-dimensional data, investigating in detail Parallel Coordinates and Star Glyphs as two traditional examples of such visual representations.

3.1.1 What are Parallel Coordinates?

In a Parallel Coordinates visualization, the n-dimensional structure of the dataset is projected onto the 2-dimensional space of the graphical window through a set of n parallel axes. In general, these axes are vertical, but restrictions such as space impose at times the use of horizontal axes. Each axis corresponds to one dimension, which, in terms of the tabular format described above, represents one column. The origin of each axis corresponds to zero or can be set to a chosen minimum value, such as the one stored in that column. All the values corresponding to the k^{th} dimension are plotted in order along the k^{th} axis, $k \in \{1,\dots,n\}$, from origin as appropriate according to their value along the axis. All the points that visualize the components of the i^{th} data item are connected by a polyline, where $i \in \{1,\dots,NrTuples\}$. The origins of all axes are connected by a line that can be considered a pivot or a base axis (see Figure(3.1)).

For increased flexibility, the tabular format is usually correlated with a database rather than with an array. This allows the components of the data items to be of different types,

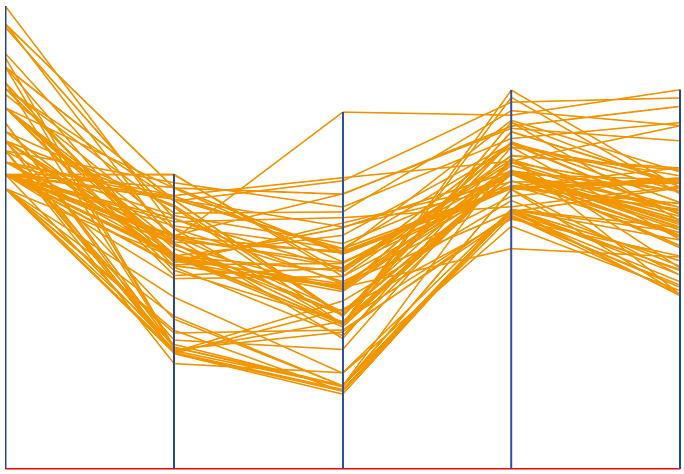


Figure 3.1: Parallel Coordinates.

numerical or not, and to have different ranges. When such a dataset is visualized in Parallel Coordinates, a few adjustments are necessary.

First, each dimension associated with a nominal scale has to be correlated with an ordered scale, either ordinal, interval, or ratio. For example, if one dimension represents the colour of the data items, some convention such as one that associates red with number 1, yellow with number 2, and so forth has to be made. This is necessary because, as mentioned above, in Parallel Coordinates all the values are sorted on the axes.

Second, if the differences between the dimensions' ranges of values are significant, another adjustment may be required. In order to provide a relevant visualization, the values are scaled in such a way as all dimensions have almost similar ranges. Otherwise, the values corresponding to some dimensions might be plotted along a very small segment, making them almost indistinguishable, while for other dimensions the values might be too widely spread.

Parallel Coordinates are a powerful visual representation of multi-dimensional data due to their capacity to display the entire dataset while supporting interactions that lead to identification of relationships among items and among dimensions [35, 39]. One great advantage of Parallel Coordinates is that all data items are visualized in the same system of axes. Thus, the use of the screen real-estate is being maximized by this layout, while a full overview of the dataset is also provided. Another advantage is that patterns and trends difficult to notice in a tabular format are revealed by the visualization. Last but not least, the solid yet simple mathematical foundation makes interactions with the Parallel Coordinates very accessible.

On the other hand, the biggest limitation of Parallel Coordinates is that the display becomes easily cluttered as the number of items or dimensions increases. The space between the equidistant parallel axes is computed by dividing the width of the graphical window to the number of dimensions. Therefore, when visualizing hundreds of dimensions, the axes can get very close to each other, thus inducing a lack of readability. Parallel Coordinates support the visualization of tens of thousands of data items (20,000 items in [92]), but as this number grows the corresponding polylines become overlapped and cluttered. These readability issues have led to research into interactions with the visualization that can be added to alleviate these problems.

3.1.2 What are Star Glyphs?

Star Glyphs are another 2D visual representation for multi-dimensional data (see Figure 3.2). Each glyph represents either a data item or a data dimension, depending on how the data is read from the table that stores the dataset. In terms of the tabular format, this means that one glyph visualizes either a row or a column. In related literature [26], [58], [61], [81], [82], [90], a common way to visualize multi-dimensional data is by using one glyph per data item. However, in this thesis one glyph represents one dimension, unless otherwise stated.

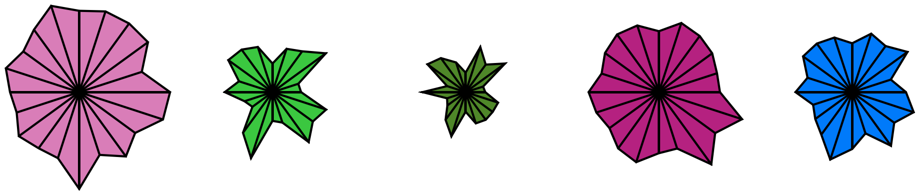


Figure 3.2: Example of Star Glyphs visualization of a data set containing information about 20 generations of plants, with one glyph per dimension. The glyphs represent in order: the number of flowers, width, height, depth, and fitness factor for each generation.

A glyph is composed of a set of spikes, frequently bounded by a polygon, that emanate radially from a central point. The first spike is chosen by convention, usually from the central point straight up. The angle between any two adjacent spikes is the same, being computed by dividing 360° by the number of spikes. If each glyph represents a data item, these spikes vary in length according to the values the selected data item has in each dimension. Therefore, the length of the k^{th} spike of the i^{th} glyph is a_k^i, thus representing the value for this item in this dimension. If, on the other hand, one glyph represents one dimension, the spikes vary in length according to the values each item has for this dimension. In this case the length of the k^{th} spike of the i^{th} glyph is a_i^k.

Star Glyphs are usually monochrome. However, colour can also be an important attribute to consider when describing them. If colour is used to convey information, it can reflect, for example, either the glyph's identification number or any other attribute that needs to be visualized of that data item or dimension. As illustrated in Figure 3.2 all the spikes usually have the same colour, but it does not have a particular meaning.

The main advantage of Star Glyphs is that shape and colour are easily and intuitively perceived by the human eye. They help the user to identify patterns. However, the disadvantage is that each glyph has a separate system of coordinates and needs a separate display area. Thus, the glyphs placement is not an easy task as it aims to optimize the use of the graphical window, while preserving an adequate ratio between the number of glyphs displayed and the size of glyphs. It is unlikely that Star Glyphs provide an overview of the entire dataset. Therefore, it is very difficult to compare non-adjacent glyphs and to reveal trends.

3.2 Parallel Glyphs: Integrating Parallel Coordinates with Star Glyphs

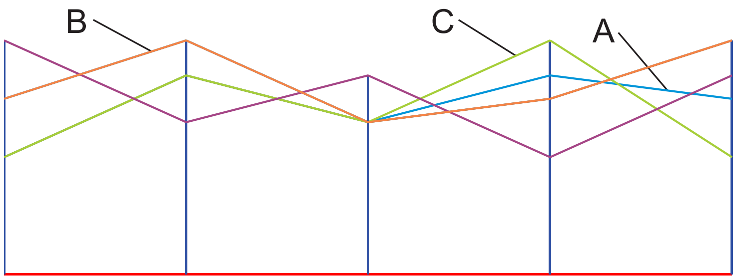
As mentioned in Chapter 2, multi-dimensional data can be visualized using several techniques, each having benefits and limitations. Nevertheless, the ultimate goal is to provide maximum support for visual analysis of data.

This thesis introduces a new approach stating that Parallel Coordinates are integrated with Star Glyphs. This aims at bridging the advantages of each individual visualization and reducing their limitations. Consequently, the information being retrieved from the new visualization will become more complex and provide increased interaction flexibility.

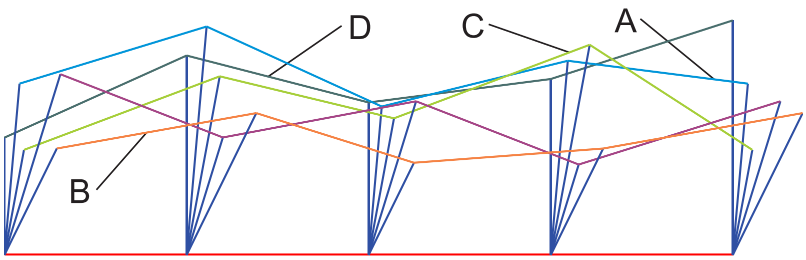
This section will first examine the concept of expanding the traditional 2D visualizations into the 3D space. Further on, the steps of how the new visualization is constructed will be examined in detail.

3.2.1 Concept: Adding the Third Dimension

One significant motivation for developing Parallel Glyphs is to address the problem of overlapping polylines. This artifact frequently occurs in Parallel Coordinates even when only a small number of data items are shown. For example, Figure 3.3(a) shows a dataset that contains only few data items. By looking at the right side of the Parallel Coordinate display in Figure 3.3(a) one can see four polylines, indicating that there are four data items. However, on the left side only three polylines are visible. Thus, the 2D visualization of Parallel Coordinates becomes ambiguous: Polyline A is obscured by either Polyline B or C between dimensions 1 and 3. This can affect even simple tasks such as revealing trends. By opening the Parallel Coordinates into 3D (Figure 3.3(b)) we are able to tell not only that Polyline A follows the path of Polyline B but also that there is in fact another polyline (D) that follows a different path which has been entirely hidden by the other polylines.



(a) Regular Parallel Coordinates.



(b) Parallel Coordinates unfolded in 3D.

Figure 3.3: Advantage of unfolding Parallel Coordinates into 3D.

As this test dataset illustrates, the problem of ambiguity created by overlapped polylines is likely to appear in real data, especially when the number of data items increases.

Another rationale for introducing the third dimension is that it provides increased interaction flexibility. For example, the order of data items cannot be perceived in 2D. However, in 3D, the sequence of data items cannot only be viewed, but also modified to enhance perception or pattern recognition, as discussed later on in Section 3.3.3. 3D rotations of the visualization is another example of the advantages of the 3D space. By applying a rotation around the y-axis, the Parallel Glyphs can be overlapped, thus enhancing their direct comparison (see Figure 3.8(a)).

3.2.2 Construction of Parallel Coordinates in 3D

The Parallel Glyphs integrates Parallel Coordinates with Star Glyphs by extending Parallel Coordinates into 3D space and unfolding them around a pivot axis (see Figure 3.4). In Parallel Coordinates data items are plotted as polylines over a set of dimension axes. In order to be able to unfold the visualization into 3D, each polyline representing a data item is assigned its own set of dimension axes and utilizes the baseline that connects the bottom of the dimension axes as a pivot axis. Thus, while the 2D view of Parallel Glyphs matches the original Parallel Coordinate view, without any further transformations, rotations can successively be applied around the pivot axis. This results in a visualization in which the data items can be interactively unfolded into three dimensions. This opening can be stopped or paused at any chosen angle to change the degree of unfolding, ranging from the original 2D view to a complete 360 degree rotation.

In the new visualization, Star Glyphs emerge during the unfolding process. By connecting the outer end points of all line segments for a given dimension the glyphs appear linked along the pivot axis (see Figure 3.5). Thus, the 3D visualization naturally integrates both types of visualizations for multidimensional data. Both Parallel Coordinates and Star Glyphs can be shown connected to each other and can be explored interactively.

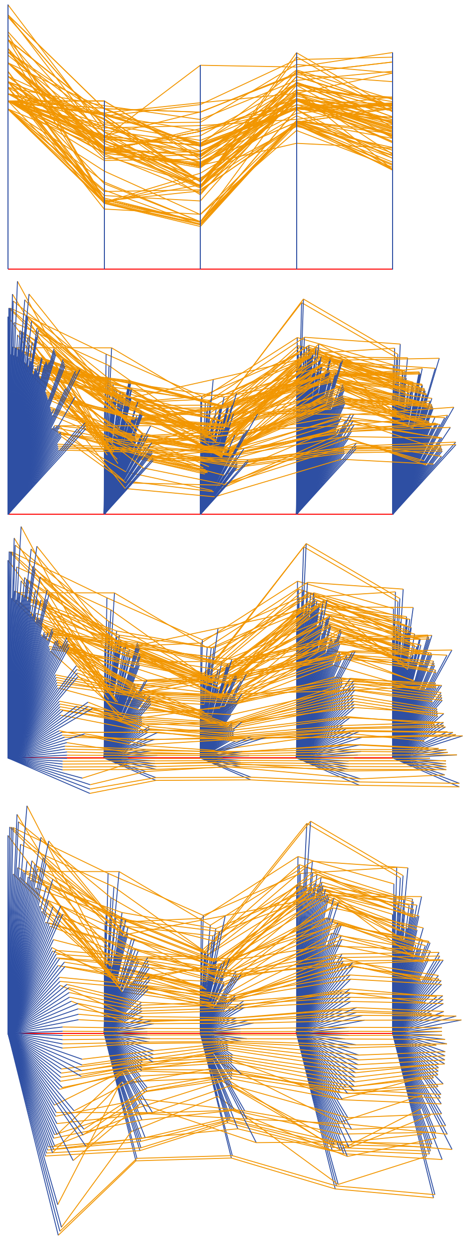


Figure 3.4: Several steps of unfolding the 2D Parallel Coordinates into the 3D Parallel Glyphs visualization.

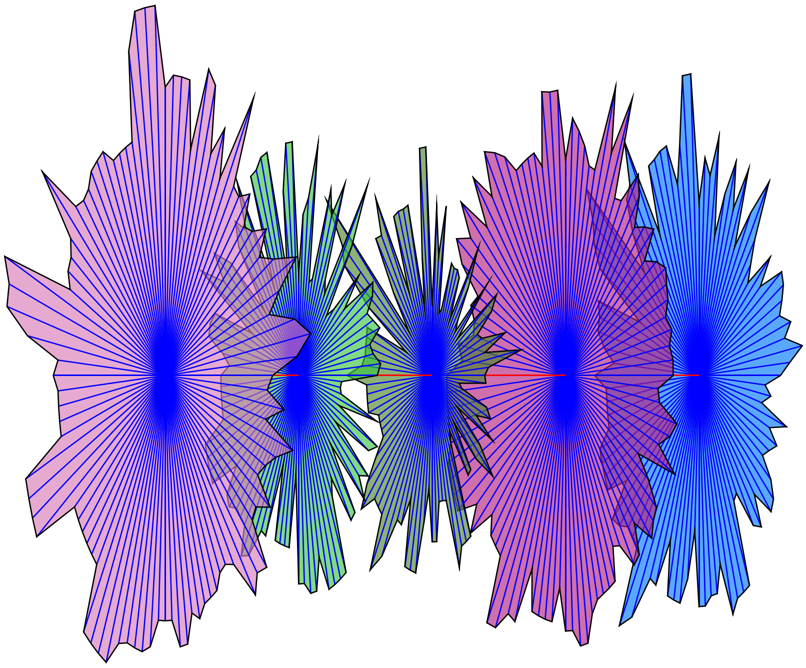


Figure 3.5: Star Glyphs emerging in the new visualization.

When building the 3D visualization I considered both orthographic and perspective projection. The perspective projection is typically used in applications that convey the sense of depth in a more realistic way. With this projection, the length of the glyphs' spikes varies, closer segments being shown longer than the farther ones. These variations make it difficult for users to accurately visually compare the lengths of the spikes. Therefore, I decided to use orthographic projection in the new visual representation thus preserving the lengths of the segments.

Traditionally, Parallel Coordinates have one axis for each dimension of the data set. Therefore, the Star Glyphs emerging in the new visualization have one glyph per dimension, each spike corresponding to one data item. On the other hand, Parallel Glyphs can generate a visualization showing each glyph representing all the dimensions of a single item (see Figure 3.6). This can be done by merely switching the way the data is read from the data table: changing from columns over rows to rows over columns or vice versa. This also results in a modified Parallel Coordinates representation that shows dimensions as polylines and objects as axes (see Figure 3.7).

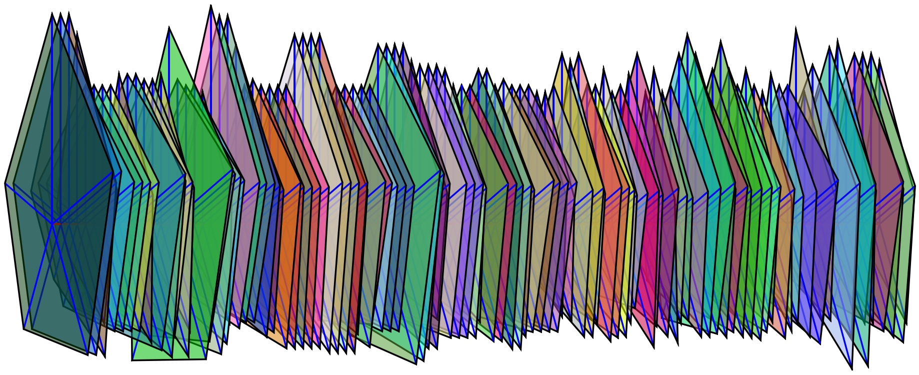


Figure 3.6: Star Glyphs, each glyph representing one object as opposed to one dimension as shown, e. g., in Figure 3.5.

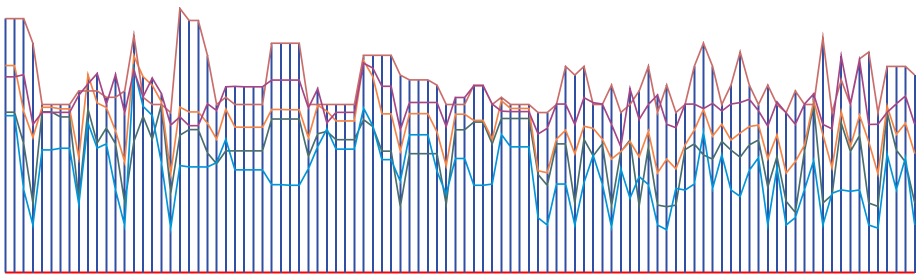
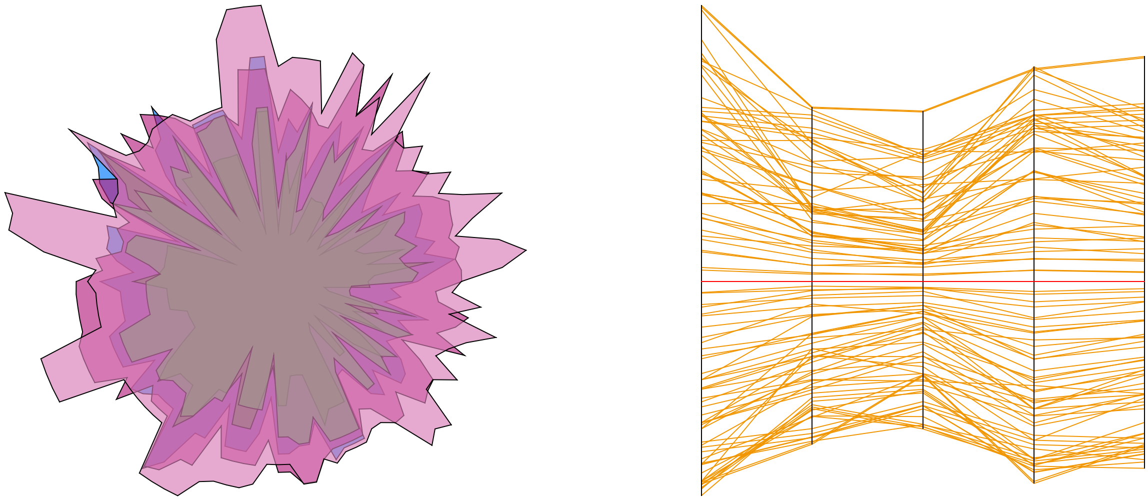


Figure 3.7: Parallel Coordinate representation resulting from Figure 3.6.

However, this switch has its limitations. Because the number of items is usually rather large, the number of glyphs aligned along the pivot line grows correspondingly (as seen in Figure 3.6). Therefore, the space between glyphs will decrease accordingly, raising the question of the maximum reasonable size of data sets for this visual representation. However, by adding some 3D interaction techniques to the new visualization, the number of displayed glyphs might be increased. Thus, the flexibility of the visualization is enhanced.

For example, such interactions are allowed because the pivot line that connects the glyphs is just an abstract element and not a preferred dimension or a time axis. Therefore, even if the order of glyphs is initially the same as the order of items in the data table, this can be modified to better support each task.

A disadvantage of traditional Star Glyphs is that a loss of context can be induced by spreading the glyphs in a 2D layout. This can lead to difficulties when conducting comparisons across glyphs. However, the new 3D visualization displays the glyphs naturally aligned on the pivot axis (see Figure 3.5). This new layout and additional 3D interactions



(a) Side view. (b) Front view.

Figure 3.8: Parallel Glyphs aligned on the pivot axis to allow comparison of data items.

can address the issue of comparisons. For example, rotating the pivot axis around y so that the data item of interest is displayed topmost offers a direct comparison of the overlapped glyphs (see Figure 3.8(a)). If no y rotation is applied, a side view of the unfolded Parallel Glyphs is obtained (see Figure 3.8(b)). The employed orthographic projection makes the comparison of glyphs valid, because the lengths of the spikes are not affected by perspective deformation. Moreover, maintaining the polyline connections provided by the Parallel Coordinates can also aid a more accurate comparison (see Figure 3.9). Additional 3D interactions are also described in Section 3.3.2.

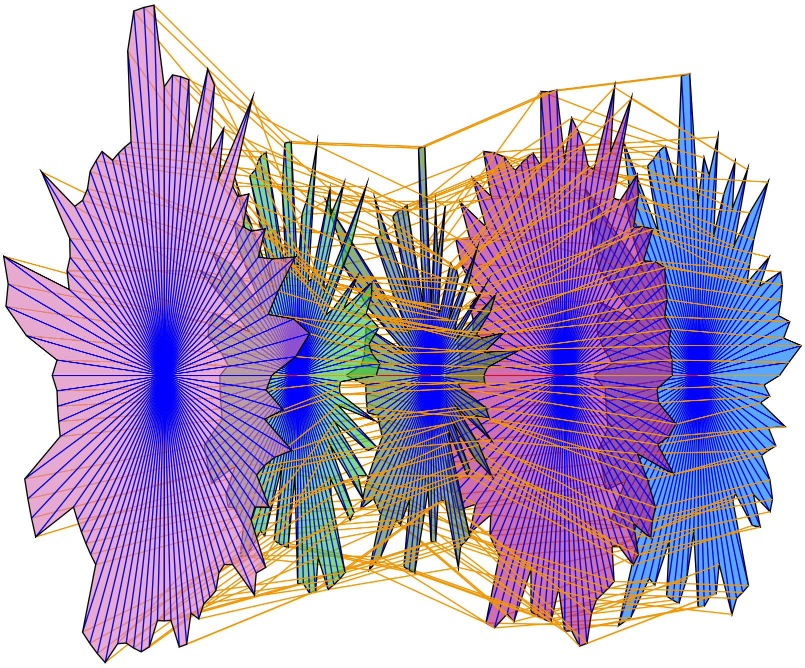


Figure 3.9: Setting glyphs into relation to each other by arranging them on the pivot axis and by connecting them with polylines.

3.3 Readability

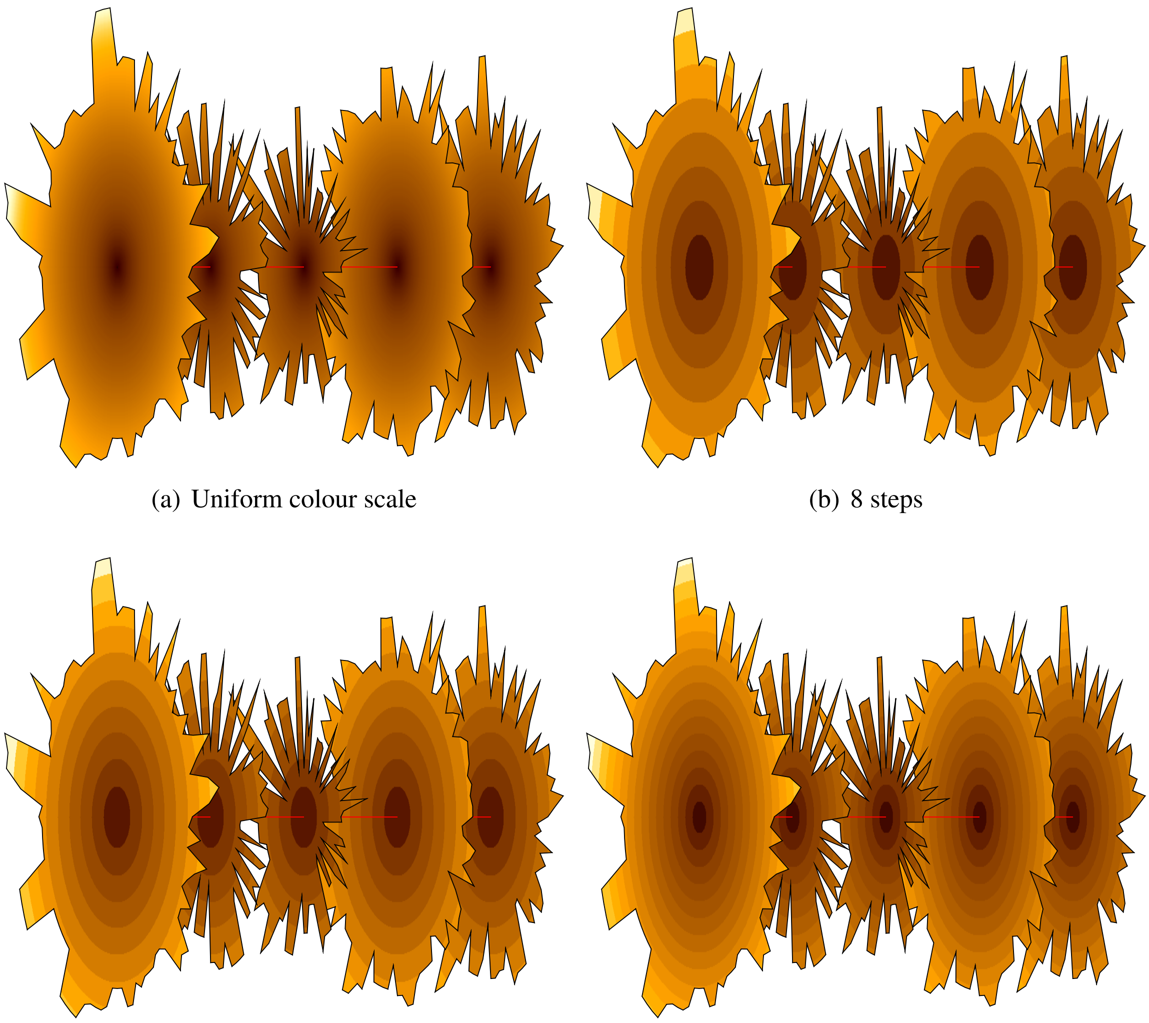
This section will show how data exploration using Parallel Glyphs is enhanced by using colours and several interactions. The uniform and stepped colour scales applied to glyphs as well as to polylines are important visual cues. In addition, interactions initially specific to the 2D Parallel Coordinates visualizations are now provided for Parallel Glyphs and extended with new interaction techniques originating from the 3D nature of Parallel

Glyphs. Basic interactions such as brushing the polylines and dimension axes and reordering dimensions are now complemented with polyline rearrangement and lens techniques for focus + context capabilities.

3.3.1 Colour

In Parallel Coordinate visualizations the polylines representing the data items are typically drawn in a uniform colour or in proximity-based shades that convey information about the hierarchical structure of the data [26]. Likewise, for glyphs in general, colour is an important attribute, used to represent data [46, 89] in addition to size, position, and orientation. In contrast, for Star Glyphs in particular the most common way of mapping data attributes is not through colour but through the spikes' lengths which represent the values of the corresponding data item. Thus, a Star Glyph's shape becomes its main visual cue. Unfortunately, comparing spikes becomes increasingly difficult with growing distance between them, especially if one wants to compare spikes across more than one glyph. Even in 2D it can be difficult to accurately compare the lengths of spikes that are set at wide angles apart from each other or belong to different glyphs. Moreover, by adding 3D rotations to Parallel Glyphs, further difficulties can be created. Although orthographic projection does make these comparisons valid for views that are orthogonal to any two of the x, y, and z-dimensions, comparison is not an easy task.

I address this problem by adding circular textures to the glyphs as an additional visual cue. I create these textures by adjusting uniform colour scales [53] (see Figure 3.10(a)) to offer an increased readability for glyphs, according to the following three criteria:



(a) Uniform colour scale

(b) 8 steps

(c) 10 steps

(d) 16 steps

Figure 3.10: Uniform and stepped heated colour scales applied to the glyphs to increase comparability between glyphs.

- provide a means for comparing pixels situated at a distance,
- follow some natural ordering of colours, and
- be readable by people with colour vision deficiencies.

While uniform colour scales are discussed extensively in the literature [53], these scales are typically based on algorithms that attempt to maximize the number of just noticeable differences (JND) between neighbouring pixels. However, this conflicts with the first criterion above, as the focus is to recognize differences across a distance, rather than JND between neighbouring pixels.

The approach I take extends the notion of a uniform colour scale. I propose to use a set of stepped scales with 16, 8, and 10 steps based on the numerical values provided by [52] (see Figure 3.10). These numbers of steps were chosen for subdivision measurement reasons: 16 and 8 are powers of two, like the size of the texture map that stores the colour scale. 10 is useful for solving tasks involving percents of the values plotted along the dimension selected to be the reference. In this case, each step represents 10% of the maximum value of that dimension. Each sector of a stepped colour scale has a single colour, obtained by averaging the amount of red, green, and blue of corresponding pixels from the uniform colour scale. Subsequently, this allows the colour scale to meet the second criterion above (the natural sense of order induced by a uniform scale) while adding the ability to easily notice ranges of values (first criterion).

As the human eye can better notice variations in luminance along the yellow-orange axis and variations in hue along the magenta axis, according to [53] the heated and magenta scales are recommended. Knowing that users have different preferences when choosing the colour scale I provide several other options as well. Moreover, I also considered colour scales based on opposite colours with the two opposite colours on each end of an axis, as recommended in [84] but I chose not to use them for two important reasons. First, the variation on the red-green channel would be difficult for people with colour vision deficiencies. Therefore, the red-green colour scale would contradict the third criterion above. Second, the variations on the blue-yellow channel are hard to distinguish on very thin sectors [84] as is likely to happen in glyphs with many spikes.

I complement the support offered by a uniform colour scale with a ring-ruler, centred around the pivot point for a glyph, that can be interactively sized (see Figure 3.11). When the user clicks a point inside a glyph, this ring is drawn on each glyph as a reference for comparison. The rings have a radius equal to the distance between the reference point and the centre of its corresponding glyph. They were created to assist in fine comparisons and can be used as a ruler.

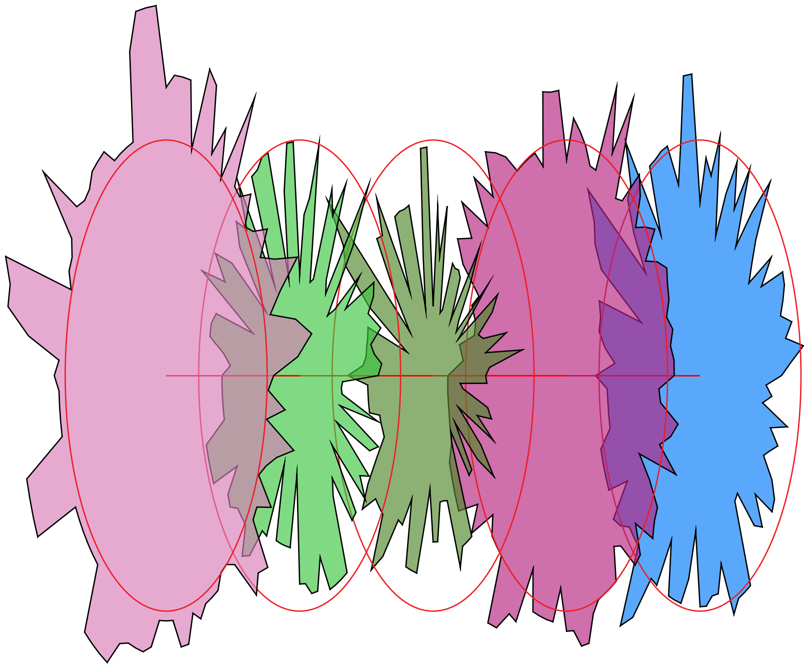
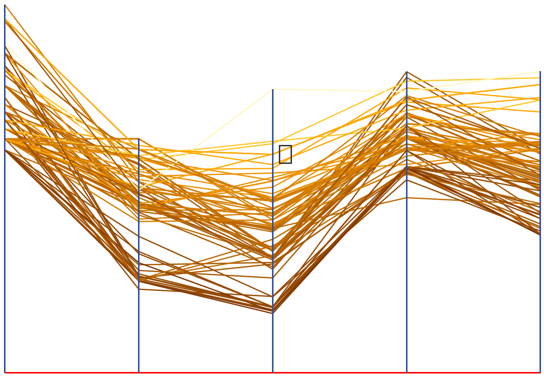
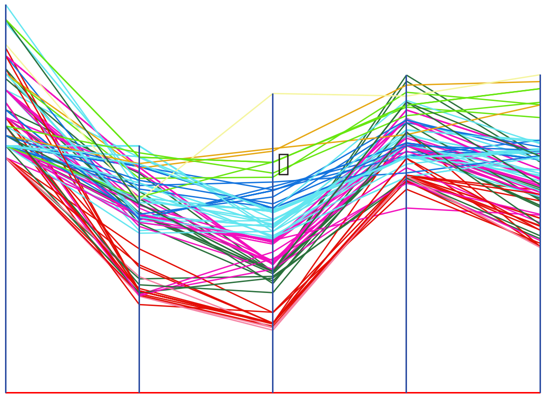


Figure 3.11: Using rings as rulers to compare values between glyphs. In this example it is now possible to tell that the big top spike on the middle glyph is slightly shorter than the corresponding spike on the glyph to its left and that it represents the third biggest value for this dimension.

The colour scales can also be applied to the polylines, having one main dimension selected as a criterion for choosing the colour (see Figure 3.12(a)). This is especially useful when using the 2D Parallel Coordinates visualization to observe, for example, data trends. However, when using a stepped colour scale based on a uniform one, the steps may not be sufficiently distinguishable for other tasks such as selection.



(a) Brush-selecting a range of polylines with the help of a stepped colour scale applied to the lines based on a selected dimension. Note that even with stepping applied to the colour scale it is hard to make out differences.



(b) Using a stepped scale with distinct colours for each step helps to increase the contrast between adjacent colour ranges to better support brushing. In addition, trends are more visible for the entire dataset compared to Figure 3.12(a).

Figure 3.12: Stepped colour scales assigned according to the middle dimension. The selection is easier using the colour scale on the right, having distinct colours for each step.

To mitigate this problem I provide a stepped scale that has a distinct colour for each step (see Figure 3.12(b)). Within this colour scale it is easy to identify and select polylines having the values corresponding to the main dimension in the same range. This is valid even if the polylines are not adjacent or if the glyphs are folded up or only slightly rotated. Of course, this colour scale will not always meet the second criterion above (natural order of colours). Therefore, a legend for the colour scale as an aid to the visualization reader is recommended.

3.3.2 Interactions

Regardless of the technique employed, visualization alone is not enough for a thorough data analysis. It is of great help for the user to be able to interact with the visualization, in order to visually solve tasks involving datasets of various size and complexity. Therefore, a number of basic interactions are common to most of the existing visual representations. They are also supported and, moreover, extended by Parallel Glyphs.

- **Selection** is an important interaction, as often a task does not refer to the entire dataset. For example, selection allows the user to look at subsets of one or more elements at a time, in order to better see details or to discover patterns. In Parallel Glyphs visualization, I provide the same set of selection tools for dimensions as well as for data items. Therefore, from this point on, I will further refer to dimension axes and polylines as ”objects”, because they have a similar description of the interactions.

  The indication and left-click action works as a toggle; that is clicking on an deselected object will select it, and clicking on a selected object will deselect it. In this mode only one object of each type (polyline or dimension) can be selected at a time. A click on an empty area will deselect all objects. One possible issue with this interaction is that when several polylines are overlapped, only the top-most is selected. Depending on the task, this fact can be useful or not. For example, if it is necessary to drag a polyline to see the path of the overlapped polylines, it is useful to select only the first object. However, if a task requires to select all polylines that have a certain value, this simple selection is not appropriate. To select more than one object at the same time, another option is available for both dimensions and data items. Brush selection allows the user to select the group of dimension axes or polylines that intersect a rectangular area that is interactively drawn on the graphical window. This area is determined by dragging the mouse between the points where the left button is pressed and released.

To allow the addition or deletion of other objects to/from a subset of objects already selected, a “multiple selections” option is available. This feature can be chosen from the toolbar corresponding to either dimension axes or polylines. Multiple selections work with both methods, click or brush. This time, selecting an object will not deselect the rest. However, for the clicked object it will still toggle the selection status.

Once the subset of desired objects or group of objects is selected, any future operation will apply only on this selection.

- **Deletion** of either the selected or the deselected objects is a common operation to choose following the selection process. It allows the user to reduce the clutter of the graphical window and to focus on fewer objects, when solving tasks such as data item comparison or pattern recognition. The objects (dimension axes or polylines) are actually only made invisible, deleting them from the graphical display, not removing from the dataset. This makes it easy to restore the deleted objects and to continue the visual analysis on the entire dataset.

- **Dimension reordering**

  Usually the dimensions are initially displayed in the order they are placed in the tabular format of the dataset. However, this ordering can obscure possible relationships among dimensions, which leads to a cluttered and less meaningful visualization.

Therefore, providing tools to reorder them is highly important. In related literature [90] these tools are either automatic, using algorithms that determine an optimal order, or manual, allowing the user's input. The automatic approach is mostly used for highly dimensional data, in conjunction with clustering algorithms. Clustering is a process that separates objects into groups, in order to display only one representative for each group, as discussed in Chapter 2. The main issue with the automatic reordering is that particular aspects of the data might not be reflected in the solution offered by the algorithm, although they are obvious for the user. Manual methods on the other hand, rely upon a user's observations and interactions. Therefore, they are limited by user's expertise.

For dimension reordering in Parallel Glyphs I have chosen the manual approach based on the following two reasons. First, it provides more flexibility, allowing the user to explore data while still considering previously known facts. Second, this visualization is currently being used for datasets with a relatively small number of dimensions, for which clustering is not necessary. To change the position of one or more dimension axes in the Parallel Glyph visualization, the axes are first selected as described above. Next, the "Drag" button is chosen from the toolbar corresponding to dimensions. Dragging the mouse will then move the selected axes in the new position, which leads to the desired order of dimensions. The distance between all the axes can be automatically evened by using the appropriate button from the toolbar.

- **Zooming and panning** are other interactions commonly used by visualization techniques that are also available for Parallel Glyphs. They provide access to details or an overview of the visualization. A complementary feature is also offered, allowing the user to change the distance between adjacent dimensions.

- **3D rotations** The interactions described above are generally specific to 2D visual representations. It is now possible to unfold the Parallel Glyphs using various angles as described in Section 3.2.2 and to apply 3D manipulations to the resulting visualization. For example, the user is able to rotate the visualization in all three directions around the pivot axis. A particularly interesting situation is when the glyphs are unfolded and the entire model is rotated such that the glyphs directly overlap each other. The transparency supports direct comparison of all glyphs (see Figure 3.8(a)). In addition to these interactions I implemented a few new techniques described later in this chapter.

- **Fog** is a visual cue specific to 3D visualizations. In general, it provides a sense of depth and realism. For Parallel Glyphs, in particular, it can be also useful for reducing the clutter, as it obscures the back side of the image (see Figure 3.13). Through the 3D rotations the objects previously hidden by the fog can be brought forward and made available for exploration. To mix the colour of the objects with the fog's colour, the user has three methods available. For the linear method, the start and the end positions of the fog need to be specified. For the exponential and Gaussian methods, the density of the desired fog effect is taken into consideration.

3.3.3 Polylines Rearrangement

In addition to the order of dimensions, the order of tuples is now also meaningful, as Parallel Coordinates have become three-dimensional. Therefore, I provide the possibility

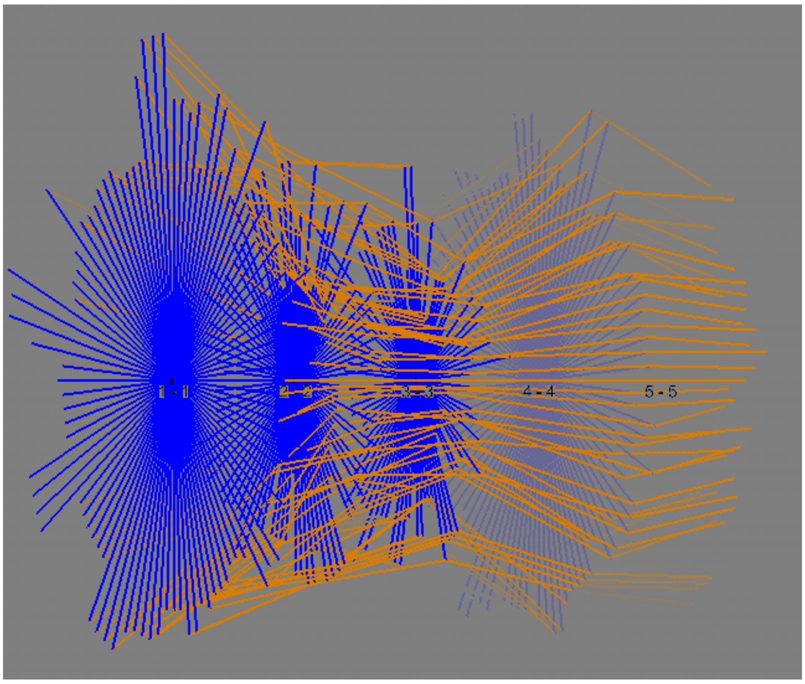


Figure 3.13: Using fog to hide farther objects and also to provide the visualization with a sense of depth.

of interactively changing this order. The user has the capability to select one (through clicking) or several (through brushing) polylines and then to rotate them around the pivot axis to a different position in the sequence. This allows one to adapt the sequence of polylines in the data set to directly compare polylines that were previously far apart. Thus, the user can avoid one of the main readability issues of a Star Glyphs visualization, which is the difficult comparison of non-adjacent spikes. Although in Parallel Coordinates the values of different data items corresponding to the same dimension are plotted along the same axis, their comparison is not necessarily easy because they might be overlapped by other polylines. Parallel Glyphs makes the best of both worlds, supporting the interactive adjustment of overlapping polylines and the possibility of placing items adjacently for a given task.

Dimension reordering has already been shown to be useful for identification of patterns or trends in the data set. Similarly, the polyline reordering can support this type of task, beside facilitating the comparison of data items. Parallel Glyphs provide interactive item reordering as the user might have knowledge about the data set that would elude an algorithm. However, an automatic operation equivalent to the automatic dimension reordering would be interesting to develop.

The fact that the angle between two consecutive spikes of the glyphs is no longer constant provides a flexibility that opens up new paths for data exploration. In Parallel Coordinates, in order to analyze a subset of the data independently from the rest of the set, the items not important for the task are either made invisible or faded out. In Parallel Glyphs, the items involved in a task can be maintained on the top side of the screen while the remaining subset can be rotated to the bottom. Thus, the entire set is visible and available for selection at any time, giving the user full overview of the data and making it easy to switch between items from one category of interest to another. Both the top and bottom subsets of the data items benefit independently from different angles of unfolding, while the entire model can be rotated in 3D in order to provide the best view (see Figure 3.14).

3.3.4 Lens interactions

Depending on the data set, the visual density of the representation can make it extremely difficult to perform visual selection of polylines that satisfy a given criterion or to identify overlapped data items. I address this problem by integrating focus + context capabilities using two types of lenses.

The first one, EDGELENS [87] facilitates the exploration of the data set in Parallel Coordinates by adjusting the segments that fall into its area of influence into Beziér curves

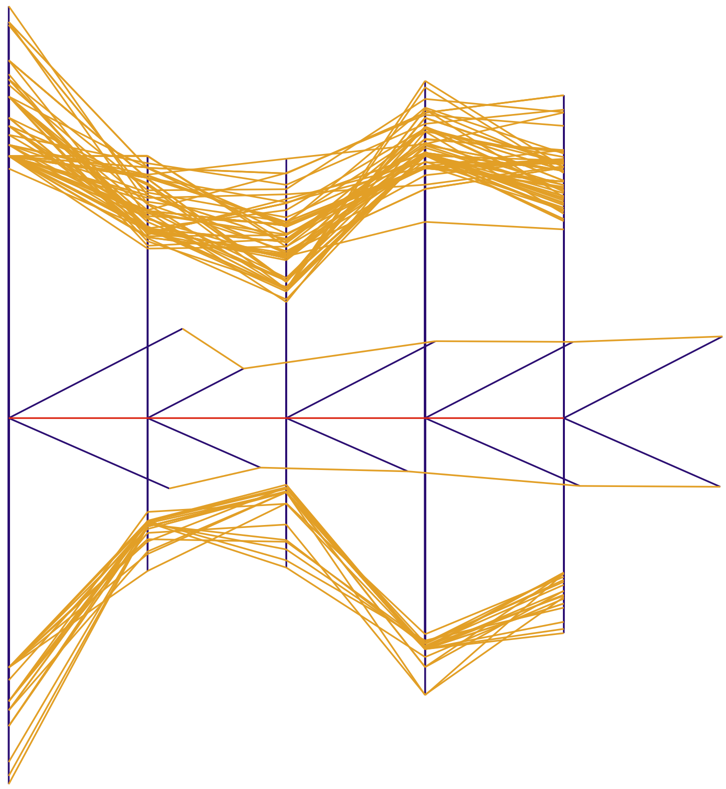


Figure 3.14: Random rearrangement of selected data items in 3D. Further comparisons with other items are possible because the rest of the data set is still visible.

(see Figure 3.15). These curves are shuffled at gradually decreasing distances from the centre of the lens, thus expanding the space available between polylines and providing more insight into the dataset. The centre of the lens can be easily moved until the desired presentation is reached. The required polylines can then be selected, either individually or by brushing.

For situations where the data items wanted have a particular value or range of values for a specific dimension, a second type of lens may be used. For ease of selection purposes a non-linear expansion of the chosen axis is offered. The implementation of this axis-specific lens uses the EPSLENS library to provide the non-linear expansion of the axis in

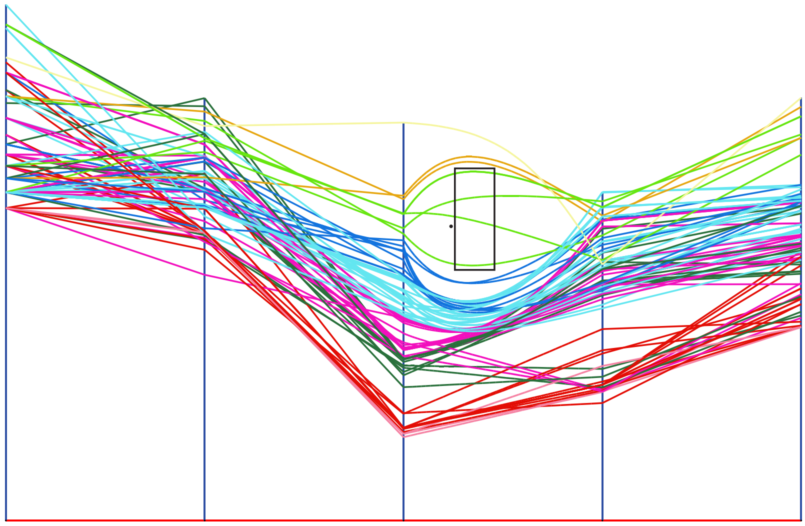
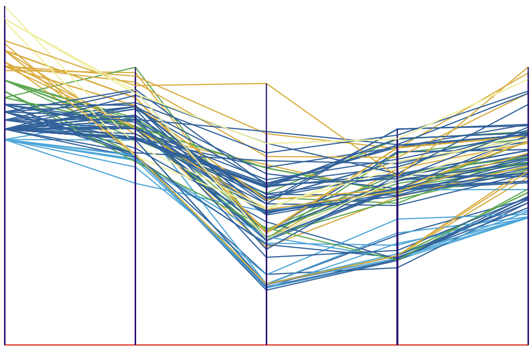
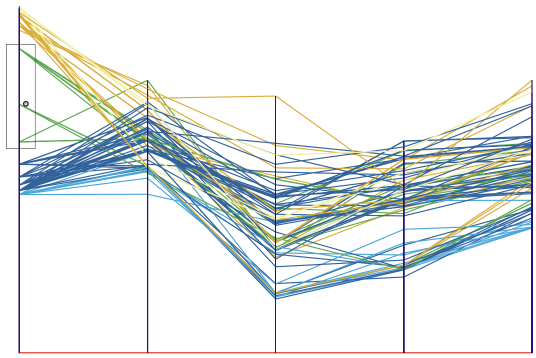


Figure 3.15: Edge Lens interaction with Parallel Coordinates.

the immediate vicinity of the lens (see Figure 3.16(b)). Thus, the points plotted along the chosen section of this axis are separated by increased distances, allowing the identification and brush selection of polylines needed for a particular visual task.



(a) Parallel Coordinates before applying the axis-lens.



(b) Axis-lens that uses the EPS library interacts with Parallel Coordinates along the first dimension axis.

Figure 3.16: Selection of data items having the values corresponding to the first dimension within a very small interval is facilitated by an axis-lens using the EPS library.

3.4 Case study

As a specific example, I apply Parallel Glyphs to a set of 100 generations of plants generated by a genetic algorithm [41], each having five attributes: number of flowers, the dimensions of the plant in 3D (width, depth, and height), and the fitness factor. These are the five dimensions used in the visual representation, with each generation represented by one polyline. In this section I look at how the integration of Parallel Coordinates and Star Glyphs along with the available colour scales supports the user in solving a series of tasks.

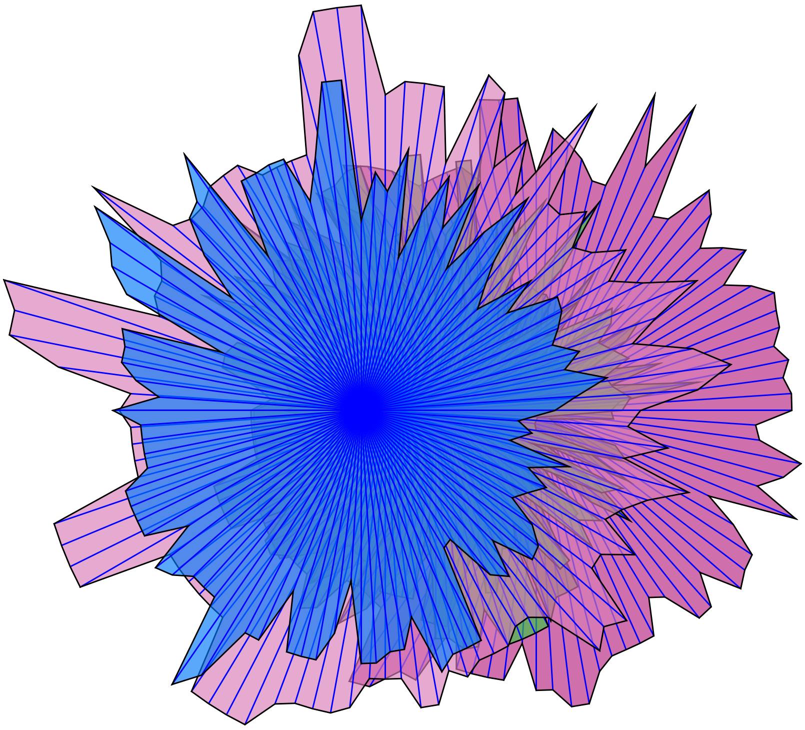
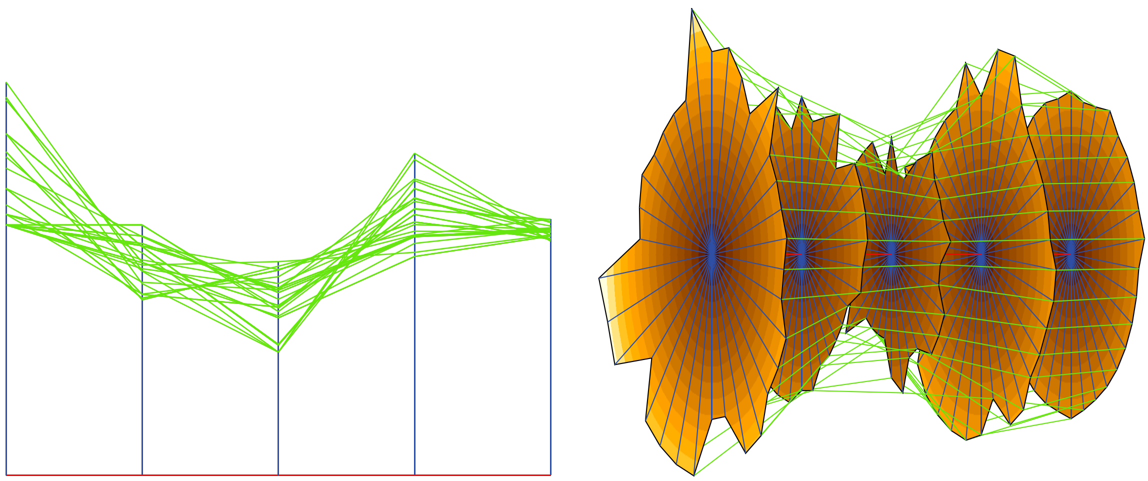


Figure 3.17: Task 1: Comparison of a sequence of data values using repositioned Star Glyphs.

The first task is to compare the variation of flowers and fitnesses along the 100 generations. In Parallel Coordinates, comparing the range of values plotted along the first and last axis is straightforward. However, this representation does not provide information about how these values vary from one generation to another. This can easily be solved through Parallel Glyphs by unfolding the glyphs and by bringing the two desired glyphs next to each other (see Figure 3.17). The variations between these two attributes can be easily compared by combining the interactive ring ruler and the rotations in any of the three dimensions. This reveals a dependence relationship between the number of flowers and the fitness factor (the two Star Glyphs in front in Figure 3.17).

In the second task, one wants to understand how the shape of plant changes over a selected number of generations for the plants that have a fitness factor of between 70%–80%. First, to visually select the desired items, the user applies the rainbow scale with 10 steps to the whole set of polylines, using the values of last dimension as a criterion for choosing the colours. Thus, each distinct colour in the colour scale corresponds to a



(a) Data items selected with help of colour scale.

(b) Unfolded selected set of data items.

Figure 3.18: Task 2: examination of a selected subset of the data.

range of 10% of the maximum fitness. Next, polylines that have a colour corresponding to the third sector of the colour scale (green) must be selected, but because the lines are too close to each other, accurate selection is very difficult. Using the dimensional axis lens, the user expands the space between those lines and then brushes only the green ones (see Figure 3.18(a)). The deselected polylines can be either made invisible (as in the Figure 3.18(a)) or rotated to the bottom of the screen to keep them available for further tasks. By choosing to keep only the selected data items the user is free to completely unfold the glyphs formed with the remaining items (see Figure 3.18(b)). The glyphs corresponding to width and height (second and third from left) have similar shapes but different sizes. This means that the variation of these two attributes is similar. However, the glyph corresponding to depth (fourth from left) has peaks where the other two have local minima. This means that in the generations where the width was small the height was also small but the depth was inversed. This display suggests that the plants in each generation varied their dimensions to try to maintain a constant fitness and volume.

3.5 Summary

In this chapter I have introduced Parallel Glyphs which are a three-dimensional extension of Parallel Coordinates integrated with Star Glyphs. It demonstrates that Parallel Coordinates and Star Glyphs are not only closely related and part of the same family of visualizations but that they are, in fact, orthogonal to each other. I showed that the traditional view of Parallel Coordinates—objects over dimensions—leads to one of two versions of Star Glyphs—one glyph representing all values for one dimension. The other version of

Star Glyphs—one glyph representing the values of one data object—can be derived from a second version of Parallel Coordinates in which dimensions are plotted over objects.

I have enriched the new 3D visualization by applying colour scales as textures to the 3D Star Glyphs to support the task of comparing spatially separated glyphs. Both uniform and stepped colour scales were discussed and their merits compared. I also showed how these colour scales can lead to colouring the data items' polylines in order to identify trends and support brush-selection. To further support visualization tasks various interaction possibilities have been included. In addition to well-known techniques for 2D Parallel Coordinates, I have introduced unfolding of 2D Parallel Coordinates into the new 3D representation, 3D transformations such as rotation around coordinate axes, and a ring ruler interaction that enables determining even small differences between Star Glyphs. Furthermore, I discussed the newly possible data item rearrangement that enables users to directly compare selected data items with each other without loosing the context of the rest of the data set. Finally, I included enhancements to interactions that are applied to the traditional 2D Parallel Coordinates visualization such as lens techniques for focus + context data exploration.

Chapter 4

A Family of Multi-Dimensional Visual Languages

Chapter 3 described how two 2D visual representations of multi-dimensional data could be united to create an integrated 3D representation. One motivation for the introduction of this new technique was to address the problem of occlusion. In the context of information visualization, occlusion is the situation when parts of the visual representation are not visible because other parts are rendered in front of them. Occlusion often appears in Parallel Coordinates because of the overlapping polylines.

This chapter comes to enable a formal description of the original 2D representations that emphasizes the fact that they belong to the same family of visualizations. Moreover, this formal description of Parallel Coordinates and Star Glyphs will provide the grounds for establishing when and how different types of occlusions can appear. I present here a linguistic formalism built in analogy with the natural languages and capable of describing a category of visual representations of multi-dimensional data. This linguistic formalism is called the Family of Multi-Dimensional Visual Languages (*FMDVL*).

This chapter is organized as follows. The introduction sets the context for *FMDVL*. The following section explains how morphemes are defined and combined to create an ordered alphabet. This is the *FMDVL* alphabet, $\mathscr{A}$. Next, the Parallel Coordinate Visual Language (*PCVL*) is defined by developing a morphology and syntax that describe the Parallel Coordinates representation. *PCVL* is just one example of a visual language that can be built with the *FMDVL* alphabet. A different morphology and syntax can specify another visualization method of multi-dimensional data. To demonstrate this, the next section details the morphology and syntax for the Star Glyph Visual Language (*SGVL*). The chapter concludes with a summary of the visual languages described by this linguistic formalism.

4.1 Introduction

In general, a language is considered to be a "system of communication that uses signs arranged in a particular way" [55]. A stricter definition is given by Chomsky, who says that a well-defined language requires an alphabet and a set of grammatical sentences, where an alphabet is a finite set of symbols, used to build sentences. Most natural languages that, according to the convention made in Chapter 1, Section 1.1.5, are written languages, are formalized through an alphabet composed of an ordered list of morphemes [33]. These morphemes are grouped according to a selected morphology to form words. While a family of related languages may use the same alphabet, each distinct language has its own morphology. In turn, these words are combined according to syntactical rules to create sentences and paragraphs. This is similar to Saint-Martin's [69] approach to defining a visual language by providing "a preliminary level of description, analogous to phonology in verbal linguistics, that can explain how primary elements are joined together to form larger units". Next, he says, their syntactic rules of association can be studied, in order to complete the grammar of the visual language. However, in Saint-Martin's view, a visual language is a way of communication that uses visual arts such as painting. In this thesis the meaning of visual language implies the use of visual representations defined in Information Visualization. This connotation is also influenced by Horn's [33] definition of a visual language as any integration of shapes, images and words but not one of these aspects independently. A misleading sense of the term "visual language" is increasingly used as an abbreviation for "visual programming language". However, this is not the meaning used in this thesis.

The linguistic formalism described in this chapter follows the aforementioned steps of defining the grammar of a visual language. The Family of Multi-Dimensional Visual Languages *FMDVL* consists of an alphabet, $\mathscr{A}$, and morphologies and syntaxes that use this alphabet, each describing a specific visual language. The alphabet $\mathscr{A}$ of *FMDVL* can be considered an analogy to the Latin alphabet, which is used by English, French and many other languages. Similarly, $\mathscr{A}$ can be used to build several visual languages by defining different morphologies and syntaxes. This capability is illustrated by building two such visual languages: the Parallel Coordinates Visual Language (*PCVL*) and the Star Glyphs Visual Language (*SGVL*). *FMDVL* can formally specify visual representations created from multi-dimensional data that can be stored in a tabular format. Whether it can describe all multi-dimensional data visualizations, or even all visualizations of multi-dimensional data that can be stored in tabular format is beyond the scope of this thesis and is left for future research.

4.2 The Alphabet

The construction of the alphabet $\mathscr{A}$ is based on two definitions. First, in the Oxford English Dictionary [34] the alphabet is defined as a finite set of letters or symbols in a fixed order used for writing a language. Second, based on Horn's [33] definition of a visual language, an elementary unit of an alphabet must integrate shapes, images and words. Combining these two definitions, this section describes an alphabet composed of a finite set of ordered morphemes that integrate shapes and text. This alphabet will actually consist of two types of morphemes: *MDmorphemes* used to describe multi-dimensional (MD) data and the auxiliary *PVmorphemes* necessary to describe elements of the information visualization technique such as pivot (PV) points.

4.2.1 Notations

Prior to building the alphabet, I provide the set of notations and definitions needed to set up the context.

Let T be the set of numerical values to be visualized, initially stored in a data table, where a row represents a data item and a column holds the values of all data items for one dimension.

Notation: *NrDim* is the number of dimensions of the dataset, which is the number of columns of a data table.

Notation: *NrVisDim* is the number of dimensions that are visible at one moment. This needs to be distinct from *NrDim* because it is possible that not all dimensions be used in all visualizations.

Notation: *NrTuples* is the number of data items of the dataset, which is the number of rows of the table.

Notation: *NrVisTuples* is the number of visible data items at one moment.

Notation: D = the set of dimensions. $D = \{1, \ldots, NrDim\}$

Notation: *visD* = the set of visible dimensions. $visD = \{1, \ldots, NrVisDim\}$

Obviously, $NrVisDim \leq NrDim \Rightarrow VisD \subseteq D$.

Notation: E = the set of elements (or data items). $E = \{1, \ldots, NrTuples\}$

Notation: $visE$ = the set of visible data items.$visE = \{1,\ldots,NrVisTuples\}$

Similarly, $NrVisTuples \leq NrTuples \Rightarrow VisE \subseteq E$.

Notation: $RGBA = [0,1] \times [0,1] \times [0,1] \times [0,1]$, $RGBA$ = the set of possible colours, described by the red, green, blue and alpha components, where alpha represents the degree of opacity.

4.2.2 MDmorphemes

Using these notations, it is possible now to define a mapping between the space of the natural language in use (e.g. the data table will be written in a natural language of words and numbers) and the space of the visual language. Later, for each definition of an element of the visual language, its correspondence to the space of the visual representation is explained.

Definition 4.2.1. This mapping is established by the following function f:

$f\colon T \rightarrow (D \times E, \mathbb{R}^3, RGBA)$.

The expression of f can be written in an encapsulated format as follows:

$f(P) = (indices_p,\ coord_p,\ clr_p),\ \forall P \in T$,

where P represents both the indices $indices_p = (id_{d_p}, id_{t_p})$ of a cell in the table T and the content of the cell specified by $indices_p$.

Any component of $f(P)$ can also be expanded, for more clarity, given that:

$indices_p = (id_{d_p}, id_{t_p})$, where:

> $id_{d_p} \in D$ is the column number of the cell represented by P. It is called the *identification number of the dimension* that contains P;
>
> $id_{t_p} \in E$ is the row number of the cell represented by P. It is called the *identification number of the tuple* that contains P;

$coord_p$ $= (x_p, y_p, z_p) \in \mathbb{R}^3$ are the 3D coordinates of the point that corresponds to *P* in the graphic space. Should one want to create 2D representations z_p can be 0 or have any fixed value;

clr_p $= (r_p, g_p, b_p, a_p)$ is this point's *RGBA* colour, with $r_p, g_p, b_p, a_p \in [0,1]$.

While it is important that each morpheme has a certain position and colour, the actual coordinates, and the red, green and blue values of the colour depend on each morphology. The upcoming sections will use various combinations of encapsulated and expanded components of $f(P)$, depending on which parts of the expression need more detailed explanations.

From this definition of *f* it follows that each element *P* in the data table has a unique correspondent in the space of the visual language, which means that *f* is a well-defined function.

At this point, a mapping from the space of a multi-dimensional data table to a space of a visual language has been defined. Each correspondent *f(P)* of a point P from the data table via *f* is a morpheme, as follows:

Definition 4.2.2. An *MDmorpheme* is an element
$(id_d, id_t, x, y, z, clr) \in (D \times E,\ \mathbb{R}^3,\ RGBA)$.

Notation: $\mathscr{MD}$ = the set of all *MDmorphemes*.
$\mathscr{MD} = \{MDmorpheme\} = \{(id_d, id_t, x, y, z, clr) \in (D \times E,\ \mathbb{R}^3,\ RGBA)\}$.

Therefore an *MDmorpheme* corresponds to an element *P* of a data table *T*. On the other hand, the correspondent of an *MDmorpheme* into the graphics space of the visual representation of the multi-dimensional data is a point (see Figure 4.1). This point has the labels id_d and id_t, the coordinates x,y,z and is drawn with colour clr. As mentioned above, detailing how x,y,z and clr are computed is strictly dependent on the visualization technique intended to be described by a particular visual language.

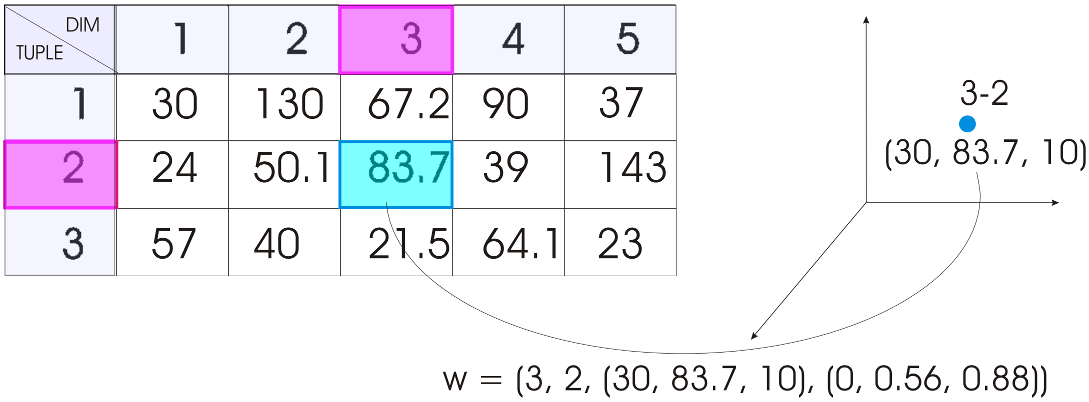


Figure 4.1: Illustrating the mapping from the multi-dimensional data table to a point in the visual representation space via an *MDmorpheme*.

Two reasons have determined the inclusion of id_t and id_d in the structure of an *MDmorpheme*. First, to comply with Horn's definition of a Visual Language [33], this definition combines the graphical point with numbers composing the two labels that state the identification number of the corresponding dimension and tuple. Second, these two numbers provide a means of sorting the set of *MDmorphemes*, thus completing the Oxford English Dictionary [34] definition of the alphabet, making the *FMDVL* alphabet ordered.

The following relation of order is established on $\mathscr{M}\mathscr{D}$.

Definition 4.2.3. $\forall m_1, m_2 \in \mathscr{M}\mathscr{D}$

$m_1 < m_2 \overset{def}{\Longleftrightarrow} (\ id_{t_1} < id_{t_2}\ )\ OR\ ((\ id_{t_1} = id_{t_2})\ AND\ (id_{d_1} < id_{d_2}))$

$m_1 = m_2 \overset{def}{\Longleftrightarrow} (\ id_{t_1} = id_{t_2}\ )\ AND\ (\ id_{d_1} = id_{d_2}\ )$

Therefore, the *MDmorphemes* are uniquely identified by their two *id* numbers, which are limited respectively by the number of rows and columns in the table. Consequently, the set $\mathscr{MD}$ is finite. This fact is worth noticing, as it will be useful for future proofs.

Theorem 4.2.1. ($\mathscr{MD}$, $\leq$) is a total relation of order.

Proof. This is a total relation of order if $\forall m_1, m_2 \in \mathscr{MD}$,
$(m_1 < m_2)$ OR $(m_1 > m_2)$ OR $(m_1 = m_2)$.
Because $(\mathbb{R}, \leq)$ is a total relation of order and $id_{t_1}, id_{t_2} \in \mathbb{R} \Rightarrow (id_{t_1} < id_{t_2})$ OR $(id_{t_1} \geq id_{t_2})$

1) if $id_{t_1} < id_{t_2} \Rightarrow m_1 < m_2$

2) if $id_{t_1} \geq id_{t_2} \Rightarrow$ if $id_{t_1} > id_{t_2} \Rightarrow m_1 > m_2$

if $id_{t_1} = id_{t_2} \Rightarrow$ if $id_{d_1} < id_{d_2} \Rightarrow m_1 < m_2$

if $id_{d_1} > id_{d_2} \Rightarrow m_1 > m_2$

if $id_{d_1} = id_{d_2} \Rightarrow m_1 = m_2$

$\square$

4.2.3 PVmorphemes

The alphabet will be extended next with a set of special morphemes that depend only on the set of dimensions, not on the elements of data table. The purpose of this extension is to provide the alphabet with more versatility, for representing a variety of visualization methods of multi-dimensional data. These additional morphemes can be used as a correspondent of interaction capabilities such as pivot points.

Definition 4.2.4. The following mapping g between the space of the natural language and the space of the visual language is restricted to the set of dimensions, D.
$g\colon D \to (D, \mathbb{R}^3, RGBA)$, $g(P) = (id_{d_p}, x_p, y_p, z_p, clr_p), \forall P \in D$, where:

each P is one of the dimensions, element of D;

id_d is the column number of dimension P. It is called the *identification number of dimension P*;

x, y, z are the 3D coordinates of the pivot point that corresponds to dimension P in the graphic space;

clr is this point's *RGBA* colour, $clr = (r, g, b, a)$, with $r, g, b, a \in [0, 1]$.

Similarly to the above-defined function *f*, *g* is also a well-defined function, each dimension P having a unique correspondent in the graphic space. Each correspondent *g(P)* of a point P via *g* thus defined is another type of morpheme, as follows:

Definition 4.2.5. A *PVmorpheme* is an element $(id_d, x, y, z, clr) \in (D, \mathbb{R}^3, RGBA)$.

Notation: $\mathscr{PV}$ = the set of all *PVmorphemes*
$\mathscr{PV} = \{$ *PVmorphemes* $\} = \{(id_d, x, y, z, clr) \in (D, \mathbb{R}^3, RGBA)\}$.

PVmorphemes correspond to graphical points with coordinates x, y, z, colour clr and one label stating the dimension's number id_d. Except for the label, all the other components of the morphemes are assigned according to morphological rules, depending on the visualization technique represented.

Similarly to $\mathscr{MD}$, $\mathscr{PV}$ requires a relation of order that is defined as follows:

Definition 4.2.6. $\forall p_1, p_2 \in \mathscr{PV}$
$p_1 < p_2 \overset{def}{\Longleftrightarrow} id_{d_1} < id_{d_2}$
$p_1 = p_2 \overset{def}{\Longleftrightarrow} id_{d_1} = id_{d_2}$.

The above defined relation of order on $\mathscr{PV}$ is also total because it strictly depends on the identification numbers, which are natural numbers, hence totally ordered. Because the set of *id* numbers of the dimensions is finite, $\mathscr{PV}$ is finite as well.

4.2.4 The *FMDVL* alphabet

At this point two sets of totally ordered morphemes have been defined. For a given data table, the *MDmorphemes* provide a correspondence from data table elements into the space of the visual language and, further, to the space of the visual representation. The *PVmorphemes* provide the same correspondence and can act as pivot points in the graphic space of the visual representation. The next step is to define the alphabet, which includes both *MDmorphemes* and *PVmorphemes*.

Definition 4.2.7. The alphabet $\mathscr{A}$ is the totality of all *MDmorphemes* and *PVmorphemes* $\mathscr{A} = \mathscr{PV} \cup \mathscr{MD}$.

Definition 4.2.8. A relation of order "$<$" is defined on $\mathscr{A}$ by extending the relations of order established on $\mathscr{PV}$ and $\mathscr{MD}$ respectively.

$$\begin{aligned}
\forall a_1, a_2 \in \mathscr{A}, a_1 < a_2 \overset{def}{\Leftrightarrow} & (a_1, a_2 \in \mathscr{PV})\, AND\, (a_1 < a_2) \\
& OR\, (a_1, a_2 \in \mathscr{MD})\, AND\, (a_1 < a_2) \\
& OR\, (a_1 \in \mathscr{PV},\, a_2 \in \mathscr{MD} \Rightarrow a_1 < a_2)
\end{aligned}$$

The relation of equality "$=$" is a natural extension of equality on $\mathscr{PV}$ and $\mathscr{MD}$.

Theorem 4.2.2. $(\mathscr{A}, <)$ is a total relation of order.

Proof. $\forall a_1, a_2 \in \mathscr{A}$ the following situations are possible:

$a_1, a_2 \in \mathscr{PV} \Rightarrow (a_1 < a_2)\, OR\, a_1 \geq a_2$ because $<$ is a total relation of order on $\mathscr{PV}$

$a_1, a_2 \in \mathscr{MD} \Rightarrow (a_1 < a_2)$ OR $a_1 \geq a_2$ because $<$ is a total relation of order on $\mathscr{MD}$

$a_1 \in \mathscr{PV}$, $a_2 \in \mathscr{MD} \Rightarrow (a_1 < a_2)$ (from Definition 4.2.8)

$\Rightarrow \forall a_1, a_2 \in \mathscr{A} \Rightarrow (a_1 < a_2) OR (a_1 \geq a_2)$.

□

Therefore, according to the Oxford English Dictionary [34], $\mathscr{A}$ is a well-defined alphabet. To make use of this alphabet in a visual language it is necessary to define morphologies and syntaxes that are capable of describing how and in what conditions these morphemes can be combined, in order to specify visualizations of multi-dimensional data.

4.3 Morphology for Parallel Coordinates

Defining an alphabet is not enough to completely describe a language, as the English alphabet, common to a number of different languages, shows. Therefore, it is essential to define morphological units by grouping morphemes into words according to well defined morphological rules. This is because, depending on the language, the same combination of morphemes can have a different morphological role. As a result, such a morphological unit can be used in a sentence only conforming to syntactical rules specific to that morphological category.

For example, the word “far”, in English is, according to [34], an adjective meaning “distant in space and time” or an adverb meaning “to or at a great or definite distance”. Consequently, it can be used in sentences complying with syntactical rules specific to adjectives and adverbs. In another language using the same English alphabet, the same word “far” can belong to a totally different morphological category and follow different morphological and syntactical rules. In Romanian, for example, “far” is a noun meaning “lighthouse” and complies with morphological and syntactical rules specific to nouns.

In this section, the alphabet $\mathscr{A}$ is used to build the morphology of the Parallel Coordinates Visual Language *PCVL* and illustrate its descriptive capabilities with Parallel Coordinates. The morphological units are called “words”, maintaining the analogy to natural language. According to their morphological role, they will be categorized into two types of structures: basic (similar to nouns and verbs, mandatory in a sentence) and auxiliary (similar to conjunctions and prepositions, that contribute to create more complex sentences).

4.3.1 Basic morphological units

First two morphological units for Parallel Coordinates will be described. They are defined using the *MDmorphemes*, thus being strictly related to the elements of the data table. They will not necessarily act like nouns or verbs, but they will be an important and mandatory part of a sentence.

Definition 4.3.1. A *PCword* is an ordered sequence of distinct *MDmorphemes* that have the same id_t.
$PCword \stackrel{def}{=} (id_t,(m_1,m_2,\ldots,m_{NrVisDim}),clr)$, with $m_j \in \mathscr{MD},\ j \in VisD$
where $\forall m_j,m_k,\ \Rightarrow id_{t_j} = id_{t_k},\ j,k \in VisD$ and $id_{d_j} \neq id_{d_k}$, for $j \neq k,\ \ j,k \in VisD$.

Observations:

- The *MDmorphemes* composing a *PCword* correspond to points from the same row, but from different columns of the data table.
- Since the id_t label is common to all components, it becomes the *PCword*’s id_t.

- For an *MDmorpheme* m_k, the colour is determined by id_{t_k}. Therefore, because all the *MDmorphemes* of a *PCword* have the same id_t, all the corresponding points are visualized with the same colour, clr (see Figure 4.2).

As mentioned in Section 4.2, the coordinates x_k, y_k, z_k, where $k \in VisD$ are assigned based on the visual representation described by a particular morphology. Therefore, because the *PCwords* are morphological units aiming to describe Parallel Coordinates, the coordinates of the corresponding *MDmorphemes* are computed as follows:

- $x_1 = min$, where $min \in \mathbb{R}$ is the minimum x value displayed in the graphical window, depending on the width of the window;

- $x_k = x_{k-1} + d_k$, $k \in \{2, ..., NrVisDim\}$, where $d_k \in \mathbb{R}$ is the distance between the consecutive dimension axes $k-1$ and k;

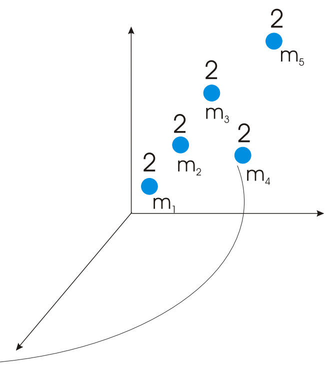
- $y_k = val_k$, $\forall k \in \{1, ..., NrVisDim\}$, where $val_k \in \mathbb{R}$ is computed depending on the data element P_k of the data table as follows:
  $-$ if P_k is a numerical value, then $val_k = s(P_k)$, where $s : \mathbb{R} \rightarrow \mathbb{R}$ can be the identity function, or any scaling function;
  $-$ else, val_k is the correspondent of P_k on an interval or ratio scale;

- $z_k = 0$ initially, because Parallel Coordinates is a 2D visualization. However, having the z coordinate available, the formalism can also represent a 3D visualization;

- Note that the computation of x_k does not depend on id_t. Therefore, the same values are used by all *PCwords*, making the dimension axes parallel.

- Formalizing the distance between two adjacent dimension axes through separate d_k constants allows the flexibility of having the axes equidistant or not;

DIM / TUPLE	1	2	3	4	5
1	30	130	67.2	90	37
2	24	50.1	83.7	39	143
3	57	40	21.5	64.1	23



$w = (2, (m_1, m_2, m_3, m_4, m_5), (0, 0.56, 0.88))$
$m_1 = (2, 1, (10, 24, 10), (0, 0.56, 0.88))$
$m_2 = (2, 2, (20, 50.1, 10), (0, 0.56, 0.88))$
$m_3 = (2, 3, (30, 83.7, 10), (0, 0.56, 0.88))$
$m_4 = (2, 4, (40, 39, 10), (0, 0.56, 0.88))$
$m_5 = (2, 5, (50, 143, 10), (0, 0.56, 0.88))$

Figure 4.2: In the data table T the second row, with id_t=2 is highlighted. The correspondent of this tuple via a *PCword* w is drawn in the graphic space on the right. Each point corresponding to an *MDmorpheme* is labelled with its id_t and is coloured with the same colour, having the *RGB* value (0, 0.56, 0.88). The x coordinates are equidistant, while z is fixed, equal to 10.

Theorem 4.3.1. In any *PCword* each visible dimension has a corresponding *MDmorpheme*.

Proof. By definition, a *PCword* $w = (id_t, (m_1, m_2, \ldots, m_{NrVisDim}), clr)$ has *MDmorphemes* with indices from 1 to $NrVisDim$. Hence the set of these indices is exactly *VisD*. A function f can be defined, $f : VisD \to VisD$, $f(k) = id_{d_k}$, where id_{d_k} is the identification number of the dimension corresponding to m_k.

From Definition 4.3.1 $\Rightarrow f(k) \neq f(j), \forall k \neq j,\ k, j \in VisD \Rightarrow f$ is injective [1]. On the other hand, because *VisD* is finite $\Rightarrow f$ is injective $\Leftrightarrow f$ is surjective [2] $\Leftrightarrow f$ is bijective. Therefore f is surjective and that means $\forall j \in VisD\ \exists!\ k$ [3] such that $f(k) = j \Rightarrow \forall j \in VisD\ \exists!\ k$ such that m_k corresponds to the j^{th} dimension. $\Rightarrow$ any *PCword* has one unique *MDmorphem* for each visible dimension. □

To provide a more flexible way of building the *PCwords* it is essential to take into consideration the order the *MDmorphemes* have in each morphological unit.

Definition 4.3.2. $p\colon VisD \to VisD$, p bijective. We say that p is a *permutation* of *VisD*.

Notation: Let p be a permutation of *VisD*.

Then a *PCword* $w = (id_t,\ (m_1, m_2, \ldots,\ m_{NrVisDim}), clr)$, where id_{d_k} is the dimension corresponding to m_k, can be written $w = (id_t,\ m_{p(1)} m_{p(2)} \ldots\ m_{p(NrVisDim)}, clr)$, given that $p(k) = id_{d_k}$.

Notation: $\mathscr{PCW}$ = the set of all *PCwords*

$\mathscr{PCW} = \{(id_t, m_{p(1)} m_{p(2)} \ldots m_{p(NrVisDim)}, clr)\ \mid id_t \in \{1, ..., NrVisTuples\},\ m_{p(k)} \in \mathscr{MD},\ k = \{1, ..., NrVisDim\},\ p =$ permutation of *VisD*$\}$.

From the definition of the *PCwords* it follows that a *PCword* corresponds to the points that visualize one row of the data table. However, in Parallel Coordinates these points are connected by a polyline. Therefore, to accurately formalize a Parallel Coordinates visualization, it is also necessary to introduce morphological units capable of describing the polylines. Because they are based on the data table, these new units are also considered basic morphological units.

[1] injective = one-to-one mathematical function

[2] surjective = onto mathematical function

[3] there exists an unique k

Definition 4.3.3. Let $w = (id_{t_w}, m_{p(1)}m_{p(2)}\dots m_{p(NrVisDim)}, clr_w)$ be a *PCword*, where p is a permutation of *VisD*.

A *PCline of the PCword* $w = (id_{t_w}, \bigcup_{i=1}^{NrVisDim-1} [m_{p(i)}, m_{p(i+1)}], clr_w)$.

Observations:

- The *PCword*'s id_t and colour will be assigned to the corresponding *PCline*.
- The correspondent of a *PCline* in the space of the visual representation is the polyline that connects all the points that visualize the *MDmorphems* composing the *PCword* (see Figure 4.3).

Notation: $\mathscr{PCL}$ =the set of all *PClines*

$\mathscr{PCL} = \{$ *PCline of the PCword* $w \mid w \in \mathscr{PCW}\}$.

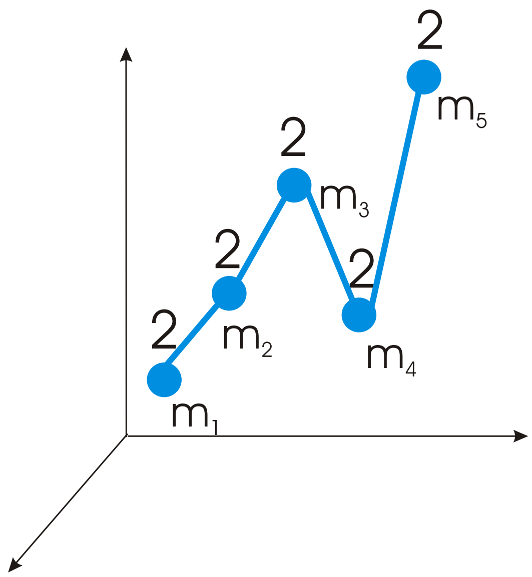


Figure 4.3: A *PCline* links the *MDmorphems* of a *PCword*.

4.3.2 Auxiliary morphological units

PCwords and *PClines* are morphological units based solely on $\mathscr{MD}$. Using the rest of the alphabet is essential for a proper formal description of the Parallel Coordinates representation. Next, the category of auxiliary morphological units will be built employing elements of $\mathscr{PV}$. First, an equivalent of the *PCwords* is defined, representing the pivot points.

Definition 4.3.4. A *PVword* is an ordered sequence of *PVmorphemes*.
$PVword = ((v_1, v_2, \ldots, v_{NrVisDim}), clr)$, with $v_j \in \mathscr{PV},\ j \in \{1,...,NrVisDim\}$, and $id_{d_j} \neq id_{d_k}, \forall j \neq k,\ j,k \in VisD$.

The colour clr used to visualize a *PVword* is the same colour used for each component v_k (see Figure 4.4). Note that a *PVword* does not have any id. Theorem 4.3.2 will show why it was not necessary to introduce one.

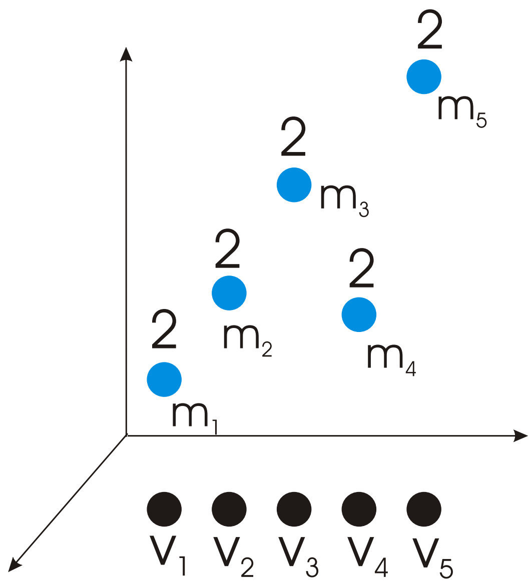


Figure 4.4: The visual representation of a *PCword* and its corresponding *PVword*.

The order of dimensions, reflected in the order of *MDmorphemes* in a *PCword*, determines also the order of PVmorphemes in a *PVword*. Therefore, using the same permutation p of *VisD*, a *PVword* can be written as follows:

Notation: Let p be a permutation of *VisD*. A *PVword* $vw = ((v_1, v_2, \ldots, v_{NrVisDim}), clr)$ can be written $vw = (v_{p(1)}v_{p(2)} \ldots v_{p(NrVisDim)}, clr)$, with $p(k) = id_{d_k}$, $k \in VisD$, where id_{d_k} is the dimension corresponding to m_k.

Theorem 4.3.2. For each permutation p of *VisD* there is only one *PVword*.

Proof. Similar with the proof of Theorem 4.3.1 it is possible to show that all the visible dimensions are represented by a *PVmorpheme* in a *PVword*. But there are only *NrVisDim* *PVmorphemes* available, hence the sole difference between any two *PVwords* consists in the order the *PVmorphemes* are used. □

Notation: $\mathscr{PVW}$ = the set of all *PVwords*

$\mathscr{PVW}$ =$\{(v_{p(1)}v_{p(2)} \ldots v_{p(NrVisDim)}, clr) \mid v_{p(k)} \in \mathscr{PV}, k = \{1, ..., NrVisDim\},\ p$ = permutation of *VisD*$\}$.

Similar to a *PCline*, the following auxiliary morphological unit complements the *PVwords*, providing formal support for the full description of the pivot line in Parallel Coordinates.

Definition 4.3.5. Let $vw = (v_{p(1)}v_{p(2)} \ldots v_{p(NrVisDim)}, clr)$ be a *PVword*, where p is a permutation of *VisD*.

A *PVline for the PVword* $vw = (\bigcup_{i=1}^{NrVisDim-1} [v_{p(i)}, v_{p(i+1)}], clr)$.

According to this definition, a *PVline* corresponds to the polyline that connects the points representing the *PVmorphemes* that compose the *PVword* and it is visualized with the same colour as the *PVmorphemes* (see Figure 4.5). The order of dimensions is respected by connecting the *PVmorphemes* in the order they appear in the *PVword*, order given by the permutation p.

From Theorem 4.3.2 it follows that, for a given permutation p of *VisD*, there is only one *PVline*.

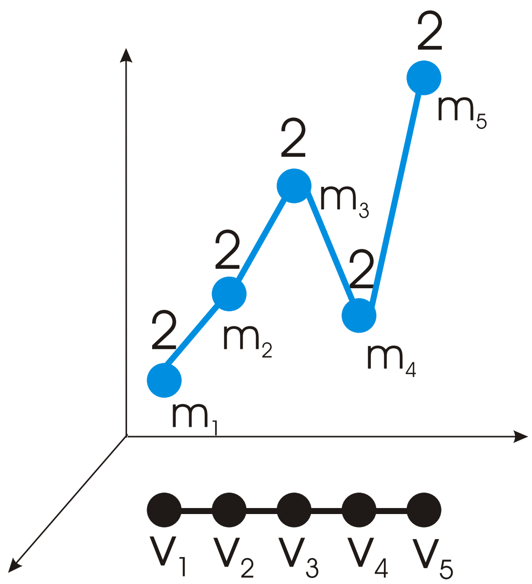


Figure 4.5: The visual representation of a *PCline* and its corresponding *PVline*.

In Parallel Coordinates the pivot line is connected with the tuples and the points representing elements of the data table through a set of parallel axes. Translated into *PCVL*, this set of axes is defined through the following auxiliary morphological unit.

Definition 4.3.6. Let $w = (id_{t_w}, m_{p(1)}m_{p(2)}\dots m_{p(NrVisDim)}, clr_w)$ be a *PCword* and $vw = (v_{p(1)}v_{p(2)}\dots v_{p(NrVisDim)}, clr)$ be a *PVword*, where p is a permutation of *VisD*.

A *PCax of the PCword* $w = (\bigcup_{i=1}^{NrVisDim} [v_{p(i)}, m_p(i)], clr_{ax})$.

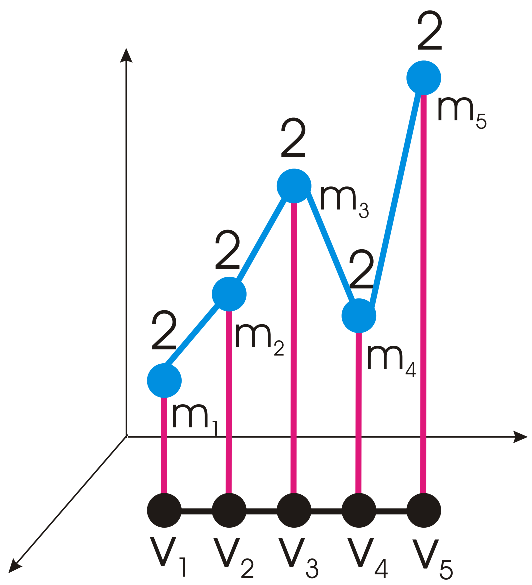


Figure 4.6: Parallel Coordinates representation showing a *PCax* that connects the *MDmorphems* of a *PCline* with the corresponding *PVmorphems* of the *PVline*.

Observations:

- Graphically, a *PCax* (see Figure 4.6) is associated to the set of segments that connect each point representing an *MDmorpheme* of the *PCword* with the corresponding *PVmorpheme* of the *PVword*.

- The colour clr_{ax} is common to all *PCaxes*, but independent of other units' colour.

- The segments composing a *PCax* are parallel to each other, because of the way the coordinates have been initially assigned for the *MDmorphemes* of the *PCword* and for the *PVmorphemes* of the corresponding *PVword*.

Notation: $\mathscr{PCA}$ = the set of all *PCaxes* $=\{$ *PCax of the PCword w* $\mid w \in \mathscr{PCW}\}$.

So far, in this section, the morphological units of *PCVL* have been introduced. The *PCwords* and *PClines* are the basic units and are built with *MDmorphemes*. The *PVwords*, *PVlines* and *PCaxes* are auxiliary units and may also use the *PVmorphemes*.

4.4 Syntax for Parallel Coordinates

Now that the morphological units are created, the following step is to combine them into sentences. This section describes the syntax of *PCVL*, which has rules to decide whether a certain combination of existing morphological units is grammatically correct. This approach of defining syntactical rules is closer to analytical grammars than to generative grammars, which have a set of rules to recursively produce new expressions, starting from a set of symbols. The rules introduced in this section have been established in analogy to natural languages, while keeping in mind that *PCVL* is intended to describe the Parallel Coordinates visual representation. The morphological units introduced in the previous section hold the structure necessary to describe Parallel Coordinates. However, as Parallel Coordinates alone cannot eliminate occlusions, this section will only establish rules that determine when occlusions appear. Solutions for avoiding occlusions will be provided in Chapter 5, where a formal representation of Parallel Glyphs and clustering will be detailed.

Definition 4.4.1. A *sentence* is a combination of morphological units.

Rule PC.1. A sentence contains at least one *PCword* or one *PCline*.

Rule PC.2. A morphological unit can be used at most once in a sentence.

In any written language, it is necessary, in general, to see the entire word, in order to correctly read and understand it. This convention is applied by the next rule to the *PCVL* language. First, the following definitions set up the terminology.

Definition 4.4.2. An *MDmorpheme occlusion* occurs when for $m_{p(1)}m_{p(2)}\dots m_{p(NrVisDim)} \in \mathscr{MDW}$ such that $\exists\, m'_{p(1)}m'_{p(2)}\dots m'_{p(NrVisDim)} \in \mathscr{MDW}$ and $\exists\, k \in VisD$ with $m_{p(k)}, m'_{p(k)} \in \mathscr{MD}$, having the coordinates $x_{p(k)} = x'_{p(k)}$ and $y_{p(k)} = y'_{p(k)}$. If $z_{p(k)} \leq z'_{p(k)}$ then $m_{p(k)}$ *is occluded by* $m'_{p(k)}$.

Definition 4.4.3. An *MDword* $m_{p(1)}m_{p(2)}\dots m_{p(NrVisDim)} \in \mathscr{MDW}$ is considered *MDocclusion-free* if there are not two consecutive, occluded *MDmorphems*.
$m_{p(1)}m_{p(2)}\dots m_{p(NrVisDim)} \in \mathscr{MDW}$ is *MDocclusion-free* $\overset{def}{\Longleftrightarrow} \nexists\, k \in \{1,\dots,NrVisDim - 1\}$ such that $m_{p(k)}$ and $m_{p(k+1)}$ are occluded.

Definition 4.4.4. An *MDsegment occlusion* occurs when for $m_{p(1)}m_{p(2)}\dots m_{p(NrVisDim)} \in \mathscr{MDW}$ and $\exists\, m'_{p(1)}m'_{p(2)}\dots m'_{p(NrVisDim)} \in \mathscr{MDW}$ such that $\exists\, k \in VisD$ with $m_{p(k)} \in \mathscr{MD}$ occluded by $m'_{p(k)} \in \mathscr{MD}$ and $m_{p(k+1)} \in \mathscr{MD}$ occluded by $m'_{p(k+1)} \in \mathscr{MD}$.

Definition 4.4.5. Let $m_{p(1)}m_{p(2)}\dots m_{p(NrVisDim)} \in \mathscr{MDW}$ be an *MDword* such that $\exists\, k \in VisD$ and an *MDsegment occlusion* occurs in position $p(k)$.
Then $\forall\, m'_{p(1)}m'_{p(2)}\dots m'_{p(NrVisDim)} \in \mathscr{MDW}$,
$m'_{p(k+1)} \in \mathscr{MD}$ is a *possible path* for $m_{p(k+1)} \in \mathscr{MD} \overset{def}{\Longleftrightarrow} m'_{p(k)} = m_{p(k)}$.

Definition 4.4.6. Let $m_{p(1)}m_{p(2)}\dots m_{p(NrVisDim)} \in \mathscr{MDW}$ be an *MDword* such that $\exists\, k \in VisD$ and an *MDsegment occlusion* occurs in position $p(k)$.
If the number of possible paths =1 then there is a *non ambiguous MDsegment occlusion*. Otherwise there is an *ambiguous MDsegment occlusion*.

Rule PC.3. A sentence where an ambiguous *MDsegment occlusion* occurs is grammatically incorrect.

A discussion about the suitability of this rule is appropriate here. While this is a common-sense rule, derived from natural languages, in the particular context of Parallel

Coordinates visualization it is extremely strong. A sentence is likely to be ruled out as grammatically incorrect, even if it is composed of a small number of morphological units (as previously discussed in Chapter 3). The obvious question is then whether the rule should be changed.

The only argument against the rule is that it is very difficult for a sentence to be declared grammatically correct. As we know, Parallel Coordinates are used to visualize thousands of data elements. Therefore, such a restrictive rule would make *PCVL* a poor formal description of the visualization.

On the other hand, waiving this rule would lead to ambiguous combinations of morphemes, even if they would provide an accurate formal description of the visualization. This fact is an example of the role that a formalism can play in revealing aspects of the visual representation. In this case, the aspect revealed is one of the important limitations of Parallel Coordinates, the overlapping of lines, as discussed in Chapter 3.

Two possible solutions of this problem are discussed in Chapter 5. One is based on Parallel Glyphs (presented in Chapter 3). The other looks at how this formalism can describe the polylines clustering. This common approach to mitigating line overlapping has been introduced in Chapter 2 and will be described using this linguistic formalism in Chapter 4.

4.5 Morphology for Star Glyphs

This section will illustrate the versatility of the alphabet $\mathscr{A}$ by developing the morphology of a new visual language that describes another visual representation of multi-dimensional data. The visual language is called *SGVL*, as it represents the Star Glyphs visualization.

The fact that another visual language can be elaborated starting from $\mathscr{A}$ underlines the importance of this alphabet. This also shows that there is a family of multi-dimensional representations described by visual languages based on $\mathscr{A}$, *FMDVL*.

4.5.1 Basic morphological unit

The morphology for Star Glyphs is analogue to the morphology for Parallel Coordinates. First, the basic morphological unit, or an *SGword*, is defined. An *SGword* is the equivalent of an *PCword*, with the difference that it corresponds to one column in contrast to one row.

Definition 4.5.1. An *SGword* is an ordered sequence of distinct *MDmorphemes* that have the same id_d.

$SGword = (id_d, (m_1, m_2, \ldots, m_{NrVisTuples}), clra)$,

with $m_j \in \mathscr{MD}$, $j \in \{1,...,NrVisTuples\}$, where $\forall m_k, m_j \Rightarrow id_{d_k} = id_{d_j}$, $k, j \in VisE$ and $id_{t_k} \neq id_{t_j}$, $\forall k \neq j$, $k, j \in VisE$.

Observations:

- Since the id_d label is common to all components, it becomes the *SGword*'s id_d.
- For an *MDmorpheme* m_k, the colour is determined by id_{t_k}. Therefore, because the *MDmorphemes* of an *SGword* have the distinct id_t, all the corresponding points are visualized with a different colour. The *SGword*'s colour $clra$ is then an array with all the colours of its component *MDmorphemes*(see Figure 4.7).

As mentioned in Section 4.2, the coordinates x_k, y_k, z_k, $k \in VisD$ of the *MDmorphemes* are assigned by a morphology according to the corresponding visual representation.

Therefore, because the *SGwords* are morphological units that describe Star Glyphs, the coordinates of the *MDmorphemes* are computed as follows:

- $\alpha_k = 360 \div NrVisTuples$;

- val_k is defined as follows, $\forall k \in \{1,...,NrVisTuples\}$, where $val_k \in \mathbb{R}$ is computed depending on the data element P_k of the data table corresponding to the *MDmorpheme* m_k:
  − if P_k is a numerical value, then $val_k = s(P_k)$, where $s : \mathbb{R} \rightarrow \mathbb{R}$ can be the identity function, or any scaling function;
  − else, val_k is the correspondent of P_k on an interval or ratio scale;

- $x_k = val_k * \cos(k * \alpha_k) + trans_x$, $\forall k \in \{1,...,NrVisTuples\}$, where $trans_x \in \mathbb{R}$ is a translation factor on the x axis;

- $y_k = val_k * \sin(k * \alpha_k) + trans_y$, $\forall k \in \{1,...,NrVisTuples\}$, where $trans_y \in \mathbb{R}$ is a translation factor on the y axis;

- Formalizing the angle between two adjacent spikes through separate α_k constants allows the flexibility of having the spikes uniformly distributed or not. Initially, these angles are identical;

- $z_k = 0$ or has any fixed value, because Star Glyphs are a 2D visualization. However, having the z coordinate available, the formalism can also represent a 3D visualization;

Theorem 4.5.1. In any *SGword* each visible tuple has a corresponding *MDmorpheme*.

Proof. The proof is similar to the proof of Theorem 4.3.1 □

Notation: Let $q : VisE \rightarrow VisE$ be a permutation of *VisE*.

Then a *SGword* $gw = (id_d,\ (m_1, m_2, \ldots,\ m_{NrVisTuples}),\ \ clra)$ where id_{t_k} is the tuple corre-

DIM / TUPLE	1	2	3	4	5
1	30	130	67.2	90	37
2	24	50.1	83.7	39	143
3	57	40	21.5	64.1	23

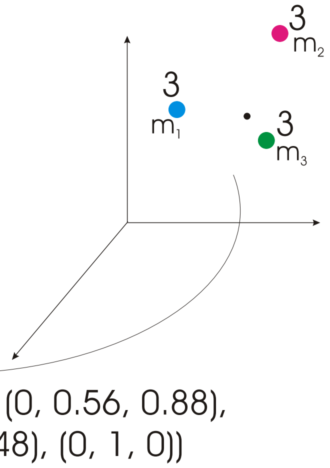


Figure 4.7: On the left, the *SGword*'s corresponding data dimension column; on the right, the *SGword*'s corresponding points in the visual representation space. The points representing the *MDmorphemes* are labelled according to their column and coloured uniquely according to their id_t.

sponding to m_k can be written $(id_d, m_{p(1)}m_{p(2)} \dots\ m_{p(NrVisTuples)}, clra)$, where

$q(k) = id_{t_k}$.

Notation: $\mathscr{SGW}$ = the set of all *SGwords*

$\mathscr{SGW}$ = $\{m_{p(1)}m_{p(2)}\dots m_{p(NrVisTuples)} \mid m_{p(k)} \in \mathscr{MD},\ k = \{1,...,NrVisTuples\},\ p =$ permutation of $VisE\}$.

4.5.2 Auxiliary morphological units

At this point in *SGVL* is included one of *PCVL*'s morphological units, *PVwords*. The auxiliary morphological units, necessary to completely formalize the Star Glyphs, are defined as follows.

Definition 4.5.2. Let $gw = (id_{d_{gw}}, m_{p(1)}m_{p(2)} \dots m_{p(NrVisTuples)},\ clr)$ be a *SGword*, where p is a permutation of *VisD*. Let v be the *PVmorpheme* that corresponds to dimension $id_{d_{gw}}$. An *SGfan of the SGword gw* is defined as $(id_{d_{gw}}, \bigcup_{i=1}^{NrVisTuples} [v,\ m_{p(i)}],\ clr_f)$ where the *SGword*'s $id_{d_{gw}}$ is also used by the *SGfan*.

Graphically, an *SGfan* is associated to the set of segments that connect the pivot point with each of the points that visualize the *MDmorphemes* of the *SGword*. These segments are visualized with the same colour clr_f, independent of the individual colours of the *MDmorphemes* that determine the *SGfan* (see Figure 4.8).

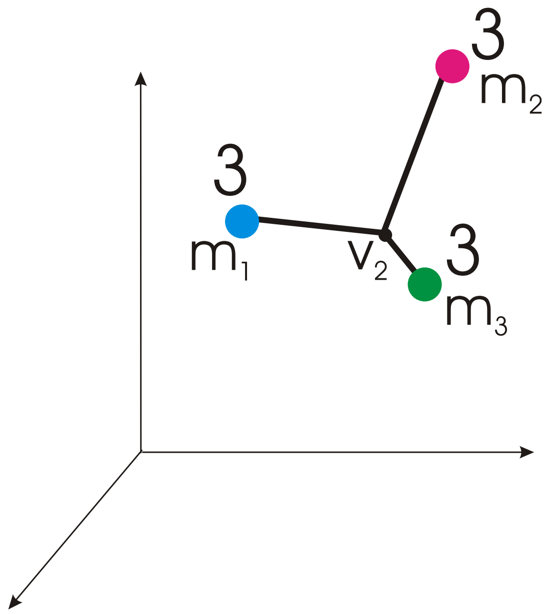


Figure 4.8: An emerging Star Glyph showing its centre, corresponding to the *PVmorpheme* v_2, its multi-coloured points corresponding to the *SGword* and its spikes corresponding to the *SGfan*.

Notation: $\mathscr{SGF}$ the set of all *SGfans*= $\{SGfan\ of\ the\ SGword\ gw \mid gw \in SGW\}$.

Definition 4.5.3. Let $gf = (id_{d_{gf}}, \bigcup_{i=1}^{NrVisTuples} [v,\ m_{p(i)}],\ clr_f)$ be a *SGfan* of the *SGword* $gw\ =\ (id_{d_{gf}}, m_{p(1)}m_{p(2)}\dots m_{p(NrVisTuples)}, clra)$.

An *SGglyph of the SGword gw* is defined as follows:
$gg\ =\ (id_{d_{gf}}, \bigcup_{i=1}^{NrVisTuples-1} \Delta v m_{p(i)} m_{p(i+1)}, clr_g)$.

Naturally, the *SGglyph* has the same $id_{d_{gf}}$ as the *SGfan* gf and the *MDmorphemes* $m_k \in \mathscr{MD}$, $k \in \{1, \dots, NrVisTuples\}$ of the *SGword* gw.

Graphically, the *SGglyph* (see Figure 4.9) represents the surface obtained as the union of all filled-triangles, each triangle having the corresponding pivot point as one vertex, and pairs of points that visualize consecutive *MDmorphemes* as the other two vertices.

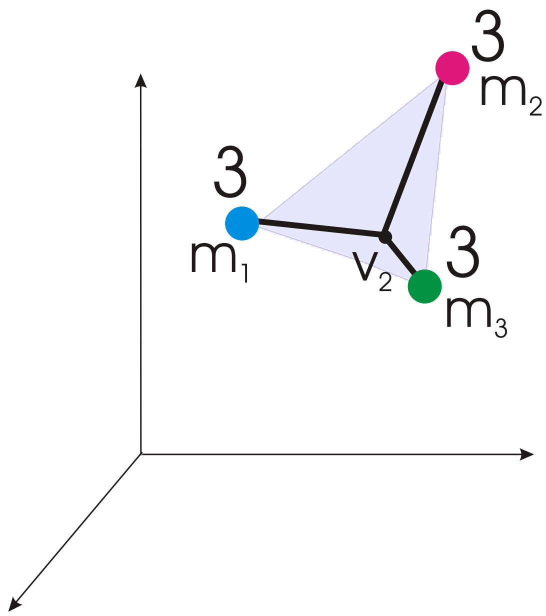


Figure 4.9: The corresponding of a *SGglyph* in the space of the visual representation.

Notation: $\mathscr{SGG}$ = the set of all *SGglyphs* = $\{$ *SGglyph of the SGword gw* $\mid gw \in \mathscr{SGW} \}$.

Definition 4.5.4. Let gg be a *SGglyph* of the *SGword* $gw = (id_{d_{gf}}, m_{p(1)}m_{p(2)}\ldots m_{p(NrVisTuples)}, clr_g)$, $gg = (\ id_{d_{gf}}, \bigcup_{i=1}^{NrVisTuples-1} \Delta v m_{p(i)}m_{p(i+1)}, clr_g)$.
The *SGborder of the SGglyph gg* is defined as $(id_{d_{gf}}, \bigcup_{i=1}^{NrVisTuples-1} [m_{p(i)}m_{p(i+1)}], clr_b)$.

Similar to the *SGglyph*, the *SGborder* keeps the same $id_{d_{gf}}$. The *SGborder* represents the polyline that connects all the points that visualize consecutive *MDmorphemes* of the *SGglyph* (see Figure 4.10).

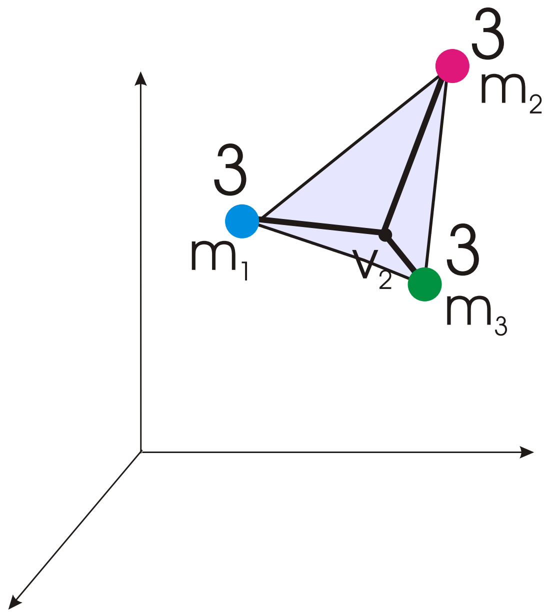


Figure 4.10: A complete Star Glyph with its *SGglyph* and *SGborder*.

Notation: $\mathscr{SGB}$ = the set of all *SGborders* = $\{$ *SGborder of the SGglyph gg* $\mid gg \in \mathscr{SGG}\}$.

In conclusion, the morphological units of *SGVL* are: *SGwords*, *SGfans*, *SGglyphs*, *SGborders*.

4.6 Syntax for Star Glyphs

The syntax of SGVL is defined in strict analogy with the syntax of PCVL.

Rule SG.1. Any sentence contains at least one *SGword*.

Rule SG.2. Any morphological unit is used at most once in a sentence.

Definition 4.6.1. A *SGmorpheme occlusion* occurs when for $m_{p(1)}m_{p(2)}\ldots m_{p(NrVisTuples)} \in \mathscr{SGW} \Longrightarrow$ $\exists m'_{p(1)}m'_{p(2)}\ldots m'_{p(NrVisTuples)} \in \mathscr{SGW}$ and $\exists k, j \in VisE$ with $m_{p(k)} \in \mathscr{MD}$ and $m'_{p(j)} \in \mathscr{MD}$ having the coordinates $x_{p(k)} = x'_{p(j)}$ and $y_{p(k)} = y'_{p(j)}$. If $z_{p(k)} \leq z'_{p(j)}$ then $m_{p(k)}$ is occluded by $m'_{p(j)}$.

Observe here the difference between *MDmorpheme occlusion* and *SGmorpheme occlusion*. The *SGmorpheme occlusion* can occur by the overlapping of *MDmorphemes* from either the same *SGword* or from a different *SGword*, which represents another dimension. On the other hand, *MDmorpheme occlusions* can occur only by overlapping of *MDmorphemes* from different *PCwords*.

Definition 4.6.2. An *SGword* $m_{p(1)}m_{p(2)}\ldots m_{p(NrVisTuples)} \in \mathscr{SGW}$ is considered *SGmorpheme occlusion-free* if it does not contain two consecutive, occluded *MDmorphemes*.
$m_{p(1)}m_{p(2)}\ldots m_{p(NrVisTuples)} \in \mathscr{SGW}$ is *SGmorpheme occlusion-free* $\overset{def}{\Longleftrightarrow}$
$\nexists k \in \{0, ..., NrVisTuples-1\}$ such that $m_{p(k)}$ and $m_{p(k+1)}$ are occluded.

Rule SG.3. A sentence where an *SGmorpheme occlusion* occurs is not grammatically correct.

The difference between the PCVL syntax and the SGVL syntax is that the third rule is not so restrictive for Star Glyphs. This fact is not surprising, because occlusion is usually not a problem for Star Glyphs. Another possible rule for the *SGVL* syntax could concern the angle α between two consecutive spikes of a *SGfan*. However, deciding the minimum acceptable value for this angle would depend on the human eye's capability of distinguishing graphical elements that are very close to each other. This is beyond the scope of this thesis.

4.7 Summary

This chapter has presented a method for formalizing visual representations of multi-dimensional data. First, an alphabet $\mathscr{A}$ of two types of ordered morphemes has been built. This alphabet has the capability of describing a number of visual languages, each having a particular morphology and syntax, depending on the corresponding visual representation. To illustrate this capability, two different visual languages have been detailed through their individual morphology and syntax, based on the alphabet $\mathscr{A}$. One such visual language describes Parallel Coordinates, the other describes Star Glyphs, and they are both part of the Family of Multi-Dimensional Visual Languages *FMDVL*.

Chapter 5

Applications of the Family of Multi-Dimensional Visual Languages

This chapter applies the Family of Multi-Dimensional Visual Languages (*FMDVL*) to Parallel Glyphs and a Parallel Coordinates clustering technique. An alphabet $\mathscr{A}$ of *FMDVL* has been introduced in Chapter 4, as composed of two types of ordered morphemes. The linguistic formalism founded on the alphabet $\mathscr{A}$ has the flexibility to describe visual representations of multi-dimensional data that can be stored in row, column tabular format, and possible interactions available in these visualizations. In Chapter 4, based on this alphabet, two visual languages have been developed to describe two visual representations of multi-dimensional data, one for Parallel Coordinates, and one for Star Glyphs. However, these are not the only possible applications of *FMDVL*.

To illustrate the extended capabilities of this formalism, this chapter presents two other applications based on $\mathscr{A}$. First, a new visual language is introduced. It describes Parallel Glyphs, the visual representation of multi-dimensional data developed in Chapter 3. One goal of Parallel Glyphs was to address the problem of overlapping polylines in Parallel Coordinates. As discussed in Chapter 2, a common way to deal with this limitation of Parallel Coordinates is to cluster the polylines, according to a user-defined measure of similarity. Therefore, the second application of this formalism is a description of this clustering technique.

5.1 Morphology for Parallel Glyphs

To define the visual language that describes Parallel Glyphs, *PGVL*, it is necessary to provide details of the corresponding morphology and syntax. Because *PGVL* is part of the *FMDVL*, its morphological units are built using elements of the alphabet $\mathscr{A}$. The morphology of *PGVL* is built with morphological units of the *PCVL* and *SGVL* morphologies that have been defined in Chapter 4. This is a natural consequence of the fact that Parallel Glyphs are an integration of Parallel Coordinates and Star Glyphs. Therefore, all the morphological units previously defined for these two visual languages are also part of the *PGVL* morphology, as follows.

5.1.1 Morphological units imported from *PCVL*

Since Parallel Coordinates are the starting point for Parallel Glyphs, all morphological units defined in Chapter 4 Section 4.3 for *PCVL* can be used by the morphology describing Parallel Glyphs.

1. A *PCword* is an ordered sequence of distinct *MDmorphemes* that have the same id_t. It is visualized by the points representing these *MDmorphemes* (see Figure 5.1). In a tabular format, these points correspond to one row.
   $PCword \stackrel{def}{=} (id_t, (m_1, m_2, \ldots, m_{NrVisDim}), clr)$, with $m_j \in \mathscr{MD},\ j \in VisD$
   where $\forall m_j, m_k, \Rightarrow id_{t_j} = id_{t_k},\ j,k \in VisD$ and $id_{d_j} \neq id_{d_k}$, for $j \neq k,\quad j,k \in VisD$.

2. A *PCline* of a *PCword* w is the polyline that connects the points representing the *MDmorphemes* composing the *PCword* (see Figure 5.2).
   $PCline \stackrel{def}{=} (id_{t_w}, \bigcup_{i=1}^{NrVisDim-1} [m_i,\, m_{i+1}],\, clr_w)$.

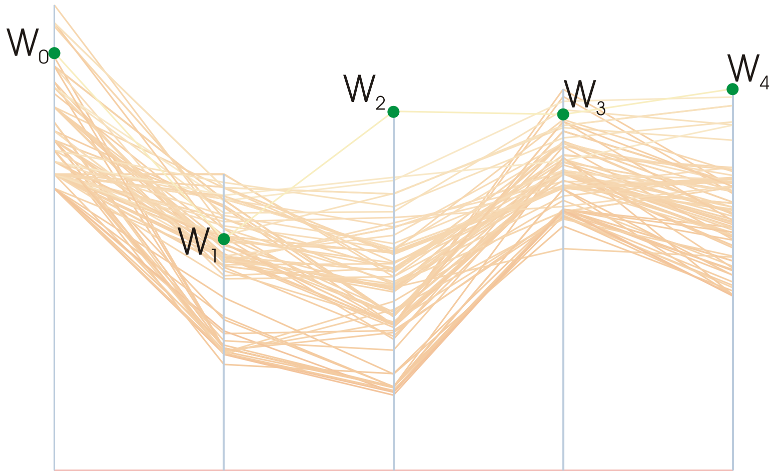


Figure 5.1: PCword in Parallel Coordinates visualization.

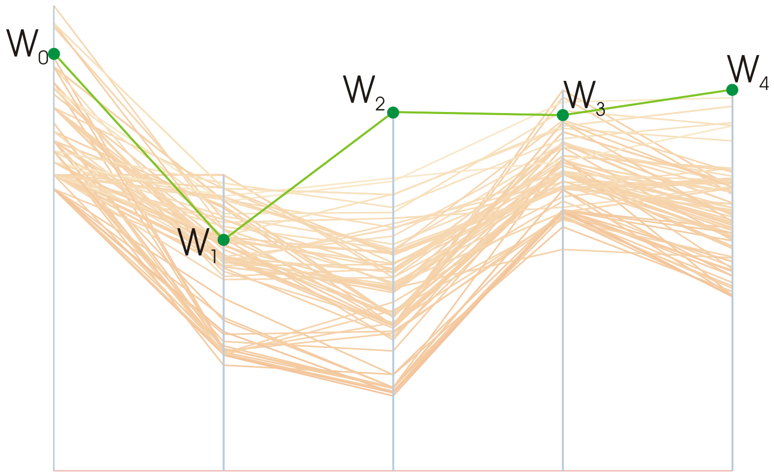


Figure 5.2: PCline in Parallel Coordinates visualization.

3. A *PVword* is an ordered sequence of *PVmorphemes*. It is visualized by the pivot points corresponding to these *PVmorphemes* (see Figure 5.3).

   $PVword \stackrel{def}{=} ((v_1, v_2, \ldots, v_{NrVisDim}), clr)$, with $v_j \in \mathscr{PV}$, $j \in VisD$, and $id_{d_j} \neq id_{d_k}$, $\forall j \neq k,\ j,k \in VisD$.

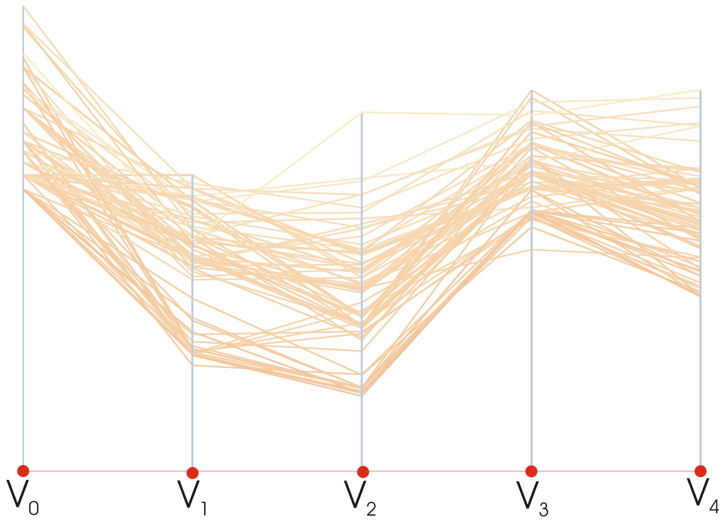


Figure 5.3: PVword in Parallel Coordinates visualization.

4. A *PVline* of the *PVword vw* is the polyline that connects the points representing the *PVmorphemes* composing the *PVword* (see Figure 5.4).
$PVline \stackrel{def}{=} (\bigcup_{i=1}^{NrDim} [v_i, v_{i+1}], clr).$

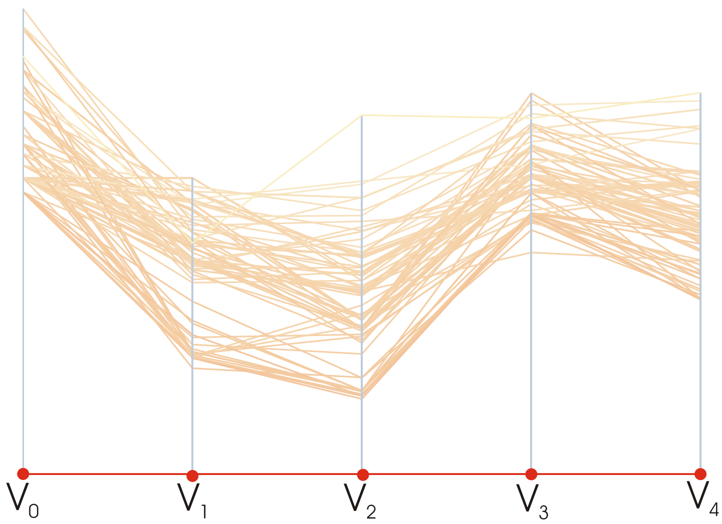


Figure 5.4: PVline in Parallel Coordinates visualization.

5. A *PCax* of the *PCword* w is the set of axes that connect the points representing the *MDmorphemes* of w with the corresponding pivot points (see Figure 5.5).
$PCax \stackrel{def}{=} (\bigcup_{i=1}^{NrVisDim} [v_i, m_i], clr_{ax}).$

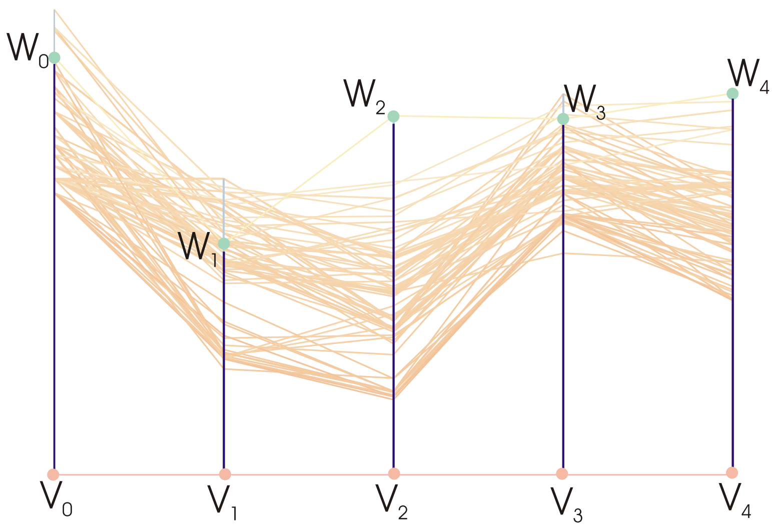


Figure 5.5: PCax in Parallel Coordinates visualization.

5.1.2 Morphological units imported from *SGVL*

The morphological units defined in Chapter 4, Section 4.5 for *SGVL* will also be used by *PGVL*. This is possible because each glyph of Parallel Glyphs visualization is actually a Star Glyph. Next, the graphical correspondence between these morphological units and Parallel Glyphs will be illustrated.

1. An *SGword* is an ordered sequence of distinct *MDmorphemes* that have the same id_d. It is visualized by the points representing these *MDmorphemes* (see Figure 5.6). In a tabular format, these points correspond to one column.
$SGword \stackrel{def}{=} (id_d, (m_1, m_2, \ldots, m_{NrVisTuples}), clra),$

with $m_j \in \mathscr{MD},\ j \in VisE,$ where $\forall m_k, m_j \Rightarrow id_{d_k} = id_{d_j},\ k, j \in VisE$ and $id_{t_k} \neq id_{t_j},\ \forall k \neq j,\ k, j \in VisE.$

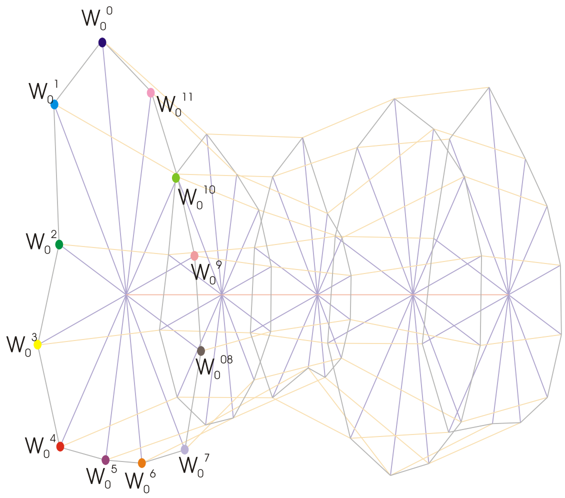


Figure 5.6: SGword in Parallel Glyphs visualization.

2. An *SGfan* of an *SGword gw* is the set set of axes that connect the points representing the *MDmorphemes* of *gw* with the corresponding pivot point (see Figure 5.7).
$SGfan \stackrel{def}{=} (id_{d_{gw}}, \bigcup_{i=1}^{NrVisTuples} [v,\ m_i],\ clr_f).$

3. An *SGborder* of the *SGglyph gg* is the polyline that connects the points representing the *MDmorphemes* of *gg* (see Figure 5.8).
$SGborder \stackrel{def}{=} (id_{d_{gf}}, \bigcup_{i=1}^{NrVisTuples-1} [m_i, m_{i+1}], clr_b).$

4. An *SGglyph* of the *SGword gw* is the polygonal surface determined by the points representing the *MDmorphemes* of *gw* (see Figure 5.9).
$SGglyph \stackrel{def}{=} (id_{d_{gf}}, \bigcup_{i=1}^{NrVisTuples-1} \Delta v m_i m_{i+1}, clr_g).$

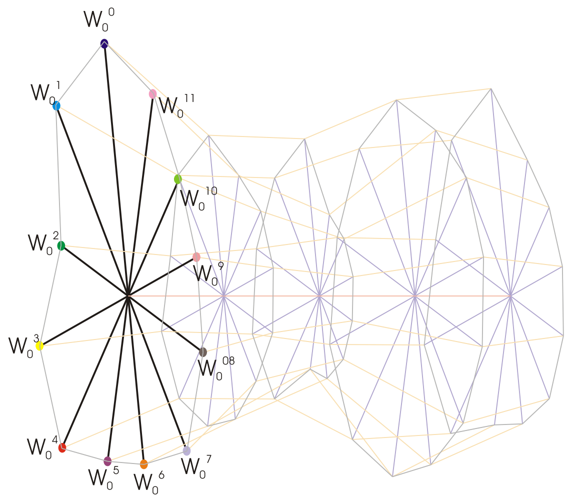


Figure 5.7: SGfan in Parallel Glyphs visualization.

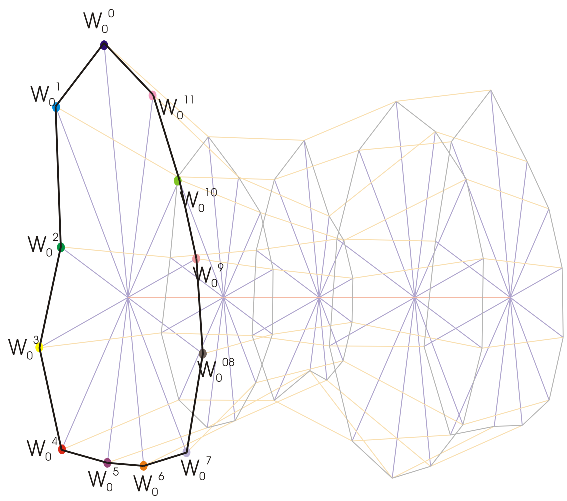


Figure 5.8: SGborder in Parallel Glyphs visualization.

5.1.3 Topological transformations of the morphological units

The new language *PGVL* not only imports these morphological units from *PCVL* and *SGVL*, but it also modifies them to better describe the corresponding visual representation of multi-dimensional data. These modifications are mathematically described by topological transformations.

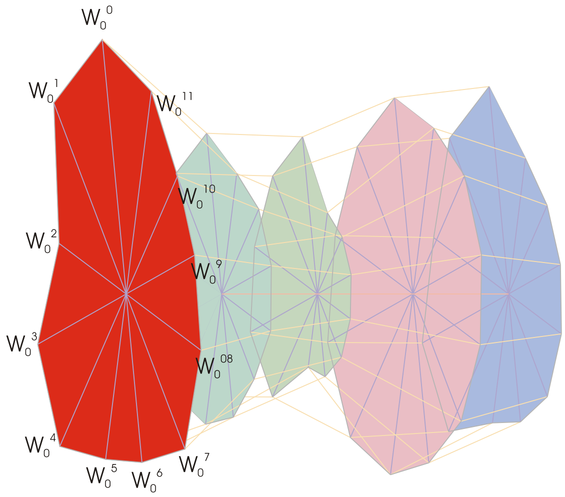


Figure 5.9: SGglyph in Parallel Glyphs visualization.

Topology is "a branch of mathematics concerned with those properties of geometric configurations (as point sets) which are unaltered by elastic deformations (as a stretching or a twisting) that are homeomorphisms" [34].

The transformations discussed are:

- scaling (both in x and y)
- rotations (in x, y, and z, and also as applied to the whole of the visual representation, or to parts of the visual representation separately)
- translations
- colour adjustments

Definition 5.1.1. The *PGVL* topology is the geometric configuration of the morphological units, preserved by a specific set of permissible transformations.

Axiom 5.1.1. The only topological transformations allowed to be applied on the morphological units are:

1. Scaling

a) The scaling in the x direction must be defined on the sets of *PVwords* and *PCwords*, $\mathscr{PVW}$ and $\mathscr{PCW}$. Moreover, if used, the x-scaling must be applied simultaneously on both of these sets, with the same scaling factor. Thus, a uniform scaling of all visible *PVwords* and *PCwords* in x is obtained (See Figure 5.10). Individual x-scaling of a *MDmorpheme* or *PVmorpheme* is not allowed, as that would constitute a mismatch with the corresponding visual representation by breaking the alignment of the points.

Let $S_x\colon \mathscr{PVW} \times \mathbb{R} \to \mathscr{PVW}$,

$S_x(v_1, v_2, ..., v_{NrVisDim}, \alpha) = (v'_1, v'_2, ..., v'_{NrVisDim})$, where

$v_i = (id_{d_{v_i}}, x_{v_i}, y_{v_i}, z_{v_i}, clr) \in \mathscr{PV}$ and $v'_i = (id_{d_{v_i}}, x_{v'_i}, y_{v_i}, z_{v_i}, clr) \in \mathscr{PV}$,

$x_{v'_i} = \alpha * x_{v_i}$, for $i \in \{1, ..., NrVisDim\}$.

Similarly we define scaling for $\mathscr{PCW}$.

Let $S'_x\colon \mathscr{PCW} \times \mathbb{R} \to \mathscr{PCW}$,

$S'_x(w_1, w_2, ..., w_{NrVisDim}, \alpha) = (w'_1, w'_2, ..., w'_{NrVisDim})$, where

$w_i = (id_{t_{w_i}}, id_{d_{w_i}}, x_{w_i}, y_{w_i}, z_{w_i}, clr) \in \mathscr{MD}$ and $w'_i = (id_{t_{w_i}}, id_{d_{w_i}}, x_{w'_i}, y_{w_i}, z_{w_i}, clr) \in \mathscr{MD}$,

with $x_{w'_i} = \alpha * x_{w_i}$, for $i \in \{1, ..., NrVisDim\}$.

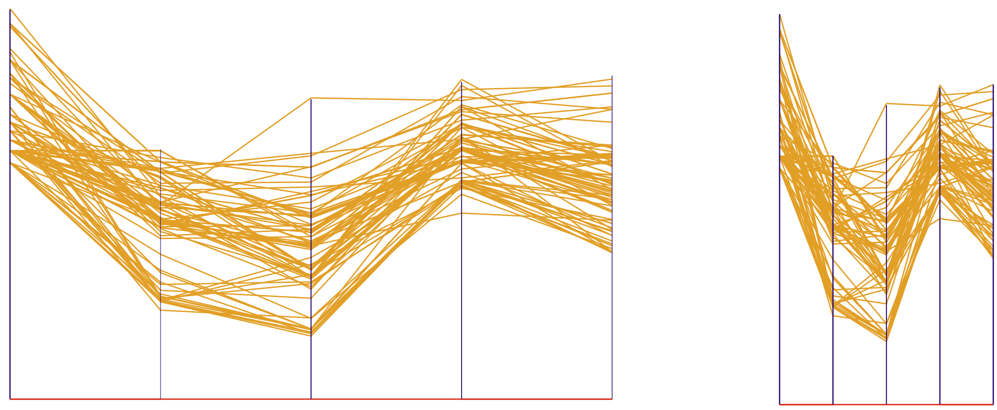
b) The scaling in the y direction, on the other hand, can be defined for individual *MDmorphemes*. The scaling factor is dependent on the *MDmorpheme*. Therefore, the y-scaling can be non-uniform, with some *MDmorphemes* being even left unchanged (See Figure 5.11).

Let $S'_y\colon \mathscr{MD} \times \mathbb{R} \to \mathscr{MD}$,

$S'_y(w, \alpha_w) = w'$, where

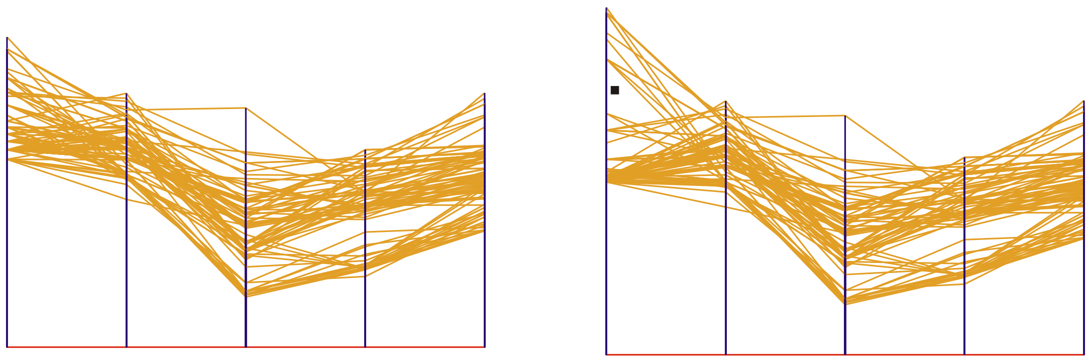
$w = (id_{t_w}, id_{d_w}, x_w, y_w, z_w, clr) \in \mathscr{MD}$ and $w' = (id_{t_w}, id_{d_w}, x_w, y_{w'}, z_w, clr) \in \mathscr{MD}$,

$y_{w'} = \alpha_w * y_w$.



(a) Initial visualization. (b) Uniform x scaling.

Figure 5.10: Scaling on the x axis, applied to Parallel Coordinates or folded Parallel Glyphs.



(a) Original visualization. (b) Scaled visualization.

Figure 5.11: The original visualization is modified by a non-uniform y-axis scaling applied to the first dimension.

Observations about scaling transformations:

Should one of these types of scaling be applied to $\mathscr{PVW}$, $\mathscr{PCW}$ or $\mathscr{MD}$, the other morphological units, *PClines*, *PVlines* and *PCaxes*, must be modified accordingly, as they contain the already scaled morphemes. Therefore S'_x and S'_y are naturally extended to the other sets defined in Chapter 4, such as $\mathscr{PCL}$, $\mathscr{PVL}$, $\mathscr{PCA}$.

2. Rotations

a) The rotation about the y axis is necessary to see the polylines that otherwise would be overlapped, due to the orthographic projection. This is why I will first discuss this rotation, since all the other rotations need a certain degree of y rotation, in order to obtain meaningful images.

The y rotation can be only united. This means that the entire visualization has to be rotated in unison. This is necessary in order to maintain the parallel structure of the glyphs.

Let R_y be the rotation about the y axis, applied to *PVmorphemes*

$R_y\colon \mathscr{PV} \times \mathbb{R} \to \mathscr{PV}$,

$R_y(v, \alpha_y) = v'$, where

$v = (id_{d_v}, x_v, y_v, z_v, clr) \in \mathscr{PV}$ and $v' = (id_{d_v}, x_{v'}, y_{v'}, z_{v'}, clr) \in \mathscr{PV}$,

$x_{v'} = x_v * \cos(\alpha_y) + z_v * \sin(\alpha_y)$,

$y_{v'} = y_v$,

$z_{v'} = -x_v * \sin(\alpha_y) + z_v * \cos(\alpha_y)$.

Similarly, the rotation about the y axis can be defined for all *MDmorphemes*.

$R'_y\colon \mathscr{MD} \times \mathbb{R} \to \mathscr{MD}$,

$R'_y(w, \alpha_y) = w'$, where

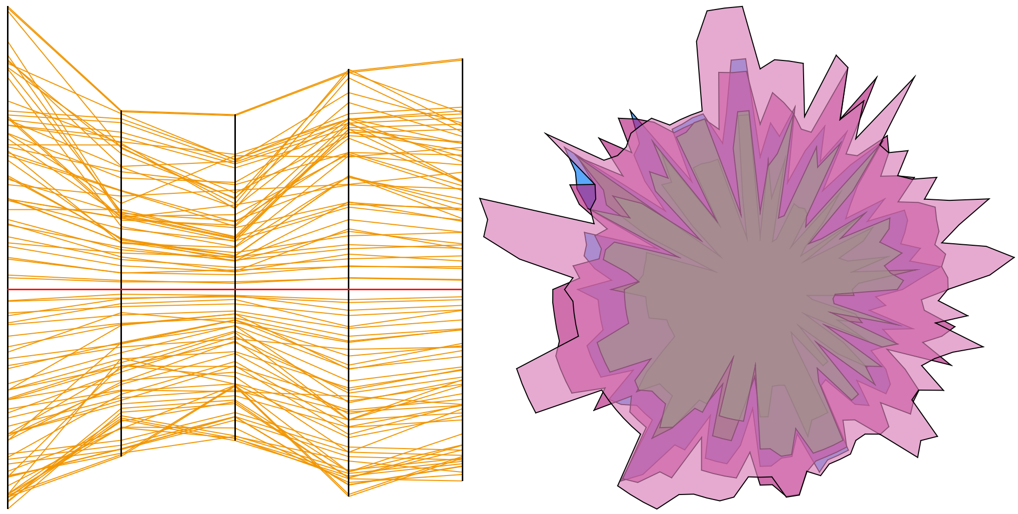
$w = (id_{t_w}, id_{d_w}, x_w, y_w, z_w, clr) \in \mathscr{MD}$ and $w' = (id_{t_w}, id_{d_w}, x_{w'}, y_{w'}, z_{w'}, clr) \in \mathscr{MD}$,

$x_{w'} = x_w * \cos(\alpha_y) + z_w * \sin(\alpha_y)$,

$y_{w'} = y_w$,

$z_{w'} = -x_w * \sin(\alpha_y) + z_w * \cos(\alpha_y)$.

In conclusion, the rotation about the y axis is united, because it uses the same angle α_y for all *PVmorphemes* and *MDmorphemes*. It can facilitate the comparison of Parallel Glyphs, by overlapping them (see Figure 5.12).



(a) No y rotation: most of the items are not visible because the orthographic projection.

(b) Rotation around y with 90°.

Figure 5.12: The united y rotation allows the control of the items overlapping.

b) The rotation about the x axis can be united or not. Moreover, the not-united x rotation can be either uniform or not-uniform.

First, a united rotation is available. In this case, all *PVmorphemes* and *MDmorphemes* are rotated about the pivot line with the same angle (see Figure 5.13).

Let R_x^u be the rotation about the x axis, applied to *PVmorphemes*

$R_x^u\colon \mathscr{PV} \times \mathbb{R} \to \mathscr{PV}$, $R_x^u(v, \alpha_x) = v'$,

where $v = (id_{d_v}, x_v, y_v, z_v, clr) \in \mathscr{PV}$ and $v' = (id_{d_v}, x_{v'}, y_{v'}, z_{v'}, clr) \in \mathscr{PV}$,

$x_{v'} = x_v$,

$y_{v'} = y_v * \cos(\alpha_x) - z_v * \sin(\alpha_x)$,

$z_{v'} = y_v * \sin(\alpha_x) + z_v * \cos(\alpha_x)$.

Let $R^{u\prime}_x$ be the rotation about the x axis, applied to *MDmorphemes*

$R^{u\prime}_x : \mathscr{MD} \times \mathbb{R} \rightarrow \mathscr{MD}$, $R^{u\prime}_x(w, \alpha_x) = w'$,

where $w = (id_{t_w}, id_{d_w}, x_w, y_w, z_w, clr) \in \mathscr{MD}$ and $w' = (id_{t_w}, id_{d_w}, x_{w'}, y_{w'}, z_{w'}, clr) \in \mathscr{MD}$,

$x_{w'} = x_w$,

$y_{w'} = y_w * \cos(\alpha_x) - z_w * \sin(\alpha_x)$,

$z_{w'} = y_w * \sin(\alpha_x) + z_w * \cos(\alpha_x)$.

Second, a not-united rotation can be defined for *PCwords*. This version of rotation about the x axis is essential for the formal description of the construction of Parallel Glyphs. In this case, the angle of rotation is different for each *PCword*, depending on its id_t.

$R_x : \mathscr{PCW} \times \mathbb{R} \rightarrow \mathscr{PCW}$, $R_x(w_0, w_1, ..., w_{NrVisDim}, \alpha_{id_{t_w}}) = (w'_0, w'_1, ..., w'_{NrVisDim})$

where $w_i = (id_{t_w}, id_{d_{w_i}}, x_{w_i}, y_{w_i}, z_{w_i}, clr) \in \mathscr{MD}$, $w'_i = (id_{t_w}, id_{d_{w_i}}, x_{w'_i}, y_{w'_i}, z_{w'_i}, clr) \in \mathscr{MD}$, $i \in \{0, ..., NrVisDim\}$.

$x_{w'_i} = x_{w_i}$,

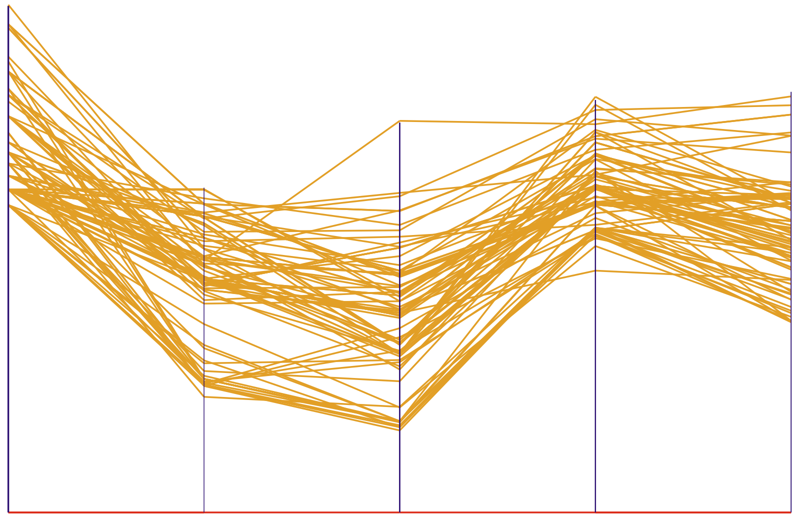
$y_{w'_i} = y_{w_i} * \cos(\alpha_{id_{t_w}}) - z_{w_i} * \sin(\alpha_{id_{t_w}})$,

$z_{w'_i} = y_{w_i} * \sin(\alpha_{id_{t_w}}) + z_{w_i} * \cos(\alpha_{id_{t_w}})$.

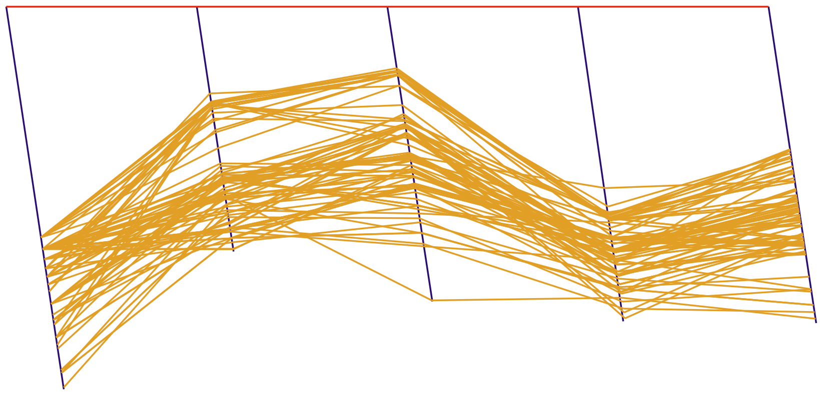
If the angle $\alpha_{id_{t_w}}$ is computed starting from a total angle α_{tot}, the *PCwords* are uniformly distributed relative to the corresponding *PVmorphemes*.

$\alpha_{id_{t_w}} = (\frac{\alpha_{tot}}{NrVisTuples} - 1) * id_{t_w}$, where $\alpha_{tot} \in [0..360]$.

Therefore, the total angle between the first and the last *PCword* is α_{tot}. The angle between any two consecutive *PCwords* is constant, being equal to $\frac{\alpha_{tot}}{NrVisTuples} - 1$. This fact is better



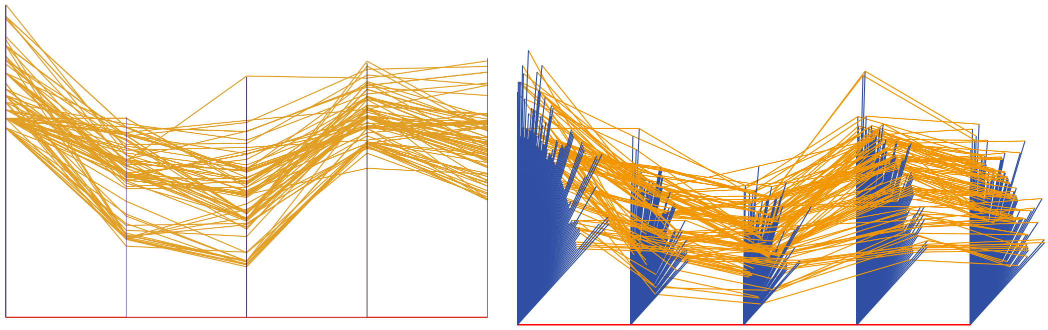
(a) Initial visualization.



(b) United rotation around x.

Figure 5.13: United rotation around the x axis. A slight y rotation is also applied.

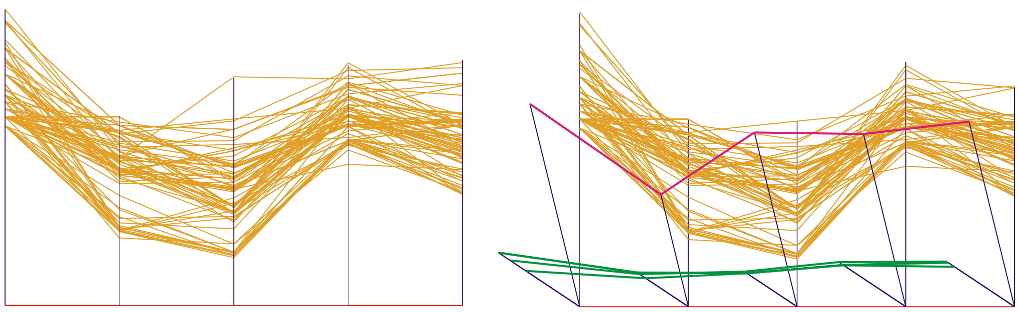
reflected by the angles between *PCaxes*, because R_x can be naturally extended to all the other sets of morphological units (see Figure 5.14).



(a) Initial visualization. (b) Not-united uniform x rotation.

Figure 5.14: Not-united rotation around the x axis, where all angles determined by two consecutive *PCaxes* are equal. A slight y rotation is also applied.

The same not-united rotation can be defined on $\mathscr{PCW}$, without imposing any constrain on the angle $\alpha_{id_{t_w}}$. This is a non-uniform rotation that allows individual *PCwords* to be rotated independently from the other *PCwords*. This transformation is the formal representation of the great flexibility of Parallel Glyphs, supporting the permutation of the polylines and the hinge effect described in Chapter 3 (see Figure 5.15).



(a) Initial visualization. (b) Not-united, non-uniform rotation around x.

Figure 5.15: Not-united rotation around x, where different rotation angles are used for the red *PCline* and for the group of green *PClines*, respectively. The other *PClines* are not rotated around x. A slight y rotation is also applied.

c) The rotation about the z axis is similar to the y rotation, because it can only be united and can be defined for both *PVmorphemes* and *MDmorphemes*. However, it is not always necessary to have a z rotation (see Figure 5.16).

Let R_z be the rotation around the z axis, applied to *PVmorphemes*:

$R_z\colon \mathscr{PV} \times \mathbb{R} \rightarrow \mathscr{PV}$, $R_z(v, \alpha_z) = v'$,

where $v = (id_{d_v}, x_v, y_v, z_v, clr) \in \mathscr{PV}$ and $v' = (id_{d_v}, x_{v'}, y_{v'}, z_{v'}, clr) \in \mathscr{PV}$,

$x_{v'} = x_v * \cos(\alpha_z) - y_v * \sin(\alpha_z)$,

$y_{v'} = x_v * \sin(\alpha_z) + y_v * \cos(\alpha_z)$,

$z_{v'} = z_v$.

Similarly, the rotation about the z axis can be defined for all *MDmorphemes*.

$R'_z\colon \mathscr{MD} \times \mathbb{R} \rightarrow \mathscr{MD}$, $R'_z(w, \alpha_z) = w'$, where

$w = (id_{t_w}, id_{d_w}, x_w, y_w, z_w, clr) \in \mathscr{MD}$ and $w' = (id_{t_w}, id_{d_w}, x_{w'}, y_{w'}, z_{w'}, clr) \in \mathscr{MD}$,

$x_{w'} = x_w * \cos(\alpha_z) - y_w * \sin(\alpha_z)$,

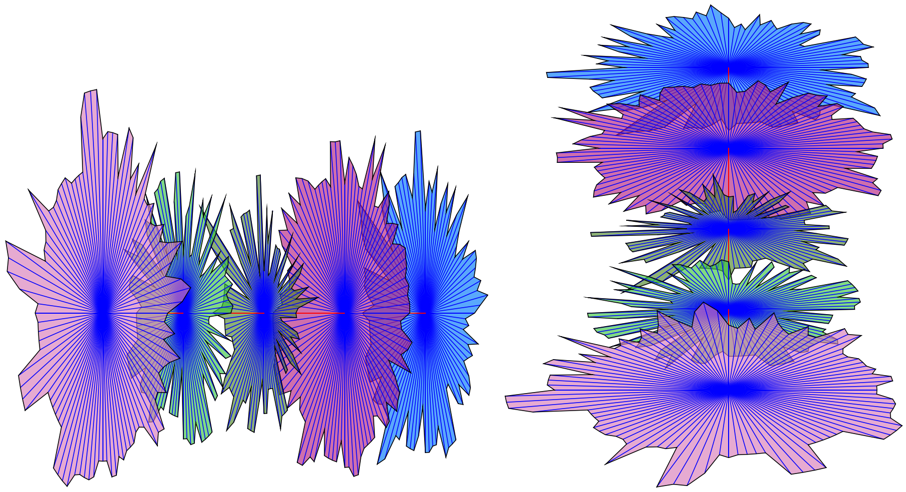
$y_{w'} = x_w * \sin(\alpha_z) + y_w * \cos(\alpha_z)$,

$z_{w'} = z_w$.

Because these transformations of *PVmorphemes* and *MDmorphemes* change the position of their corresponding points, the rest of the morphological units imported by *PGVL* from both *PCVL* and *SGVL* must be transformed accordingly.

3. Translation

Translations in y and z directions need to be united, in order to maintain the structure of the Parallel Glyphs. An x united translation is also available. These united transformations are useful for scrolling, to set the focus on a particular part of the visualization. However,



(a) No z rotation. (b) Rotation around z with 90°.

Figure 5.16: United rotation around the z axis, combined with a slight y rotation.

in graphics the same visual effect can be obtained by translating the camera around a fix 3D model.

The only not-united translation allowed is in the x direction, along the pivot line. If used, this transformation must be applied to all *MDmorphemes* having the same id_d and to the corresponding *PVmorpheme*, in order to maintain the alignment of their corresponding points along the dimension axis. Therefore, translation can only be defined on the sets of *SGwords*, *SGfans*, *SGborders*, and *SGglyphs*.

$T_x\colon \mathscr{PV} \times \mathbb{R} \to \mathscr{PV}$, $T_x(v, \alpha_x) = v'$, where

$v = (id_{d_v}, x_v, y_v, z_v, clr) \in \mathscr{PV}$, and $v' = (id_{d_v}, x_{v'}, y_v, z_v, clr) \in \mathscr{PV}$,

$x_{v'} = x_v + \alpha_x$.

Similarly, the translation of the corresponding *SGwords* is defined.

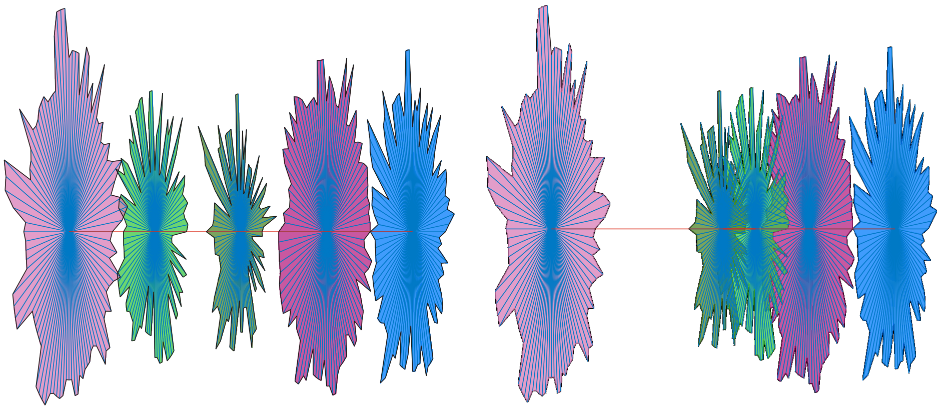
$T'_x\colon \mathscr{SGW} \times \mathbb{R} \to \mathscr{SGW}$,

$T(g, \alpha_x) = g'$, where $g, g' \in \mathscr{SGW}$ and $\forall$ *MDmorpheme* $w_i \in g$,

$w_i = (id_d, x_{w_i}, y_{w_i}, z_{w_i}, clr)$, the corresponding *MDmorpheme* $w'_i \in g'$,

$w'_i = (id_d, x_{w'_i}, y_{w_i}, z_{w_i}, clr)$ has the property:

$x_{w'_i} = x_{w_i} + \alpha_x$, for $i \in \{1, ..., NrVisTuples\}$.



(a) Initial visualization. (b) The second glyph is translated between the third and the fourth.

Figure 5.17: Permutation of the Parallel Glyphs using not-united non-uniform translation along the x axis.

This transformation, just described, is essential to support the permutations p used in Chapter 4 to generalize each morphological unit. Note that the translation factor α_x can be different for each data dimension (see Figure 5.17). The same observation from rotations can be made here: the other morphological units, containing translated *MDmorphemes* must be transformed accordingly. This is done by extending the application T_x to the corresponding sets of morphological units.

4. Colour/Transparency

Colour, as an *RGBA* value, is an attribute of each *MDmorpheme* and, consequently, of each morphological unit. Therefore, it is important to define a transformation that provides formal support in *PGVL* for interactions provided by the visual representation of Parallel Glyphs. The change of colour and/or transparency can be included in the category of topological transformations because it acts like an identity function for the position or shape of any morphological unit.

In Chapter 4 the set of all colours *RGBA* has been defined as $RGBA = [0,1] \times [0,1] \times [0,1] \times [0,1]$. A colour $clr \in RGBA$ has the form $clr = (r, g, b, a)$, where $r, g, b, a \in [0,1]$ are the red, green, blue and alpha components.

The topological transformation that allows the change of colour and/or transparency of a *MDmorpheme* is $C\colon \mathscr{MD} \times RGBA \to \mathscr{MD},\ C(md, clr') = md'$, where
$md = (id_d, id_t, x, y, z, clr)$ and $md' = (id_d, id_t, x, y, z, clr')$.
Note that if $clr = (r, g, b, a)$ and $clr' = (r, g, b, a')$ the transformation C modifies only the transparency of the *MDmorpheme*.

Similar transformations can be defined for all the other morphological units. For *SGwords*, where the associated colour is an array of colours, C needs to be defined on $\mathscr{SG} \times RGBA^{NrVisTuples}$.

5.2 Syntax for Parallel Glyphs

Syntax is "the study of the rules, or "patterned relations" that govern the way the words in a sentence come together" [5]. It studies groups of words as parts of sentences and the relationships among them, as opposed to morphology, which studies words as individual units.

This section will detail the syntax of the Parallel Glyphs Visual Language *PGVL*, establishing rules for using the morphological units presented in section 5.1 together. Moreover, the role of Parallel Glyphs in addressing the problem of overlapping polylines in Parallel Coordinates is formally illustrated here by showing how a *PGVL* sentence can become grammatically correct by applying a combination of topological transformations.

Definition 5.2.1. A sentence in *PGVL* is a combination of *PGVL* morphological units.

In linguistics, a sentence requires two main units: a subject and predicate. However, a thought can be conveyed through one word only, if that word is an imperative predicate. For example: "Stop!". In the context of this linguistic formalism, a complete thought corresponds for now to either an entire row or a column of the data table. Therefore, a sentence in *PGVL* must obey the following rule:

Rule PG.1. A sentence must contain at least one *PCword* or one *SGword*.

Unlike natural languages, where some words can be repeated within the same sentence, in Parallel Glyphs each item is visualized only once. This particularity must also be reflected in the *PGVL* syntax:

Rule PG.2. A morphological unit can be used at most once in a sentence. However, an *MDmorpheme* or *PVmorpheme* can belong to different morphological units. Therefore, they can be used more than once in a sentence.

Using the topological transformations introduced in section 5.1.3 it is now possible to show how Parallel Glyphs address the occlusion problem pointed out by the syntax of Parallel Coordinates and Star Glyphs (Chapter 4).

Theorem 5.2.1. Sentences declared non-grammatically correct in *PCVL* or in *SGVL* due to their ambiguous occlusions can be corrected in *PGVL* via combinations of the topological transformations defined above.

Proof. Proof by induction.

Step 1 $k = 1$: If a sentence has one ambiguous occlusion, there exists one combination of topological transformations that, when applied to that sentence, eliminates the occlusion. Such an ambiguous occlusion is common to a 2D Parallel Coordinates visualization (See Figure 5.18). Because this is the starting position of Parallel Glyphs, this problem must be addressed by the available topological transformations. Indeed, there exists a combination

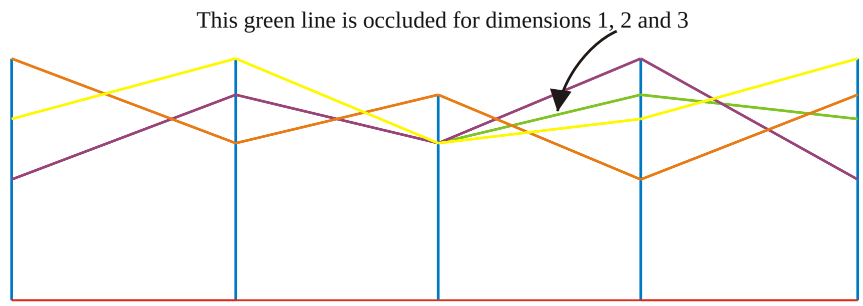
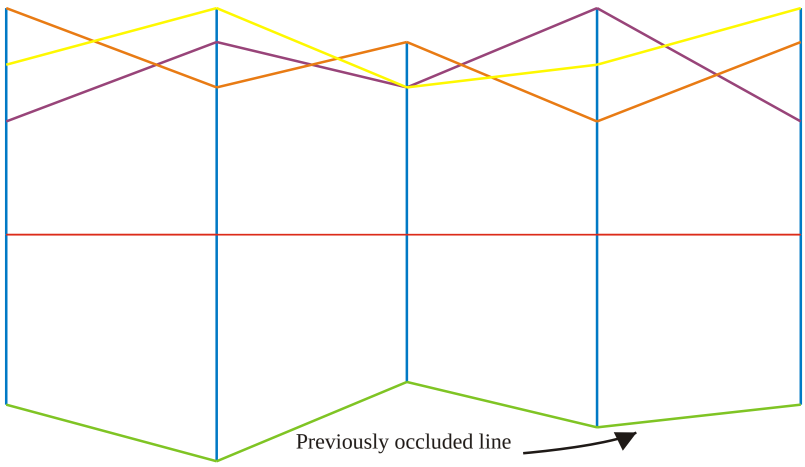


Figure 5.18: Ambiguous occlusion in Parallel Glyphs visualization.

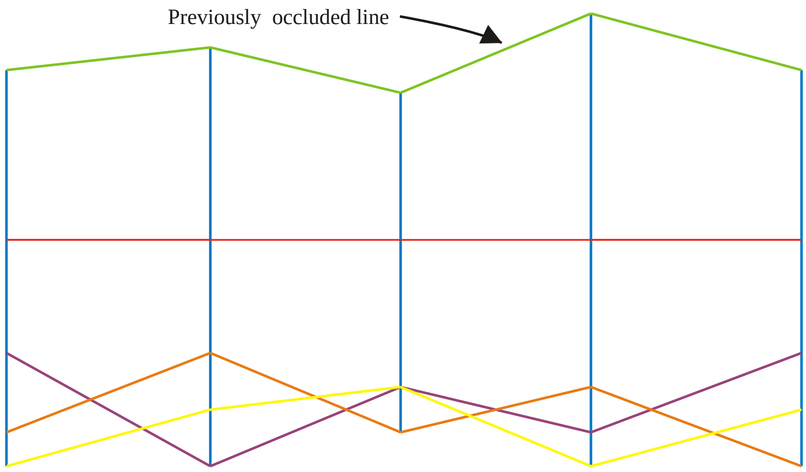
of rotations $\mathscr{C}_1 = R_x \circ R_y \circ R_z$ that can transform a Parallel Glyphs sentence so that the ambiguous occlusion is eliminated. For example, such a combination of rotations with $\alpha_x = 180°$, $\alpha_y = 0°$, $\alpha_z = 0°$ can be applied either to the overlapped line or to the visible lines (see Figure 5.19).

Step 2 $k = n$: Assume that if a sentence has n ambiguous occlusions, there is a combination of topological transformations $\mathscr{C}_n$ that eliminates these occlusions.

Step 3 $k = n+1$: Using the assumption from Step 2, show that if a sentence has $n+1$ ambiguous occlusions there exists a combination of topological transformations $\mathscr{C}_{n+1}$ that eliminates these occlusions (See Figure 5.20).



(a) Solution 1: the overlapped line is rotated with 180° around x.



(b) Solution 2: the visible lines are rotated 180° around x.

Figure 5.19: Example of solutions for the ambiguous occlusion presented in Figure 5.18.

Indeed, the first n occlusions can be solved by the combination $\mathscr{C}_n$, according to assumption made in Step 2. If no other occlusion is induced by applying $\mathscr{C}_n$, it results that there is only one occlusion remaining.

According to Step 1, there exists a combination $\mathscr{C}_1$ that solves this occlusion. Therefore, there exists a combination of topological transformations $\mathscr{C}_{n+1} = \mathscr{C}_1 \circ \mathscr{C}_n$ that elim-

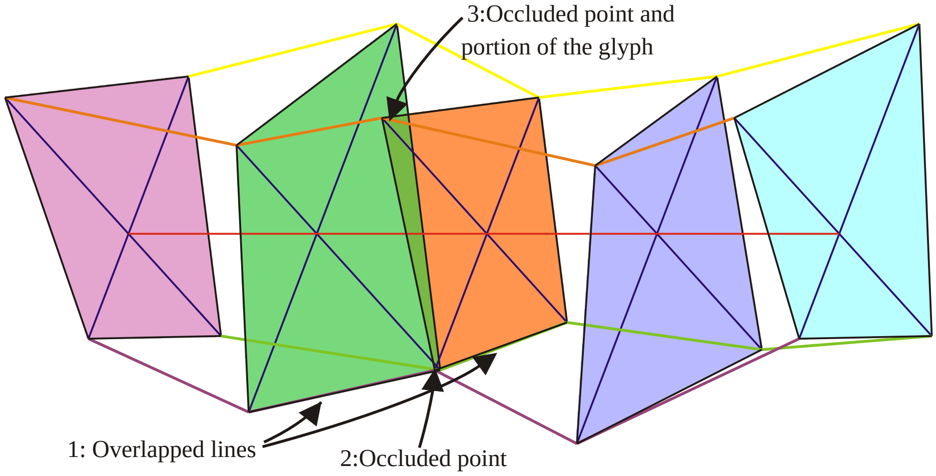
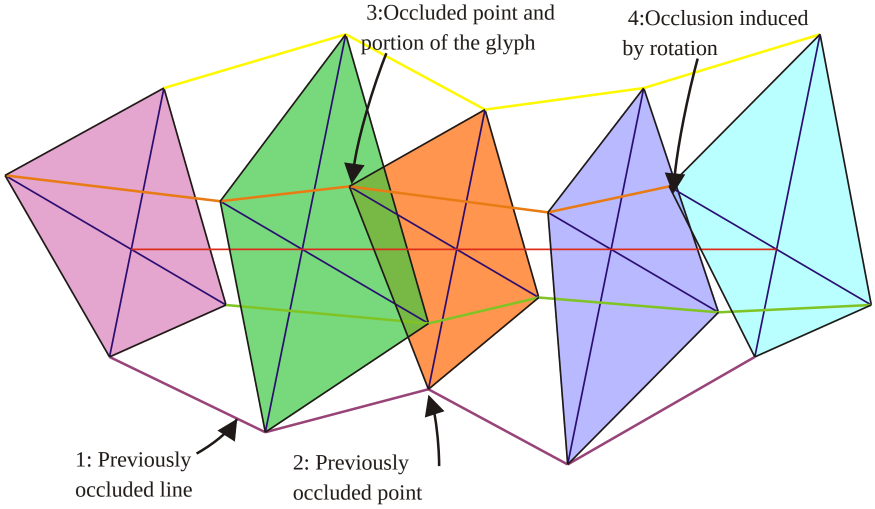


Figure 5.20: Ambiguous occlusions in Parallel Glyphs visualization.

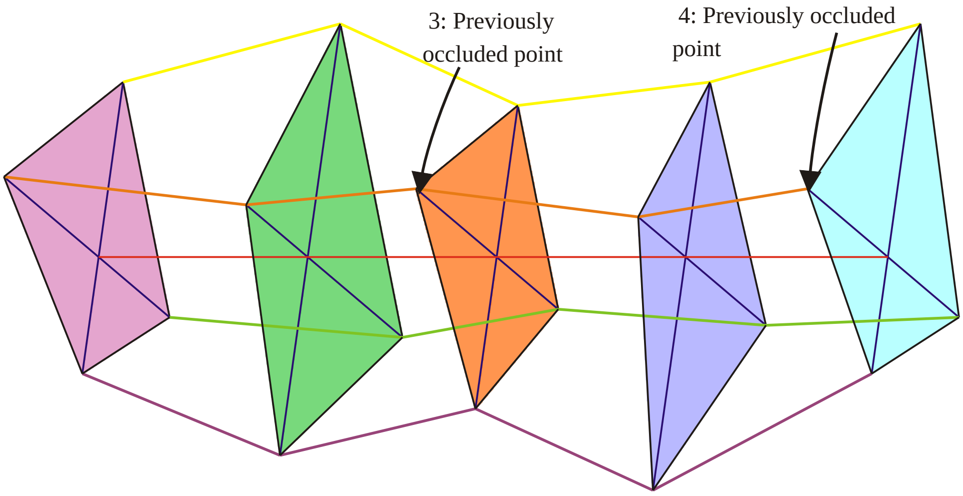
inates all $n+1$ ambiguous occlusion of a *PGVL* sentence. If there are other p occlusions induced by applying $\mathscr{C}_n$, they can be solved by successively applying combinations $\mathscr{C}_p$ until there is only one occlusion left. The last one is eliminated by $\mathscr{C}_1$. Therefore, all $n+1$ occlusions are eliminated by a combination $\mathscr{C}_{n+1} = \mathscr{C}_1 \circ \mathscr{C}_p \circ \mathscr{C}_n$ (See Figure 5.21).

Step 4 *Conclusion*: Any *PGVL* sentence can be made grammatically correct by eliminating all ambiguous occlusions, via a combination of topological transformations. □

Observation: If the number of dimensions or tuples is large, more complex combinations of topological transformations are required. Therefore, a combination that would make all units occlusion-free at the same time could be difficult to find. However, a less restrictive approach that would require any unit (not all) be visible at a time is more accessible. One solution for reducing the complexity of combinations necessary to eliminate all occlusions is to use transparency. Thus, the definition of occlusion would be more permissive. Moreover, a controlled overlapping of transparent Parallel Glyphs can be desirable, for



(a) Step 1: Rotation around x to address ambiguous occlusions presented in Figure 5.20.



(b) Step 2: Additional rotation around y eliminates all remaining or induced occlusions.

Figure 5.21: Combination of transformations eliminates the ambiguous occlusions presented in Figure 5.20.

comparison purposes. Another solution for an easier elimination of occlusions is the use of clustering algorithms that would actually reduce the number of visualized elements.

The upcoming section will explain clustering and how it can be formally described using the alphabet $\mathscr{A}$.

5.3 Clustering

The question raised by this section is whether the formalism of *FMDVL* can provide support for describing interactions such as clustering, that modify the visualization of the initial dataset. The answer is yes, but not directly. The explanation of this answer will be elaborated next.

Clustering is "the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often proximity according to some defined distance measure" [5]. Furthermore, the centroid of each cluster is the only element visualized. Each clustering algorithm defines the way in which the clusters are created based on the user-defined measure of similarity and how the centroids are computed.

One simple such algorithm is K-Means Clustering, composed of the following steps:

1. Initialize (usually randomly) the K points representing the centroids $c_1, ..., c_k$.

2. Distribute each item into the cluster that has the closest centroid.

3. When all items are distributed, recalculate the centroids.

4. Repeat Steps 2 and 3 until the centroids are fixed.

The results depend on the number of clusters, the initial values and on the metric d used to measure the distance between an item and a centroid. An item w_i, $i \in \{1,$ ..., $NrVisTuples\}$ belongs to cluster j, $j \in \{1, ..., k\}$ if $d(w_i, c_j)$ is the minimum of all k distances. One possible such metric d is the Euclidean distance.

An immediate observation is that $c_1,...,c_k \notin \mathscr{A}$. This fact has two causes: first, c_j are 3D points, so they have a format different of the *MDmorphemes*. Second, the values of c_j are computed as means of the values from the data table. Therefore, it is not likely that c_j belong to the initial dataset. However, each of these problems can be eliminated, making it possible for c_j to belong to $\mathscr{A}$. *FMDVL* will then be able to formally describe clustering.

The format can be solved by creating *K-morphemes* with the same structure as the *MDmorphemes* $(id_d, id_t, x, y, z, clr)$. Assume that $c^i_j = (x^i_j, y^i_j, z^i_j)$ is the centroid of the j^{th} cluster computed for *MDmorphemes* corresponding to the i^{th} dimension (that is, having $id_d = i$). Then a *K-morpheme* $K^i_j = (i, NrVisTuples + j, x^i_j, y^i_j, z^i_j, clr_j)$. Relative to the colours of the cluster's components, clr_j can be either a new, independent colour, or an average of those colours. The important thing is that the same colour clr_j is assigned to all j^{th} centroids c^i_j, $\forall i \in \{1,...,NrVisDim\}$.

Although the *K-morphemes* defined above have the same structure as the *MDmorphemes*, the problem of x^i_j, y^i_j, z^i_j not being elements of the initial dataset T cannot be avoided. Therefore, the only solution is to allow the extension of the alphabet $\mathscr{A}$. Actually, this is an extension of T, which now has the dimensions $(NrVisTuples + k) * NrVisDim$, being increased with the values of all centroids c^i_j, where $i \in \{1,...,NrVisDim\}$, $j \in \{1,...,K\}$. The extended alphabet $\bar{\mathscr{A}} = \mathscr{A} \cup \mathscr{K}$, where $\mathscr{K}$ is the set of all *K-morphemes*. All the definitions of morphological units and syntactical rules can now be naturally extended to $\bar{\mathscr{A}}$.

In conclusion, clustering can be formally described by *FMDVL* via the extended alphabet $\bar{\mathscr{A}}$. Although this can seem a stretch of the initial definition of *FMDVL*, one should not forget that for natural languages it is common practise to import foreign morphemes in order to accommodate neologisms. Analogously, *FMDVL* needs to import the *K-morphemes* to be able to represent "neologisms" visualized by centroids. Because the change has been made only at the alphabet level, independent of the visual language, it follows that clustering can be formally represented for each individual visualization (Parallel Coordinates, Star Glyphs and Parallel Glyphs).

5.4 Summary

This chapter has extended the Family of Multi-Dimensional Visual Languages *FMDVL* with a formal representation of Parallel Glyphs, the visual representation of multi-dimensional data introduced in Chapter 3. In addition, the example of a formal definition of clustering in Parallel Coordinates has illustrated the capability of a visual language to describe interactions specific to the corresponding visual representation. Therefore, this chapter has emphasized the flexibility of *FMDVL* by showing two different areas of applicability. First, this linguistic formalism can provide a formal description of yet another visual representation of multi-dimensional data, Parallel Glyphs. Second, it can describe interactions available for these visual representations, such as clustering. This chapter concludes by mentioning some of the strengths and limitations of *FMDVL*.

Strengths The linguistic formalism based on the alphabet $\mathscr{A}$ of *FMDVL* has the capability to describe visual representations of multi-dimensional data stored in a data table. The morphemes composing the alphabet $\mathscr{A}$ are uniquely identified by the position of the corresponding value in the table. However, all the other attributes of a morpheme are customizable in order to accommodate each morphology, describing a particular visual representation. The structure of the morphemes provides the alphabet with the flexibility to describe both 2D and 3D techniques.

As shown in Chapter 4, formal representations of different visual representations that have in common only the dataset, can use the same alphabet to develop a visual language. Furthermore, as shown in this chapter, section 5.1, the topological transformations allow for different morphologies to be combined into a new one. This can emphasize connections between the original visual representations that might have been overlooked otherwise.

Another positive aspect of *FMDVL* is that it can describe operations or interactions applied to the visual representations corresponding to each visual language. This is exemplified in section 5.3 of this chapter by a formal description of clustering. The clustering technique was introduced in Chapter 2, Section 2.3.1, where it was described how it can be applied to Parallel Coordinates and Star Glyphs.

Limitations The definition of morphemes implies that the data is stored in a data table and uses the identification number of columns and rows as *id* for morphemes. This fact restricts the applicability of this linguistic formalism based on $\mathscr{A}$ to describe only visual representations supporting a tabular format. Therefore, visual representations such as trees, with a hierarchical representation cannot be described using $\mathscr{A}$. However, by using the example of alphabet $\mathscr{A}$ for multi-dimensional tabular data, another alphabet, suitable for hierarchies, could be defined.

Another problem with the Family of Multi-Dimensional Visual Languages is the confusion created by the names: the Family of Multi-Dimensional Visual Languages (*FMDVL*) and visual programming languages, which are pretty similar. This causes a tendency to look for applicabilities of *FMDVL* in supporting visual programing languages, although there are absolutely no inter-relations. The cause of this is that *FMDVL* has been named according to Horn's definition [33] of Visual Language.

There is one exception of interaction available for Parallel Glyphs, described in Chapter 3, that cannot be directly described by this formalism. This is the case of EdgeLens, which transforms one segment of a polyline into a B*é*zier curve. The reason for this limitation is that the control points of the B*é*zier curve are computed by using external variables, namely the centre of the lens, and its radius and magnification. Therefore, these values being independent of the data, they do not have a corresponding morphological unit. A simple solution would be to consider these variables as parameters $\alpha_x, \alpha_y \in \mathscr{R}$, and $\beta, \gamma \in \mathscr{R}_+$ and compute the control points based on them. However, further investigations are necessary to rigorously define such a topological transformation.

In conclusion, this linguistic formalism opens a new direction for specifying information visualization techniques. It sets an example of formal representations of multi-dimensional data visualizations that support a tabular format. This example can be extended or adapted for other visualization techniques. This is important, as the ultimate goal of such a formal representation is to assess existing visualization methods and to help the development of other new, improved techniques.

Chapter 6

Conclusion and Future Work

This chapter wraps up this thesis by summarizing the contributions made by the research presented here. The first section restates the problems related to Parallel Coordinates and Star Glyphs that have been identified in Chapter 1. The next section shows how each goal from Chapter 1 has been achieved. The chapter concludes by identifying several directions for future work in the area of multi-dimensional data visualization and the associated linguistic formalism.

6.1 Problem statement

The following limitations of Parallel Coordinates and Star Glyphs were identified in Chapter 1.

Problem 1. A multi-dimensional dataset can be visualized using either Parallel Coordinates or Star Glyphs, but it is not sufficiently explored whether these visual representations could benefit from each other.

- **a.** Parallel Coordinates are hard to read, while Star Glyphs are, comparatively, easily understood.
- **b.** Vice-versa, Star Glyphs have limitations that are well addressed by Parallel Coordinates. For example Star Glyphs do not scale well to large datasets, do not

offer an overview of the entire dataset and do not provide support for comparison of non-adjacent data items.

Whether or not the two visual representations can support each other and preserve their individual benefits while at the same time reducing their individual limitations, is a problem to investigate.

Problem 2. There is no formal method to specify and assess visual representations of multi-dimensional data. With the proliferation of access to digital media, it is becoming increasingly common for people to present information visually. This has led to a myriad of new types of visual representations that frequently come into existence without an associated formalism. It is often difficult to retroactively fit a formalism to an existing visual representation. To the best of my knowledge, existing visual representations of multi-dimensional data have no formal foundation that supports their theoretical investigation and assessment.

Problem 3. The benefits of such formalism are not assessed. Once a formal method of describing visual representation of multi-dimensional data is defined, a series of questions rise. Will this formalism describe Parallel Coordinates and Star Glyphs? Is it applicable to other visual representations of multi-dimensional data? To what extent can it be generalized?

6.2 Thesis goals

This thesis has addressed each of the above mentioned problems, with the following goals in mind:

Goal 1. Enhancing visual representation of multi-dimensional data, by addressing both aspects of the problem:

a. Provide Parallel Coordinates with qualities that Star Glyphs have.

b. Provide Star Glyphs with qualities that Parallel Coordinates have.

(Addresses problem 1)

To achieve this goal, Chapter 3 introduced Parallel Glyphs, which are a three-dimensional extension of Parallel Coordinates integrated with Star Glyphs. This work demonstrates that Parallel Coordinates and Star Glyphs are not only closely related and in the same family of visualizations but that they are, in fact, orthogonal to each other. The traditional view of Parallel Coordinates—objects over dimensions—leads to one of two versions of Star Glyphs—one glyph representing all values for one dimension. The other version of Star Glyphs—one glyph representing the values of one data object—can be derived from a second version of Parallel Coordinates in which dimensions are plotted over objects.

The new 3D visualization has been enriched by applying colour scales as textures to the 3D Star Glyphs. This has been done in order to support the task of comparing spatially separated glyphs, thus addressing a limitation specific to Star Glyphs. Both uniform and stepped colour scales have been discussed, and their merits compared. It has been shown how these colour scales can lead to colour schemes for the data items' polylines that identify trends and support brush-selection. To further support visualization tasks, various interaction possibilities have been integrated. In addition to well-known techniques for 2D Parallel Coordinates, the unfolding of 2D Parallel Coordinates into the new 3D representation, and 3D transformations such as rotation around coordinate axes a ring ruler interaction have been introduced. This enables determining even small differences between Star Glyphs. Furthermore, Parallel Glyphs provide a newly possible data item rearrangement. This enables users to directly compare selected data items without loosing the context of the rest of the data set. Finally, additional enhancements to interactions that had previously been applied to the traditional 2D Parallel Coordinates visualization are included, such as lens techniques for focus + context data exploration.

All these features of Parallel Glyphs support the idea that, by integrating Parallel Coordinates and Star Glyphs, the limitations that each visual representation has individually can be reduced, while the benefits are at least maintained, if not enhanced.

Goal 2. Defining a formalism that provides a notation capable of describing visual representations of multi- dimensional data *(Addresses problem 2)*. Chapter 4 has presented a formal approach to description of visual representations using an analogy to natural languages. An alphabet $\mathscr{A}$ of a Family of Multi-dimensional Visual Languages *FMDVL* has been defined. $\mathscr{A}$ consists of two types of ordered letters that can be used as the basis for the development of several languages. Two examples, Parallel Coordinates and Star Glyphs, illustrate the way formal descriptions of a family of visual representations can be based on this alphabet. The morphology and the syntax for these two visual representation examples of multi-dimensional data, are detailed.

This linguistic formalism for visual representations extends the influence of Chomsky grammars from visual programming languages to information visualization techniques. The approach proposed here provides a theoretical foundation for description of visual representations, which can be further investigated for other techniques than those detailed in Chapter 4.

Goal 3. Exploring the versatility and applicability of the formalism previously defined in Goal 2 *(Addresses problem 3).* In Chapter 5 a new visual language *PGVL* has been built, starting from the formalisms that separately describe Parallel Coordinates and Star Glyphs. The syntax of the unified visual language has been constructed by applying topological transformations on the previously defined morphological units. This new visual language belongs to *FMDVL* and constitutes the formal foundation of Parallel Glyphs. In addition, *PGVL* provides a formal description of how the Parallel Glyphs visualization actually does address the Parallel Coordinates problem of occlusion.

Another direction of extending the applicability of *FMDVL* is to formally describe the interactions applied on the visualization technique corresponding to each visual language. To illustrate this capability of *FMDVL* and of the alphabet $\mathscr{A}$ the formal description of clustering has been detailed.

6.3 Future Work

The most important contribution of any research is to bring up ideas that open new horizons. This fact also applies to this thesis, as several avenues for further investigations can be identified. For the applied side, regarding Parallel Glyphs, the research can be extended in the following directions.

A user study could be designed and run, to assess how Parallel Glyphs support the user in solving various tasks. The elimination of the individual limitations of Parallel

Coordinates and Star Glyphs has been the goal of adding all the features to Parallel Glyphs. However, actually measuring and evaluating the differences between all three methods could be of interest and could bring up new solutions for improving Parallel Glyphs.

As mentioned in Chapter 3, the size of the data set visualized by Parallel Glyphs has to be small to moderate. To improve the scalability of Parallel Glyphs to large data sets it would be interesting to integrate clustering algorithms, based on a user defined measure of similarity. The glyphs would then be unfolded only after the clustering of the data items or the dimensions. Therefore, the advantage of Parallel Coordinates of handling thousands of data items and tens of dimensions would be extended to Parallel Glyphs.

So far the polylines and the dimension axes can only be reordered manually, via a drag-and-drop mechanism. An interesting direction would be to apply ideas about algorithmic re-ordering of data items and dimensions [90], extending this research toward designing algorithms for automatic reordering of data items and dimensions in Parallel Glyphs.

Beside visual cues such as fog, that is already implemented, it is possible to add shading and other 3D rendering techniques to improve the visualization technique. For now, each polyline is monochrome from one end to the other. This approach to using colour can be extended by allowing each segment or even pixel of a polyline to have a different colour scheme. If the change is in the alpha channel, then only the transparency varies. This would be useful for reducing the visual clutter, because objects in the back-end could be made transparent, or even invisible. Moreover, it could be worth exploring 3D shape analysis methods, to extract more information from the data set.

For the theoretical side of this thesis, the research can be extended in the following directions:

The Family of Multi-Dimensional Visual Languages *FMDVL* can be applied to other visual representations of multi-dimensional data, beside Parallel Coordinates, Star Glyphs and Parallel Glyphs. In addition, an analogous visual language might be defined for hierarchical data. There are enough distinctions between hierarchical and tabular multi-dimensional data that while some of the basic ideas in the *FMDVL* might transfer, further research would be required to discover to what extent it would be applicable, or whether or not the same idea could be used to develop another alphabet.

At times, visual representations of multi-dimensional data employ interactions or transformations that are not always linear. Therefore, it is useful to investigate how non-linear transformations, such as EdgeLens distortions, using B*é*zier curves, can be described by *FMDVL*.

Further investigations can be made to see whether or not the structure of a *PCword* can become more flexible. A direction would be to define a *PCword* as a function of id_t, thus taking id_t outside the structure. In addition, clr can also be defined as a function of id_t or *PCword*. This would be especially useful if Parallel Glyphs support a variable colour scheme along each polyline, as mentioned above.

To complete the analysis of the *FMDVL* grammar would be necessary to develop the associated semantics. This important part of the grammar is beyond the scope of the thesis.

6.4 Conclusion

The research presented in this thesis has two major contributions. The first one is Parallel Glyphs, a new 3D visual representation of multi-dimensional data that integrates Parallel Coordinates and Star Glyphs. The visualization is enhanced by applying colour scales to polylines or as textures on the Star Glyphs. Moreover, various interactions available also support data exploration and visual analysis.

The second contribution is the definition of a linguistic formalism that has the capability to describe multi-dimensional data visualizations. A family of multi-dimensional visual language *FMDVL* has been developed, starting from a common alphabet $\mathscr{A}$. To illustrate the potential of *FMDVL*, the morphology and syntax of three visual languages have been built. They are *PCVL*, *SGVL* and *PGVL*, corresponding to Parallel Coordinates, Star Glyphs and Parallel Glyphs, respectively.

Appendix A. Co-Author Permission

UNIVERSITY OF
CALGARY

March 10, 2006

University of Calgary
2500 University Drive NW
Calgary, Alberta
T2N 1N4

I, Tobias Isenberg, give Elena Fanea permission to use co-authored work from our paper, "An Interactive 3D Integration of Parallel Coordinates and Star Glyphs" for Chapters 3, 5, and 6 of her thesis and to have this work microfilmed.

Sincerely,

Tobias Isenberg

2500 University Drive N.W., Calgary, Alberta, Canada T2N 1N4 • www.ucalgary.ca

UNIVERSITY OF
CALGARY

March 10, 2006

University of Calgary
2500 University Drive NW
Calgary, Alberta
T2N 1N4

I, Sheelagh Carpendale, give Elena Fanea permission to use co-authored work from our paper, “An Interactive 3D Integration of Parallel Coordinates and Star Glyphs” for Chapters 3, 5, and 6 of her thesis and to have this work microfilmed.

Sincerely,

Sheelagh Carpendale

Bibliography

[1] Chernoff Face - MathWorld. http://mathworld.wolfram.com/ChernoffFace.html. visited March 11, 2006.

[2] Merriam Webster Online. http://www.meriamwebster.com/. visited January 8, 2006.

[3] p53 - Biocrawler. http://www.biocrawler.com/encyclopedia/P53. visited January 17, 2006.

[4] Protein Data Bank. http://www.rcsb.org. visited January 17, 2006.

[5] Wikipedia, the Free Encyclopedia. http://www.wikipedia.org/. visited January 8, 2006.

[6] Keith Andrews and Helmut Heidegger. Information Slices: Visualising and Exploring Large Hierarchies Using Cascading, Semi-Circular Discs. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'98) – Late Breaking Hot Topic Paper*, pages 9–12, 1998.

[7] Almir Olivette Artero, Maria Cristina Ferreira de Oliveira, and Haim Levkowitz. Uncovering Clusters in Crowded Parallel Coordinates Visualizations. In *Proceedings of the IEEE Symposium on Information Visualization 2004 (InfoVis 2004)*, pages 81–88, Los Alamitos, CA, 2004. IEEE Computer Society Press.

[8] Eric H Baehrecke, Niem Dang, Ketan Babaria, and Ben Shneiderman. Visualization and Analysis of Microarray and Gene Ontology Data with Treemaps. *BMC Bioinformatics*, 5(84), 2004.

[9] Todd Barlow and Padraic Neville. A Comparison of 2-d Visualizations of Hierarchies. In *Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*, page 131, 2001.

[10] Jaques Bertin. *Semiology of Graphics*. The University of Wisconsin Press, Madison, Wisconsin, 1983.

[11] Irving Biederman. Recognition-by-Components: A Theory of Human Image Understanding. *Psychological Review*, 2:115–147, 1994.

[12] Mark Bruls, Kees Huizing, and Jarke van Wijk. Squarified Treemaps. In *Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization (TCVG 2000)*, pages 33–42, Los Alamitos, CA, USA, 2000. IEEE Press.

[13] Stuart K. Card, Jock D. MacKinlay, and Ben Shneiderman. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufman, San Fransisco, USA, 1999.

[14] Matthew Chalmers. A Linear Iteration Time Layout Algorithm for Visualising High-dimensional Data. In *Proceedings of IEEE Visualization 1996 (VIS 1996)*, pages 127–132, Los Alamitos, CA, 1996. IEEE Computer Society Press.

[15] Hong Chen. Compound Brushing. In *Proceedings of the IEEE Symposium on Information Visualization 2003 (InfoVis 2003)*, pages 181–188, Los Alamitos, CA, 2003. IEEE Computer Society Press.

[16] Herman Chernoff. Using Faces to Represent Points in k-dimensional Space. *Journal of the American Statistical Association*, 68:361–368, 1973.

[17] Ed H. Chi, James Pitkow, Jock Mackinlay, Peter Pirolli, Rich Gossweiler, and Stuart K. Card. Visualizing the Evolution of Web Ecologies. In *Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems*, pages 400–407, 644–645, 1998.

[18] Eleanor Boyle Chlan and Penny Rheingans. Multivariate Glyphs for Multi-Object Clusters. In *Proceedings of the IEEE Symposium on Information Visualization 2005(INFOVIS 2005)*, pages 141–148, Los Alamitos, CA, 2005. IEEE Computer Society Press.

[19] Noam Chomsky. *Aspects of the Theory of Syntax*. The M.I.T. Press, Cambridge, Massachusetts, 1965.

[20] Noam Chomsky. *Syntactic Structures*. Mouton de Gruyter, Berlin-New York, second edition, 2002.

[21] Gennaro Costagliola, Andrea Delucia, Sergio Orefice, and Giuseppe Polese. A Classification Framework to Support the Design of Visual Languages. *Journal of Visual Languages and Computing*, 13(6):573–600, 2002.

[22] Robert M. Edsall. The Parallel Coordiante Plot in Action: Design and Use for Geographic Visualization. *Computational Statistics & Data Analysis*, 43:605–619, 2003.

[23] Steven Feiner and Clifford Beshers. Worlds Within Worlds: Metaphors for Exploring n-dimensional Virtual Worlds. In *Proceedings of UIST 1990*, pages 76–83, New York, 1990. ACM Press.

[24] Paolo Fiorini and Alfred Inselberg. Configuration Space Representation in Parallel Coordinates. In *Proceedings of the IEEE International Conference on Robotics and Automation 1989*, volume 2, pages 1215–1220, Los Alamitos, CA, 1989. IEEE Computer Society Press.

[25] Camilla Forsell, Stefan Seipel, and Mats Lind. Simple 3D Glyphs for Spatial Multivariate Data. In *Proceedings of the IEEE Symposium on Information Visualization 2005(INFOVIS 2005)*, pages 119–124, Los Alamitos, CA, 2005. IEEE Computer Society Press.

[26] Ying-Huey Fua, Matthew O. Ward, and Elke A. Rundensteiner. Hierarchical Parallel Coordinates for Exploration of Large Datasets. In *Proceedings of IEEE Visualization 1999 (VIS 1999)*, pages 43–50, Los Alamitos, CA, 1999. IEEE Computer Society Press.

[27] Evelyn Goldsmith. *Research into Illustration: An Approach and a Review*. Cambridge University Press, Cambridge, England, 1984.

[28] Volker Haarslev. *Theory of Visual Languages*, chapter A Fully Formalized Theory for Describing Visula Notations, pages 261–292. Springer-Verlag, New-York, 1998.

[29] Steven T. Hackstadt and Allen D. Maloney. Visualizing Parallel Programs and Performance. *IEEE Computer Graphics and Applications*, 15(4):12–14, oct 1995.

[30] Helwig Hauser, Florian Ledermann, and Helmut Doleisch. Angular Brushing of Extended Parallel Coordinates. In *Proceedings of the IEEE Symposium on Information Visualization 2002 (InfoVis 2002)*, pages 127–130, Los Alamitos, CA, 2002. IEEE Computer Society Press.

[31] Christopher G. Healey. Choosing Effective Colours for Data Visualization. In *Proceedings of the 7th IEEE conference on Visualization '96*, pages 263–270, 1996.

[32] Christopher G. Healey and James T. Enns. Large Datasets at a Glance: Combining Textures and Colors in Scientific Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 5(2):145–167, April 1999.

[33] Robert E. Horn. *Visual Language–Global Communication for the 21th Century*. MacroVU, Inc, Bainbridge Island, Washington, Bainbridge Island, Washington, 1998.

[34] A S Hornby. *Oxford Advanced Learners Dictionary of Current English*. Oxford University Press, sixth edition, 2000.

[35] Alfred Inselberg. The Plane with Parallel Coordinates. *The Visual Computer*, 1(2):69–91, October 1985.

[36] Alfred Inselberg. Multidimensional Detective. In *Proceedings of the IEEE Symposium on Information Visualization 1997 (InfoVis 1997)*, pages 100–107, Los Alamitos, CA, 1997. IEEE Computer Society Press.

[37] Alfred Inselberg. Visualization and Knowledge Discovery for High Dimensional Data. In *Proceedings of the Second International Workshop on User Interfaces to Data Intensive Systems (UIDIS 2001)*, pages 5–24, Los Alamitos, CA, 2001. IEEE Computer Society Press.

[38] Alfred Inselberg and Tova Avidan. Classification and Visualization for High-Dimensional Data. In *Proceedings of the Sixth ACM SIGKDD International Con-*

ference on Knowledge Discovery and Data Mining (KDD 2000), pages 370–374, New York, 2000. ACM Press.

[39] Alfred Inselberg and Bernard Dimsdale. Parallel Coordinates: A Tool for Visualizing Multi-Dimensional Geometry. In *Proceedings of IEEE Visualization 1990 (VIS 1990)*, pages 361–378, Los Alamitos, CA, 1990. IEEE Computer Society Press.

[40] Alfred Inselberg, Mordechai Reif, and Tuval Chomut. Convexity Algorithms in Parallel Coordinates. *Journal of the ACM*, 34(4):765–801, October 1987.

[41] Christian Jacob. Genetic L-System Programming: Breeding and Evolving Artificial Flowers with Mathematica. In *Proceedings of the International Mathematica Symposium (IMS 95)*, pages 215–222, 1995.

[42] Norman Wittels Jeffrey LeBlanc, Matthew O. Ward. Exploring N-dimensional Databases. In *Proceedings of the 1st conference on Visualization '90*, pages 230–239, 1990.

[43] Brian Johnson and Ben Shneiderman. Treemaps: a Space-filling Approach to the Visualization of Hierarchical Information Structures. In *Proceedings of the 2nd International IEEE Visualization Conference*, pages 284–291, 1991.

[44] Robert R. Johnson. Visualization of Multi-Dimensional Data with Vector-Fusion. In *Proceedings of IEEE Visualization 2000 (VIS 2000)*, pages 297–302, Los Alamitos, CA, 2000. IEEE Computer Society Press.

[45] Eser Kandogan. Visualizing Multi-dimensional Clusters, Trends, and Outliers Using Star Coordinates. In *Proceedings of the Seventh ACM SIGKDD International Con-*

ference on Knowledge Discovery and Data Mining (KDD 2001), pages 107–116, New York, 2001. ACM Press.

[46] Daniel A. Keim. Pixel-Oriented Database Visualizations. *ACM SIGMOD Record*, 25(4):35–39, December 1996.

[47] Daniel A. Keim. Information Visualization and Visual Data Mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, January 2002.

[48] P. Keller and M. Keller. *Visual Cues: Practical Data Visualization*. IEEE Computer Society Press, Los Alamitos, CA, 1993.

[49] Jun Kong and Kang Zhang. On a Spatial Graph Grammar Formalism. In *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing 2004*, pages 102–104. IEEE CS Press, 2004.

[50] John Lamping and Ramana Rao. The Hyperbolic Browser: A Focus + Context Technique for Visualizing Large Hierarchies. *Journal of Visual Languages and Computing*, 7(1):33–55, 1996.

[51] Hing-Yan Lee, Hwee-Leng Ong, Eng-Whatt Toh, and Sieu-Kong Chan. A Multi-Dimensional Data Visualization Tool for Knowledge Discovery in Databases. In *19th International Computer Software and Applications Conference (COMPSAC'95)*, pages 26–31, Dallas, Texas, USA, August 9-11 1995. IEEE Computer Society.

[52] Haim Levkowitz. ColorCenterColormaps. Web site at `http://www.cs.uml.edu/~haim/ColorCenter/ColorCenterColormaps.htm`. visited April 19, 2005.

[53] Haim Levkowitz and Gabor T. Herman. Color Scales for Image Data. *IEEE Computer Graphics and Applications*, 12(1):72–80, January 1992.

[54] Aristid Lindenmayer. Mathematical Models for Cellular Interaction in Development, parts i and ii. *Journal of Theoretical Biology, vol, 18 1968, pp. 280315*, 18:280–315, 1968.

[55] Iouri Lotman. *La Structure du Texte Artistique*. Gallimard, Paris, 1973 (original 1970).

[56] Kim Marriott. Constraint Multiset Grammars. In *Proceedings of the IEEE Symposium on Visual Languages*, pages 118–125. IEEE CS Press, 1994.

[57] Kim Marriott, Bernd Meyer, and Kent B.Wittenburg. *Theory of Visual Languages*, chapter A Survey of Visual Language Specification and Recognition, pages 5–85. Springer-Verlag, New York, 1998.

[58] Allen R. Martin and Matthew O. Ward. High Dimensional Brushing for Interactive Exploration of Multivariate Data. In *Proceedings of IEEE Visualization 1995 (VIS 1995)*, pages 271–278, Los Alamitos, CA, 1995. IEEE Computer Society Press.

[59] Scott McCloud. *Understanding Comics*. HarperCollins, New York, NY, 1993.

[60] Tamara Munzner. H3: Laying out Large Directed Graphs in 3D Hyperbolic Space. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis '97)*, pages 2–10, 1997.

[61] Wei Peng, Matthew O. Ward, and Elke A. Rundensteiner. Clutter Reduction in Multi-Dimensional Data Visualization Using Dimension Reordering. In *Proceed-*

ings of the IEEE Symposium on Information Visualization 2004 (InfoVis 2004), pages 89–96, Los Alamitos, CA, 2004. IEEE Computer Society Press.

[62] Przemyslaw Prusinkiewicz. Simulation Plants and Plant Ecosystems. *Communications of the ACM*, 43(7):84–93, 2000.

[63] Ramana Rao and Stuart K. Card. The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus + Context Visualization for Tabular Information. In *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*, pages 318 –322, 1994.

[64] Jan Rekers and Andy Schurr. Defining and Parsing Visual Languages with Layered Graph Grammars. *Journal of Visual Languages and Computing*, 8(1):27–55, 1997.

[65] Theresa-Marie Rhyne, Melanie Tory, Tamara Munzner, Matthew O. Ward, Chris Johnson, and David H. Laidlaw. Information and Scientific Visualization: Separate but Equal or Happy Together at Last. In *IEEE Visualization*, pages 619–621, 2003.

[66] William Ribarsky, Eric Ayers, John Eble, and Sougata Mukherjea. Glyphmaker: Creating Customized Visualizations of Complex Data. *Computer*, 27(7):57–64, July 1994.

[67] George G. Robertson, Jock D. Mackinlay, and Stuart K. Card. Cone Trees: Animated 3D Visualizations of Hierarchical Information. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, pages 189–194, 1991.

[68] Geraldine E. Rosario, Elke A. Rundensteiner, David C. Brown, Matthew O. Ward, and Shiping Huang. Mapping Nominal Values to Numbers for Effective Visualization. *Information Visualization Journal*, 3(2):80–95, 2004.

[69] Fernande Saint-Martin. *Semiotics of Visual Language*. Indiana University Press, Bloomington and Indianapolis, 1990.

[70] Christopher D. Shaw, James A. Hall, Christine Blahut, David S. Erbert, and D. Aaron Roberts. Using Shape to Visualize Multivariate Data. In *Proceedings of the 1999 Workshop on New Paradigms in Information Visualization and Manipulation*, pages 17–20, New York, 1999. ACM Press.

[71] Ben Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings of the IEEE Symposium on Visual Languages*, pages 336–343, Los Alamitos, CA, USA, 1996. IEEE Press.

[72] Harri Siirtola. Direct Manipulation of Parallel Coordinates. In *Proceedings of IEEE Visualization 2000 (VIS 2000)*, pages 373–378, Los Alamitos, CA, 2000. IEEE Computer Society Press.

[73] John Stasko. An Evaluation of Space-filling Information Visualizations for Depicting Hierarchical Structures. *International Journal of Human-Computer Studies*, 53(5):663–694, November 2000.

[74] John Stasko and Eugene Zhang. Focus+Context Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualizations. In *Proceedings of the IEEE Symposium on Information Visualization 2000 (InfoVis 2000)*, pages 57–68, Los Alamitos, CA, 2000. IEEE Computer Society Press.

[75] Maureen C. Stone. *A Field Guide to Digital Color*. A K Peters, Ltd., Natik, Massachusetts, 2003.

[76] Bruce Tognazzini. *Tog on Software Design*. Addison-Wesley Professional, 1995.

[77] Christian Tominski, James Abello, and Heidrun Schumann. Axes-Based Visualizations with Radial Layouts. In *Proceedings of the 2004 ACM Symposium on Applied Computing*, pages 1242–1247, New York, 2004. ACM Press.

[78] Edward R Tufte. *Envisioning Information*. Graphics Press, Cheshire, Connecticut, 1990.

[79] Edward R. Tufte. *Visual Explanations: Images and Quantities, Evidence and Narrative*. Graphics Press, Cheshire, Connecticut, 1997.

[80] Jarke van Wijk and Huub van de Wetering. Cushion Treemaps: Visualization of Hierarchical Information. In *Proceedings of the IEEE Symposium on Information Visualization 1999 (INFOVIS99)*, page 73, Los Alamitos, CA, USA, 1999. IEEE Press.

[81] Matthew O. Ward. XmdvTool: Integrating Multiple Methods for Visualizing Multivariate Data. In *Proceedings of IEEE Visualization 1994 (VIS 1994)*, pages 326–333, Los Alamitos, CA, 1994. IEEE Computer Society Press.

[82] Matthew O. Ward. Creating and Manipulating N-Dimensional Brushes. In *Proceedings of Joint Statistical Meeting 1997*, pages 6–14, 1997.

[83] Matthew O. Ward. A Taxonomy of Glyph Placement Strategies for Multidimensional Data Visualization. *Information Visualization*, 1(3–4):194–210, 2002.

[84] Colin Ware. *Information Visualization–Perception for Design*. Morgan Kaufmann Publishers, San Francisco, 2000.

[85] Edward J. Wegman. On Some Mathematics for Visualizing High Dimensional Data. *The Indian Journal of Statistics*, 64(Series A, Pt. 2):429–452, 2002.

[86] Eric A. Wernert, Donald K. Berry, John N. Huffman, and Craig A. Stewart. Tree3D-A Systemfor Temporal and Comparative Analysis of Phylogenetic Trees. In *Proceedings of the IEEE InfoVis 2003 Poster Compendium*, pages 114–115, Los Alamitos, CA, 2003. IEEE Computer Society Press.

[87] Nelson Wong, Sheelagh Carpendale, and Saul Greenberg. EdgeLens: An Interactive Method for Managing Edge Congestion in Graphs. In *Proceedings of the IEEE Symposium on Information Visualization 2003 (InfoVis 2003)*, pages 51–58, Los Alamitos, CA, 2003. IEEE Computer Society Press.

[88] Pak Chung Wong and R. Daniel Bergeron. Multivariate Visualization Using Metric Scaling. In *Proceedings of IEEE Visualization 1997 (VIS 1997)*, pages 111–118, Los Alamitos, CA, 1997. IEEE Computer Society Press.

[89] Jing Yang, Anilkumar Patro, Shiping Huang, Nishant Mehta, Matthew O. Ward, and Elke A. Rundensteiner. Value and Relation Display for Interactive Exploration of High Dimensional Datasets. In *Proceedings of the IEEE Symposium on Information Visualization 2004 (InfoVis 2004)*, pages 73–80, Los Alamitos, CA, 2004. IEEE Computer Society Press.

[90] Jing Yang, Wei Peng, Matthew O. Ward, and Elke A. Rundensteiner. Interactive Hierarchical Dimension Ordering, Spacing and Filtering for Exploration of High

Dimensional Datasets. In *Proceedings of the IEEE Symposium on Information Visualization 2003 (InfoVis 2003)*, pages 105–112, Los Alamitos, CA, 2003. IEEE Computer Society Press.

[91] Jing Yang, Matthew O. Ward, and Elke A. Rundensteiner. InterRing: An Interactive Tool for Visually Navigating and Manipulating Hierarchical Structures. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis 2002)*, pages 77–84, Los Alamitos, CA, 2002. IEEE Computer Society Press.

[92] Jing Yang, Matthew O. Ward, Elke A. Rundensteiner, and S. Huang. Visual Hierarchical Dimension Reduction for Exploration of High Dimensional Datasets. In *Proceedings of the ACM Symposium on Data Visualisation 2003*, pages 19–28, 282, New York, 2003. ACM Press.