TECHNISCHE UNIVERSITÄT WIEN

VIENNA UNIVERSITY OF TECHNOLOGY

MASTERARBEIT

Visualizing Hierarchically Structured Categorical Data

Ausgeführt am VRVis Zentrum für
Virtual Reality und Visualisierung Forschungs-GmbH

unter der Anleitung von
Priv.-Doz. Dipl.-Ing. Dr.techn. Helwig Hauser
und der Mitbetreuung von
Dipl.-Ing. Harald Piringer

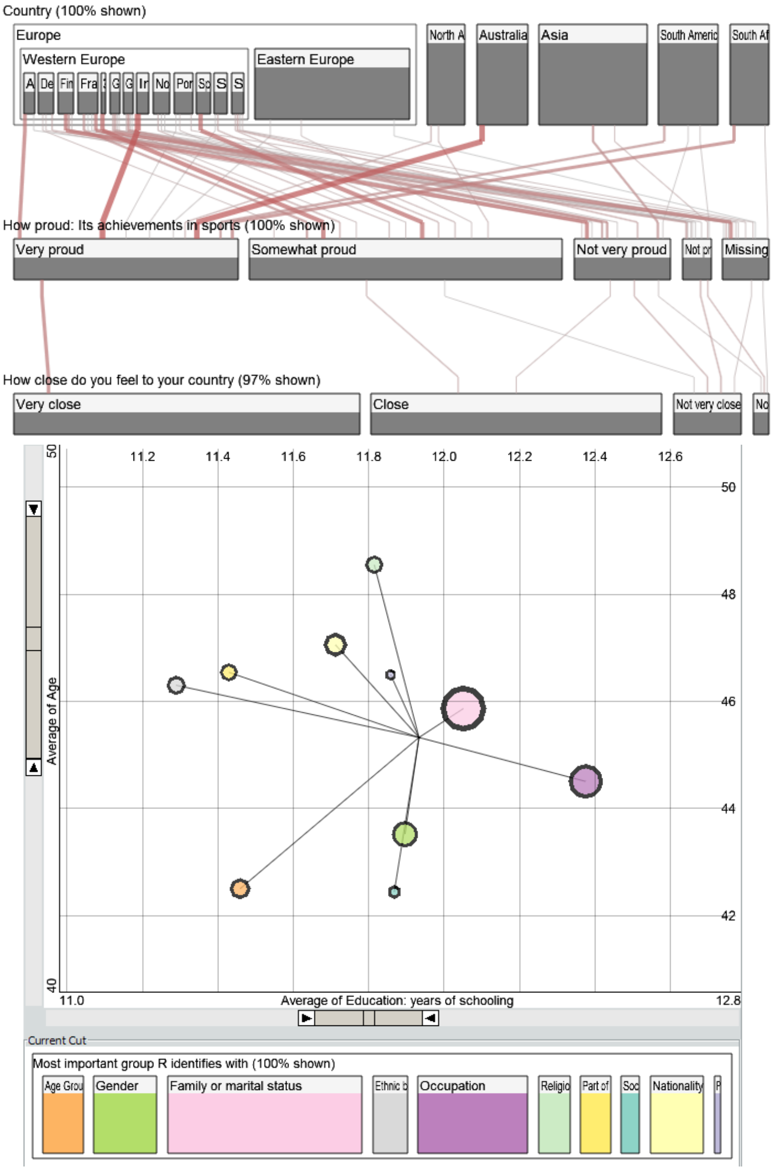
eingereicht an der Technischen Universität Wien,
Fakultät für Informatik
durch

Matthias Buchetics
Wimbergergasse 8/18
1070 Wien

Wien, im November 2007 Matthias Buchetics

Visualizing Categorical Hierarchies

Matthias Buchetics



mailto:mbuchetics@gmail.com
http://www.vrvis.at/via/resources/DA-MBuchetics/

Abstract

Huge amounts of data are generated and collected every day and usually need to be analyzed in order to extract useful information. However, the analysis of large amounts of data can be very challenging. By using graphical representations, visual analysis techniques use the capabilities of the human visual system to assist in this task. Unlike the data used in Scientific Visualization, the field of Information Visualization deals with displaying abstract data, which has usually no spatial structure, thus requiring additional steps in order to map the data to the 2D computer display. This work focuses on a special type of data, namely categorical (qualitative) data. Categorical data dimensions typically have only a limited number of distinct values. The categories may lack any inherent ordering and meaningful ways to compute distances. Additionally, categorical data is often hierarchically structured. These characteristics make the visualization of categorical data variables challenging. Traditional (item-based) visualization techniques are usually not ideal for presenting categorical.

Besides surveying available categorical data visualization techniques, the main contribution of this work are two new approaches for the visualization of hierarchically structured, categorical data. The first technique, *Parallel Hierarchies*, visualizes multiple hierarchies simultaneously, using a parallel axes layout and using the frequency of each category to scale its respective visual representation. Categories of adjacent hierarchies are connected and statistics are used to emphasize relationships between categories. The second presented visualization, the *Hierarchical Scatterplot*, is a novel approach to visualize a categorical hierarchy in respect to two numerical dimensions. Again, statistics are used to support this task.

Both approaches allow for an interactive data analysis and are integrated in an existing visual analysis framework.

Kurzfassung

Jeden Tag werden riesige Datenmengen produziert, was die Analyse dieser Daten sehr anspruchsvoll macht. Mit der Darstellung von grafischen Repräsentationen anstelle von Tausenden Zahlen, machen sich Visualisierungstechniken die Fähigkeiten des menschlichen Sehsystems bei der Unterstützung in dieser Aufgabe zu nutzen. Der Bereich der Informationsvisualisierung beschäftigt sich hauptsächlich mit abstrakten Daten. Abstrake Daten haben normalerweise keine räumliche Strukture und benötigen daher zusätzliche Schritte, um die Daten auf den Bildschirm abzubilden. Kategorische Daten zeichnen sich in der Regel durch eine limitierte Anzahl von möglichen Ausprägungen aus und haben nicht notwendigerweise eine Ordnung, räumlichen Zusammenhang oder Distanz. Zusätzlich sind kategorische Daten oft hierarchisch strukturiert. Diese Merkmale machen die Visualisierung von kategorischen Variablen schwer und traditionelle Visualisierungstechniken reichen meist nicht aus, um sie sinnvoll darzustellen. Im Folgenden werden aktuell verfügbare Techniken zur Visualisierung von kategorischen Daten vorgestellt.

Der wichtigste Beitrag dieser Arbeit ist allerdings die Präsentation von zwei neuen Visualierungstechniken für hierarchisch strukturierte, kategorische Daten. Die erste Technik, *Parallel Hierarchies*, stellt mehrere Hierarchien gleichzeitig mit Hilfe eines Parallelachsen-Layouts dar und verwendet die Häufigkeiten der einzelnen Kategorien, um ihre grafische Darstellung zu skalieren. Kategorien von benachbarten Hierarchien werden dabei grafisch verbunden und interessante Verbindungen mit aufgrund von statistischen Berechnungen hervorgehoben. Da die meisten Datensätze aber weder ausschließlich numerisch, noch ausschließlich kategorisch sind, beschreibt der zweite Ansatz eine neuartige Technik, den *Hierarchical Scatterplot*, um eine kategorische Hierarchie in Bezug auf zwei numerische Dimensionen darzustellen. Auch hier unterstützt Statistik die Aufgabe.

Beide Techniken sind für eine interaktive Analyse der Daten ausgelegt und wurden in ein bestehendes Visualisierungssystem integriert.

Contents

Chapter 1

Introduction

Due to the tremendous performance increase of computers in the last years (Moore's law [48]), the amount of data being produced increases at an incredible rate. Advances in technology allow the collection and storage of huge amounts of information gathered by various sources such as financial transactions, medical data, or surveys. Researchers from the University of Berkeley estimate that, every year, more than 1 Exabyte (= 1 Million Terabytes) of data is generated, of which a large portion is available in digital form [33]. The more difficult task, however, is analyzing the data in order to find valuable information which may be hidden in the data sets and efficient tools to support the user are required. Visual analysis techniques use the great bandwidth offered by the human visual system to support this task by communicating the data in a visual form [24]. Graphical representations are used instead of displaying thousands of numbers to ease the exploration and analysis of the data.

This work aims at a specific topic, i.e. the visualization of categorical data. Typical data sets often consist of hundreds of dimensions which may be separated into numerical and categorical. Because categorical data has a limited number of distinct values and characteristics like the ordering or distance between categories may not be given or possible to calculate, the visualization of this data type is especially challenging and traditional visualizations are often not effective. Furthermore, data is frequently structured in a hierarchy, also requiring appropriate visualization.

In the following chapters two new visualization techniques for hierarchically structured, categorical data are proposed. The first technique, called Parallel Hierarchies, is based on the work of Bendix et al. [7]. Built upon the idea of Parallel Sets, Parallel Hierarchies is an approach to visualize multiple dimensions to find associations between them. The Hierarchical Scatterplot, on the other hand, is a novel approach to visualize categorical together with numerical data. Based on aggregates of subsets of the data, the technique can be used to analyze the distribution of categories within numerical dimensions. Both techniques were integrated in an existing information visualization tool (Bulk Analyzer) and designed using common visualization techniques like interactive view linking and brushing.

1.1 Visualization

> *Visualize: to recall or form mental images or pictures* (Random House Unabridged Dictionary, Random House, Inc. 2006).

Basically, visualization is the use of any graphical representation, for example images, diagrams or animations, to communicate a message [78]. While visualization has been used to present abstract as well as concrete ideas since the dawn of man, it is now used in computer science to transform complex and abstract data into an image on the screen. Visualization makes use of the human visual system and brain capabilities using external resources for cognition, hypothesis building, and reasoning. This process, called external cognition [10] is one of the main advantages of visualization since it uses the very large bandwidth of the human visual system to support understanding and decision making. Card and Shneiderman [10] point out the importance of interaction in visualization, which allows the user to change the visualization constantly to react to new insights gained while analyzing the shown information.

1.1.1 Scientific and Information Visualization

Visualization is often divided in two fields: Scientific Visualization and Information Visualization. As Matt Ward points out [55] Information Visualization and Scientific Visualization share common goals and techniques, which makes a clear separation of the two very hard. Although there is no clear consensus on the boundaries there is a commonly used distinction.

Scientific Visualization

Scientific Visualization mostly deals with inherently spatial data (2-dimensional or 3-dimensional) [24] such as MRI data or wind flows. It is often further divided into the areas of Volume Visualization and Flow Visualization and others.

Information Visualization

On the other hand, Information Visualization (InfoVis) can be seen in contrast to Scientific Visualization (SciVis) because it is usually used to visualize abstract data. Examples include survey results and financial transactions. Unlike the data used in SciVis the non-scientific, abstract data of InfoVis does not have an inherent mapping to space. Whereas the spatial layout of SciVis data makes it easier to visualize intuitively, additional steps have to be performed to map InfoVis data to the computer screen.

The two fields do not only share goals and techniques, but they also increasingly share data making a distinction even harder. Predominantly abstract data sets found in InfoVis may include scientific data, like the location of a store in longitude and latitude and scientific data could include abstract meta data. Therefore, Chris Johnson thinks that the goal is to create integrated visualization and analysis capabilities that use the best of Information and Scientific Visualization research techniques and to create new integrated ”Scientific-Information” Visualization software systems [55].

1.1.2 Objectives

The overall goal of visualization is to allow the human user to gain insight into the presented data. Using one’s perceptual abilities, conclusions can be drawn while interacting with the visualization. One may distinguish between three types of goals for visualization [32]:

Exploration

Data exploration (also known as explorative analysis [32]) is the process of gaining insight into data without any given hypothesis. The main goals are the identification of structures and trends in dataset which have not been discovered before [73]. Interaction is essential and the graphical visualizations have to support the user in the fact finding process. The verification of results discovered during the exploration process can for example be further investigated using InfoVis techniques (see *confirmation*) or data mining methods.

Example: A manager notices that one of the company’s stores sold considerably more products than the other stores and wants to find the cause. Using the interactive tools of a visualization system, the manager can analyze the data provided by all stores to find possible answers.

Confirmation

The starting point is a hypothesis about the data, therefore this process is sometimes called *hypothesis testing.* In contrast to exploration the confirmative analysis is oriented towards goals, which are known in advance. Given the visualization of the chosen data the user should be able to decide whether the hypothesis was correct or not.

Example: A patient shows symptoms of a brain tumor. The doctors analyze the visualization of data gathered by a computer tomography to find out, whether their assumption was correct or not.

Presentation

Most people think of visualization as a way to present and explain complex facts to people. Adequate techniques are chosen to display the facts or data in a way that others can understand the communicated ideas. In data analysis, this is often used as a post processing step after the data has been analyzed. Presentation techniques are widespread and can be found in common consumer software like Microsoft Excel.

Example: A recently launched marketing campaign doubled the company's revenue. By using diagrams, the company is able to present their success in a comprehensible way to its shareholders.

The described categories above often overlap; explorative and confirmative analysis are often used together in an interactive way.

Visual Information Seeking Mantra

Visual data exploration often follows the *Visual Information Seeking Mantra* as described by Shneiderman [61]:

> ***Overview first, zoom and filter, then details on demand***

In cases where little is known about data at the beginning of the data exploration process the user needs to be able to identify interesting regions or patterns in the data as a first step, requiring a kind of *overview. Zooming* can then be used to focus on such patterns and the user may decide to *filter* out currently unimportant information. Drilling down in interesting regions even further allows to reveal *details on demand.* InfoVis techniques can be used for all steps in the exploration process, requiring the visualization to display data at different granularity levels and provide interactive controls (see chapter 2.2 for details).

Keim extended Schneiderman's Visual Information Seeking Mantra to get the *Visual Analytics Mantra* [34] which addresses the problem, that fully visual and interactive methods are often not sufficient for very large data sets. Therefore, the data is *analyzed first* and only the important parts are visualized.

> ***Analyse first, show important, zoom, filter and analyse further, details on demand***

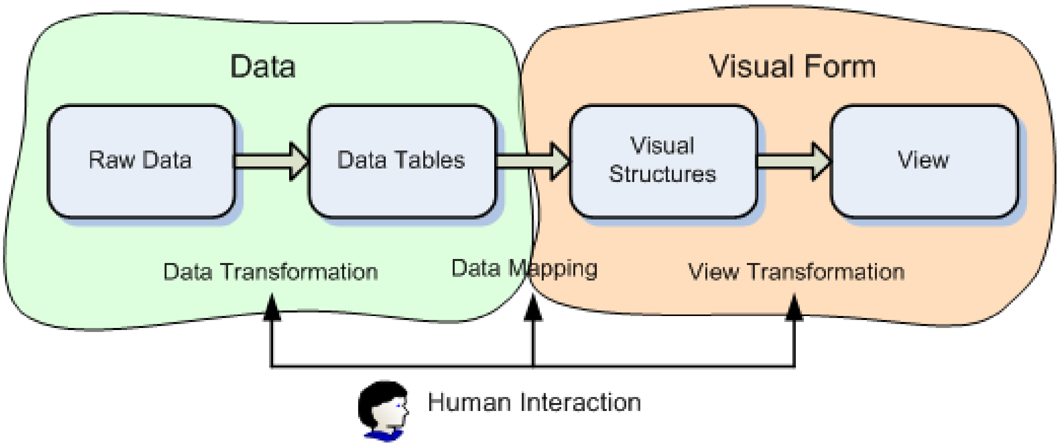


Figure 1.1: Information Visualization Pipeline (image adapted from Card et al. [10])

1.2 Information Visualization

> *Information Visualization is the communication of abstract data through the use of interactive visual interfaces* (Keim et. al., Challenges in Visual Data Analysis, 2006 [36]).

1.2.1 Information Visualization Pipeline

The Information Visualization Pipeline (Figure 1.1) was first proposed by Card et al. [10] and shows the process of converting raw data to visual representations. First, the raw data is transformed into a convenient and well-organized data format. The result of this step (called *Data Transformation*) is typically a dataset containing data entities with data attribute values associated. Often the dataset is stored as tables, with columns being the data attributes (or dimensions) and rows containing single data records. Data processing such as filtering and aggregation can be applied to further adapt the data for analysis. The second step is the *Data Mapping*, which is also the heart of the Information Visualization pipeline. It is used to generate visual structures based on the data or subsets of it. It is also the transition from data to a visual form after which the information is no longer data dependent. The main challenge here is to identify a visual structure which is suitable for the specific kind of data and supports the user. Next, the visual structures are mapped to the screen and the user is provided with various *View Transformations* to interact with the view (e.g. zooming and panning). The resulting view is presented to the user, who may customize any of the steps of the pipeline to adjust the visualization to match his or her goals.

Ind. Var.	Dep. Var.	Example
1	1	1D Function
2	1	Heightfield
2	2	2D Vectorfield
3	1	3D Density Distribution
3	3	3D Flows and Streams
(1)	N	Database Table

Table 1.1: Examples of data with different numbers of independent and dependent variables.

1.2.2 Data Types

Visual representations are usually influenced by the data type and most visualization taxonomies are based on the type of data involved [61]. Common characteristics to classify visualization with respect to the data types include:

Number of data dimensions

One of the most frequently used data characteristics is the data dimensionality. The dimensionality of a data set is defined by the number of independent variables (*arguments* in mathematic functions). Besides the independent variables, there may be one or many dependent variables. For example, in a temporal data set (with time as the only independent variable) one or multiple data values can be associated to each point in time (e.g., a series of stock prices). A 2D height field, on the other hand, consists of two independent variables (the geographical location defined by X and Y) but only one dependent variable (the height at each point). To display this data, all three variables have to be visualized, e.g., by using 3-dimensional space. Table 1.1 lists a number of other examples.

Whereas the visualization of inherently spatial data (e.g. height fields or MRI data sets) is usually quite intuitive, mapping multi-dimensional and multivariate data is not straight forward. Survey data sets often consist of hundred dimensions and even thousands of dimensions are not uncommon in relational databases. More sophisticated visualization techniques are needed for these datasets [33].

Nature of the data

Quantitative data (also called *numerical data*) is data measured (or computed) on a numerical scale, a numerical dataset may be any list of numbers. Such data usually has an inherent order and any two values can be easily compared to each other, allowing distance computations between them. Quantitative data can further be divided into *continuous and discrete data.* Continuous variables may assume any floating point value

Categorical Data	Traditional Visualizations
Discrete	(Usually) Continuous
No inherent ordering	Assumes an ordering
Many dimensions	Few dimensions
Few possible values	Many values

Table 1.2: Discrepancies between qualitative data and traditional visualization techniques.

(e.g., temperature: 2.5°C, 10°C, 15.5°C, 20.29°C, etc.), discrete variables may only assume a finite number of possible values (e.g., number of students in a class: 10, 15, 23, etc.).

Qualitative data or *categorical data* is extremely varied in nature. The variables of such data can be measured using only a limited number of values or categories. Therefore categorical data can be numerical as well as non-numerical. It can further be classified into *ordinal and nominal data.* The latter does not have any inherent order (e.g., family status: single, married, divorced, etc.) whereas ordinal data has a conventional ordering (e.g. months: January, February, March, etc.). Discrete data with only a few unique values is also often handled as categorical data (e.g., coded survey answers: 1 = very good, 2 = good, etc.). Besides that, numerical data can easily be categorized by using intervals and binning to subdivide the data into categories. Visualizing categorical data is challenging because the calculation of numeric differences may not be possible and there does not need to be any inherent ordering. Therefore one of the main issues is to find a suitable visual mapping (see details in Chapter 1.2.4) and due to a number of discrepancies (see Table 1.2) traditional visualizations are often not suited well for categorical data.

More complicated types are often neither numerical, nor categorical. Such variables may be complex numbers, vectors, tensors or other more complex structures, like texts or hypertexts. Whereas vectors can be described by multiple numerical values, they can not easily be compared and lack of an inherent ordering. Texts and hypertexts, on the other hand, can not be described by numbers, making standard visualization techniques usually not applicable [72].

Structure of the data

Linear data is usually stored in data structures like tables or arrays and visualized using graphs, an example is temporal data which is data that changes over time. *Hierarchical data* is intrinsically structured and based on the concept of contained items (e.g. trees). A hierarchy is a collection of items which are linked using a parent-child relationship,

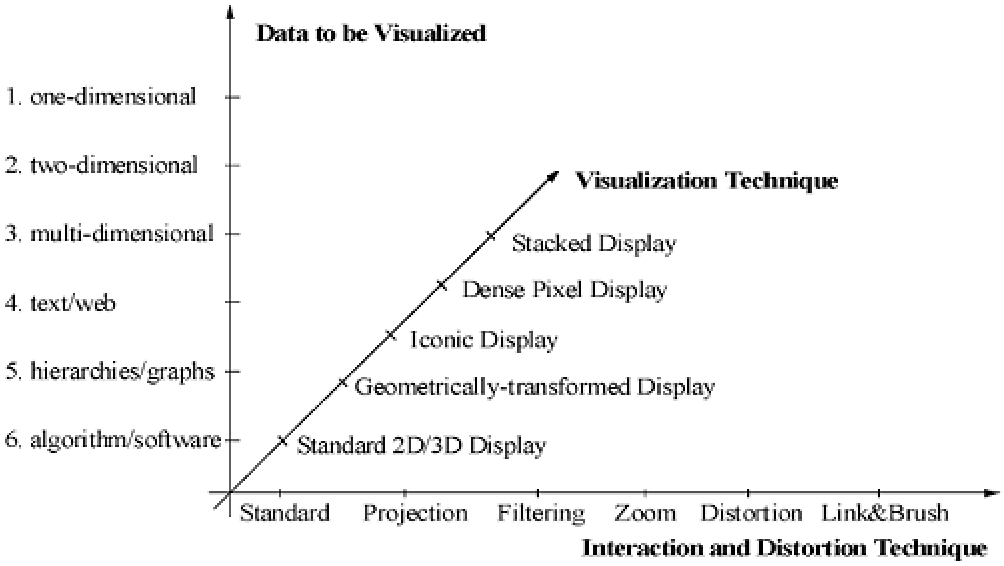


Figure 1.2: Keim's classification of Information Visualization techniques (image courtesy of Keim [33]).

i.e., each items has a link to one parent item. *Network data* is often represented with graph structures and describes any sort of entities linked together.

1.2.3 Visualization Taxonomies

Various taxonomies to classify Information Visualization techniques have been proposed over the last few years. Early approaches where mostly based on the data types before Shneiderman incorporated tasks into his taxonomy [61]. He used seven types of data (1-, 2-, 3-dimensional data, temporal data, multi-dimensional data, tree and network data) and defined seven tasks: overview, zoom, filter, details-on-demand, relate, history, and extract [61]. Keim developed a similar classification [33]. While he also used the data type as one criterion (although a slightly different classification of data), he used the visualization technique as well as interaction and distortion techniques instead of Shneiderman's tasks (see Figure 1.2). The three dimensions in Keim's classification can be assumed as orthogonal, which means any data type can be used with any visualization and any interaction technique. It must also be noted that a visualization system will most likely use different data types as well as a combination of different visualization or interaction techniques.

Whereas Shneiderman's and Keim's taxonomies are (at least partly) based on data type, Tory and Möller proposed a different approach where visualization techniques are based on algorithms rather than data [72]. Assumptions about the data are used to categorize the algorithms. Because those decisions are made by the visualization

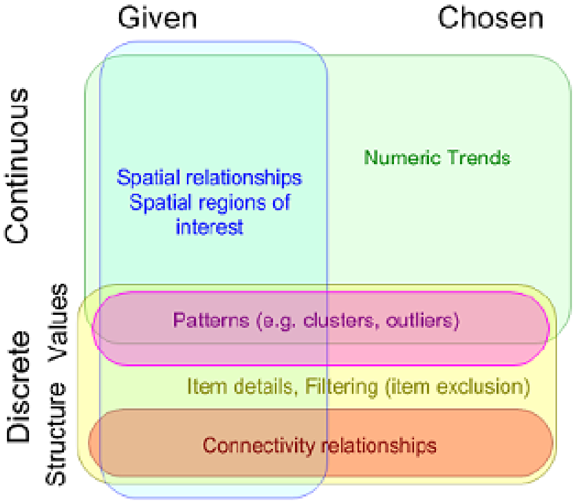

Given
Chosen
Continuous
Discrete
Structure
Values
Spatial relationships
Spatial regions of interest
Numeric Trends
Patterns (e.g. clusters, outliers)
Item details, Filtering (item exclusion)
Connectivity relationships

Figure 1.3: Tory and Möller's classification of visualization tasks. The classification is structured by the data model (continuous or discrete) and by how much the spatialization is constrained (image courtesy of Tory and Möller [72]).

designers, the set of assumptions is called design model in contrast to the user model, which is the set of ideas the user has about the object of study. As stated by Tory and Möller, users will favor techniques that are based on a design model matching their own ideas. Therefore user and design models are linked making a model based classification reasonable. Design models are first classified, in terms of whether they are discrete or continuous. Secondly, they are divided based on the influence of the designer on different attributes like spatialization or color. Examples for continuous models where the spatialization is given by the dataset are medical images. In contrast, the implied design model for categorical data will most likely be discrete and chosen since the spatial layout is not inherent in the abstract data (some categorical data, e.g. categories of temperatures, will lean toward a constrained model). At lower levels the models are further classified similar to the above described data based taxonomies.

1.2.4 Categorical and Hierarchical Data Visualization

As stated in Chapter 1.2.2, categorical data is usually discrete and variables are measured using a limited number of values. Categorical variables usually do not have any inherent ordering, spatial layout, or distance metric. According to Friendly [16] existing visualization techniques for categorical data often tend to be designed for a special purpose and can only be applied to a limited range of tasks (e.g., Mosaic Displays [15] are designed for discovering associations whereas whereas Parallel Coordinates [28] can be used for detecting outliers, analyzing associations or exploring clusters) and specifically designed visualization methods for categorical data have not been as common as visualizations for numeric data. A lot of early techniques are also limited by the amount of data dimensions or the size of the visualized data in general [58].

Visualization techniques designed for categorical data can be classified as either non-transformational or transformational:

Non-transformational Techniques directly map data onto visual representations on the screen and do not need to transform or map it first. Some other data or algorithm is used to calculate a location for the graphical objects, e.g. Glyphs (see Chapter 2.1.2). In general, non-transformational techniques do not work with traditional visualizations such as Parallel Coordinates or Scatterplots.

Transformational techniques first transform each category to a numeric value, which may then be used within standard views. One commonly used mapping transformation is a frequency-based mapping. The frequency of each unique value is calculated and later utilized for spatialization (see frequency-based approaches in Chapter 2.1.3).

While a lot of research has gone into improving the mapping transformation from categorical to numerical data [43, 58], standard InfoVis views are generally not well suited for the display of categorical data if they are not adapted to it (see Chapter 2 for examples). Especially techniques like Parallel Coordinates, which implement a continuous design model (see Tory's and Möller's taxonomy in Chapter 1.2.3) suffer from the discrepancy of design model and user model which is usually discrete if categorical data is used. Categorical data is almost always associated with some meta information, for example the name of the category. This meta information is often important for the user, but is usually not supported by standard views.

Hierarchies are often used to structure categorical data. The hierarchies may be stored in the data set or generated either by the user or classification algorithms. For example, a data set containing financial transactions could be divided into categories of year, which are then further divided in months or days. It is also common to base specialization (”drill down”) and generalization (”roll up”) on certain categories in OLAP (”Online Analytical Processing”) applications (see Chapter 2.3.2) and a simultaneous visualization of different hierarchical levels is often required (e.g. comparing the average financial results of a year to a month).

Many approaches to visualize hierarchies have been proposed over the last few years and many are adaptions of traditional techniques for hierarchical structures (e.g. Hierarchical Parallel Coordinates [18], TreeMaps Layouts for Hierarchical Data [59]).

1.3 Thesis Organization

This thesis is structured as follows: Chapter 2 contains an overview of existing visualization techniques for categorical data and important InfoVis fundamentals. Chapter 3 explains the data model, the views are based on. In Chapter 4, Parallel Hierarchies are described as a visualization technique to visualize multiple hierarchical structured categorical dimensions. Another novel visualization technique, the Hierarchical Scatterplot, allowing for analysis of categorical and numerical data is presented in Chapter 5. In Chapter 6 both methods are evaluated visualizing a real-world data set. Details about the implementation are given in Chapter 7. The thesis is then wrapped up by the summary and conclusion in Chapters 8 and 9.

Chapter 2

State of the Art and Fundamentals

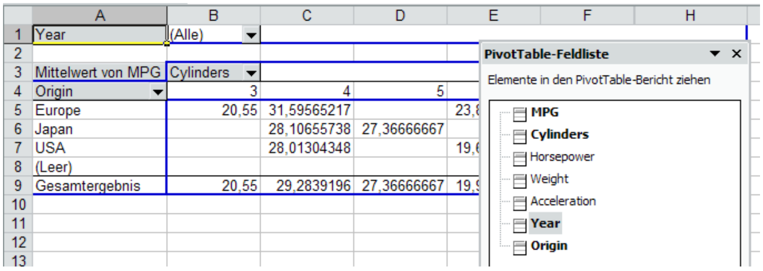
In the introduction it was explained that the visualization of multi-dimensional categorical data provides many challenges. After describing traditional and new Information Visualization methods and how they deal with categorical data, the most important interaction techniques are explained. Later an introduction into the related areas of data mining and "Online Analytical Processing" as well as an overview of the Bulk Analyzer visualization system is given.

2.1 Visualization Metaphors

The subsequent section provides a survey of common InfoVis methods. As the focus of this thesis is on the visualization of hierarchically structured categorical data and due to limited space, only a subset of InfoVis techniques was surveyed. The various approaches are analyzed with categorical data in mind and therefore, advantages and disadvantages with respect to that are pointed out. After starting with spreadsheets and tables, the methods are divided into item-based and frequency-based techniques. Whereas the former display single data points, the latter use the item counts of categories or data intervals for the graphical representations.

2.1.1 Spreadsheets and Tables

A spreadsheet is a rectangular table where values are arranged in rows and columns. Spreadsheets have been used for hundreds of years and when Microsoft introduced the Windows operating system, Excel was one of the first released products [52]. Various aggregations are used to reduce large data sets to a comparable small amount of concise information. Traditional 2-D tables can only compare two dimensions of variables at a time which is a major disadvantage when dealing with multi-dimensional data sets.



	A	B	C	D	E
1	Year	(Alle)			
2					
3	Mittelwert von MPG	Cylinders			
4	Origin	3	4	5	
5	Europe	20,55	31,59565217		23,8
6	Japan		28,10655738	27,36666667	
7	USA		28,01304348		19,6
8	(Leer)				
9	Gesamtergebnis	20,55	29,2839196	27,36666667	19,9
10					
11					
12					
13					

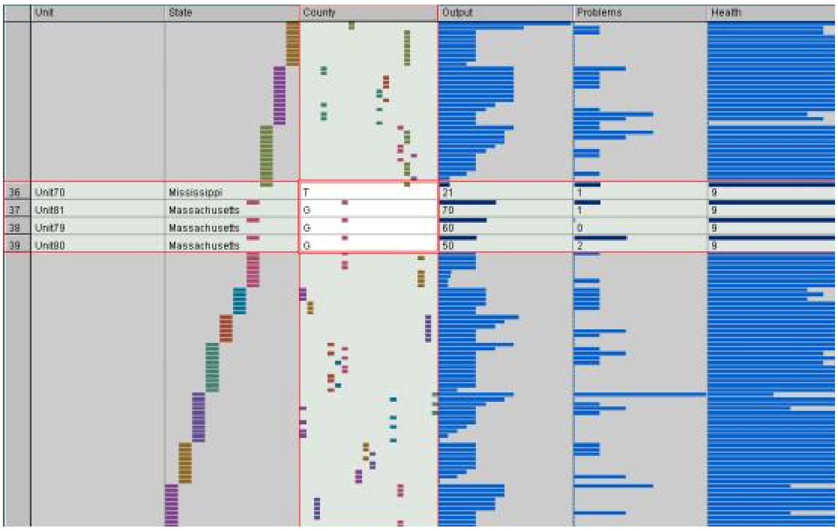
Figure 2.1: Using a pivot table in Microsoft Excel to calculate the average miles per gallon in relation to their origin and number of cylinders.

Pivot tables can be used to summarize large data sets quickly by aggregating the data as they allow the calculation of summary information without requiring the user to write formulas like in spreadsheets. For example, pivot tables can automatically count, sum or count the data of a spreadsheet table. Different dimensions of the dataset can be assigned to rows or columns and arranged dynamically which is called *pivoting*: the users can examine the data from various angles, independent from the layout of the involved spreadsheet tables. Other dimensions can be aggregated and the numbers displayed in the table's cells (see Figure 2.1). They are one of the most popular interfaces for multi-dimensional databases and are often used for OLAP (see Chapter 2.3.2).

Crosstabulations are used to show the joint distribution or frequency of two (or more) variables in a table. Each cell of the table is used to display one cross tabulation. Crosstabulation is useful for categorical data with a limited number of unique values and is used to identify relationships between the crosstabulated variables. It is a very popular method to analyze survey data where each cell of the table displays the number of respondents (or the distribution) that gave a certain combination of responses. A simple form of the crosstabulation is the 2 x 2 table where only two variables with two unique values are used. The fourfold display is a method to visualize 2 x 2 tables [17]. An example including a detailed description of a crosstabulation can be found later in this chapter in Section 2.1.3 (see also Figure 2.11).

Sifer points out that while scatterplots are better for reading clusters, tabular approaches are better for analyzing and comparing distributions [62]. Pivot tables support many dimensions and can also be used for hierarchical structures which makes them a powerful tool.

Table Lens is an interesting approach to visualize an entire data set at once, without the need of scrolling [54]. The Table Lens presents the data in a table, where each row corresponds to a data item and each column represents a data variable. By dynamically distorting the spatial layout of the table, large amounts of information can be displayed. Initially, the data is shown in a compressed form, where categorical data fields are displayed using using colored boxes and variable length bars represent numerical data



	Unit	State	County	Output	Problems	Health
36	Unit70	Mississippi	T	21	1	9
37	Unit81	Massachusetts	G	70	1	9
38	Unit79	Massachusetts	G	60	0	9
39	Unit80	Massachusetts	G	50	2	9

Figure 2.2: The Table Lens allows the user to view large amounts of data at once using a *Focus+Context* approach. Image courtesy of Inxight Software, Inc. [30]

fields (see Figure 2.2). Because rows and columns are not bent by warping, the user can efficiently scan the data and recognize trends and patterns. The user can select certain areas which are then enlarged and detailed information about the data is presented. Additionally, these *focus areas* can easily be moved, enlarged or shrunk.

2.1.2 Item-based Techniques

Item-based techniques display single data points and are the most popular approaches to visualize numerical data. Because categorical data is usually described by a limited number of unique values, item-based visualizations are usually not suitable. Additionally, categorical variables may need to be transformed into numeric values first. Nevertheless, several approaches to visualize categorical data are based upon item-based techniques.

Scatterplots

A scatterplot is a visualization that displays and relates two (or three in the three-dimensional case) quantitative variables of a data set. The data is drawn as a set of points where each point represents one item of the data set (see Figures 2.17 and 2.18 for examples). While a scatterplot does not specify dependent or independent variables it can show various kind of relationships in the data (correlations are suggested by patterns of dots) [67]. Scatterplots are also often used to identify outliers. Their major advantage is the ease of use and interpretation. Although scatterplots usually can handle large data sets better than many other visualization techniques, they are still limited by

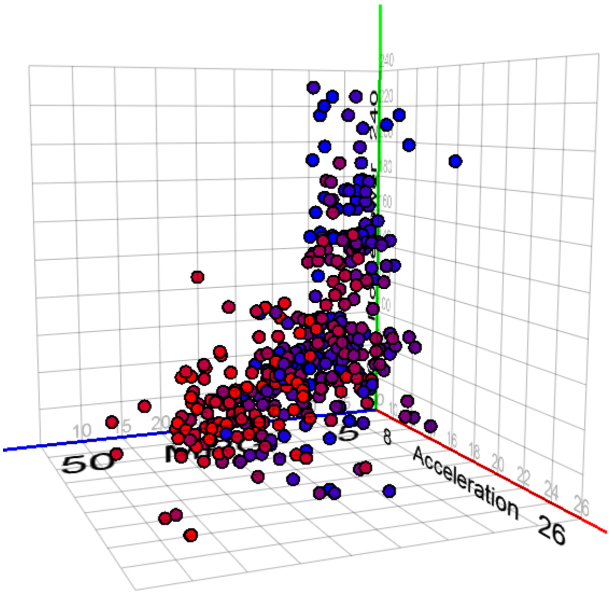


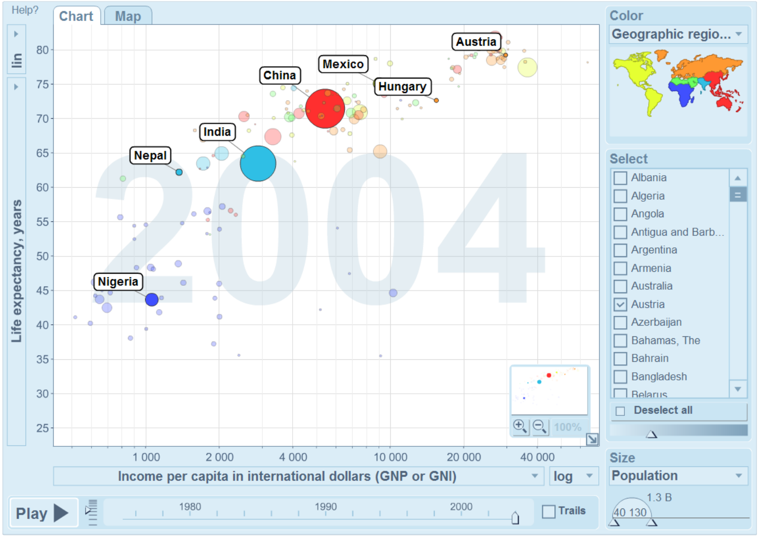
Figure 2.3: 3-D scatterplot using coloring to display a forth dimension.

the number of points reasonably displayed at one time and read by users as well as the number of displayed dimensions.

Three dimensional scatterplots use orthogonal or perspective projections to visualize three dimensions at once. Figure 2.3 also shows how coloring with a transfer function can be used to incorporate an additional dimension. Scatterplot matrices are a method to visualize multiple dimensions where each pairwise combination of dimensions is displayed as a scatterplot. Although conceptionally not limited, the available screen space sets a practical limit to the number of simultaneously shown dimensions.

Scatterplots are generally not very suitable for visualizing categorical data nor can they display any hierarchical structuring in the data set. Ma and Hellerstein show an approach to order categories so they can be better viewed with techniques such as scatterplots [43]. Another typical problem of scatterplots are overlapping points, which are displayed on top of each other and the user can tell how many data items each point represents. One widely used approach to solve this problem, is the use of "jittering" [12]. Jittering scatters the overlapping data points across a wider area, allowing the user to identify the number of data items at each point. Unfortunately, the jitter technique causes data items to be positioned incorrectly, which may affect the interpretation of the data negatively. Therefore, Manson recommends the use of additional attributes (e.g., coloring of points) or animation to solve the problem [45].

Trendanalyzer, a visualization system developed by the Gapminder Foundation [21], allows the analysis of world development indicators (e.g., income per capita, life expectancy, child mortality, etc.) using a scatterplot approach. The countries, represented as circles, are placed on the two numerical axes according to chosen indicators. The size



Color: Geographic regio...

Select
- ☐ Albania
- ☐ Algeria
- ☐ Angola
- ☐ Antigua and Barb...
- ☐ Argentina
- ☐ Armenia
- ☐ Australia
- ☑ Austria
- ☐ Azerbaijan
- ☐ Bahamas, The
- ☐ Bahrain
- ☐ Bangladesh
- ☐ Belarus

☐ Deselect all

Size: Population

Figure 2.4: The visualization software *Trendanalyzer* allows the user to explore the relationship between health and wealth of countries, which are represented by circles. Screenshot of *Gapminder World 2006* [21].

of the circle, as well as its color, also depend on indicators, such as population or income group (see Figure 2.4 for a screenshot).

Spotfire [65] uses a scatterplot as its main visualization view and although several other methods can be chosen by the user, the scatterplot tends to be the most popular as a survey by Kobsa has shown [38].

Glyphs and Icons

In glyph based visualizations, one or more attributes of a data point are mapped to a particular symbol or shape, which is in many cases a metaphor of the respective application domain. Glyphs are used to display multiple dimensions at once, enabling the user to identify similarities, relationships, or anomalies in the data set. See Figure 2.5 for a few examples of glyphs ranging from star glyphs to the well known Chernoff faces [11, 50, 80]. Each glyph is represented by a number of geometric and appearance attributes which are mapped to certain dimensions of the data.

After a glyph has been generated it must be placed on the screen. Various strategies for placing glyphs exist: while sometimes a certain dimension of the raw data is assigned to the position attribute, others are based on the structure of the data set which can be very effective for ordered or hierarchical data. Many glyph placement strategies are explained by Ward [76] and implemented in the XmdvTool [75].

Lee et al. [42] also found that glyph placement is a major challenge in Information Visualization. In an evaluation of several glyph visualizations, glyphs usually lead to

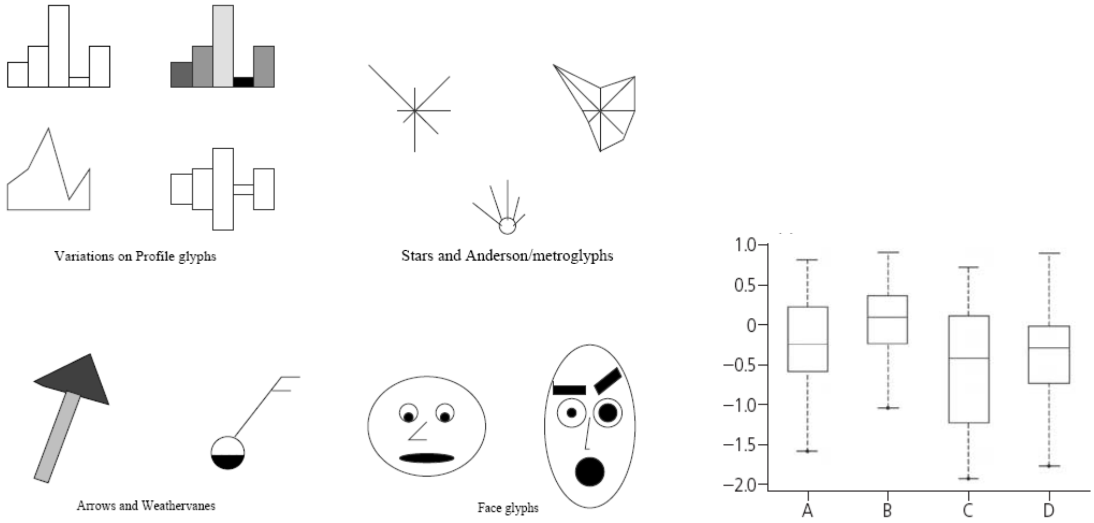


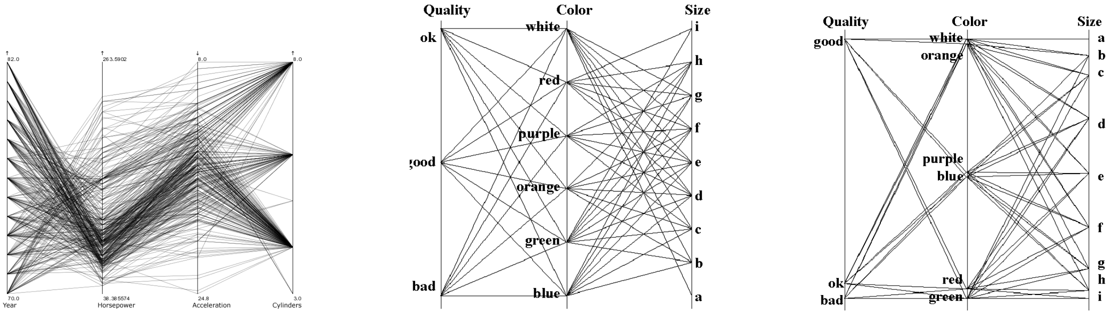
Figure 2.5: The left image shows various examples of glyphs (image courtesy of Ward et al. [76]). The right image shows four plots (image courtesy of Massart [46]).

slow and inaccurate responses. Most users had problems understanding the spatial location of the glyph compared to others.

In addition to the placement problem glyphs have some other drawbacks. First, the number of simultaneously displayed glyphs is limited by the screen space. Depending on the glyph type, there is a minimal size below which an interpretation becomes increasingly hard. Furthermore, users may perceive different appearance attributes differently.

Box Plots (also known as box-and-whisker plot) are a very popular type of glyph [46]. The box plot is a commonly used technique in the field of statistical analysis to quickly interpret the distribution of data (see Figure 2.5 for an example). While regular glyphs represent one or more attributes of a single data point, a box plot visualizes a range of data points. It may therefore be considered a range- or distribution-based technique, since it is neither item- nor frequency-based.

Based on robust statistics the box plot displays the median and the interquartile range (IQR) of a distribution to make it resistant to outliers. The IQR is the range between the third and first quartiles which is the range where the middle 50% of the ranked data are found. The length of the box is equal to the IQR and the position of the median is indicated by a line. The whiskers show the range of the data. Strong outliers are excluded by limiting the range to the third quartile + 1.5 x IQR in one and first quartile - 1.5 x IQR in the other direction. Sometimes ticks or points are drawn to indicate these strong outliers.



(a) Traditional parallel coordinates are mostly used for continuous data. Image courtesy of Hauser et al. [26]

(b) Using parallel coordinates with categorical data.

(c) Pre-processing the data using a DQC approach leading to better results. Images courtesy of Rosario [58]

Figure 2.6: Three examples of Parallel Coordinates.

Parallel Coordinates

Parallel Coordinates are a common technique to visualize multi-dimensional data sets. The approach was first mentioned by Inselberg and Dimsdale [28] and has since been incorporated into many visualization systems to identify correlations between variables.

The main idea is to position the axes in parallel, which has the major advantage compared to scatterplots that this approach has no theoretical limit with respect to the number of simultaneously shown dimensions (the amount of dimension shown at once is only limited by the available screen space which may be enlarged using scrolling).

Each N-dimensional point is represented by a polyline which intersects each axis at the position corresponding to the value of the respective dimension of that entry (see Figure 2.6). It is a special property of Parallel Coordinates that lines are equivalent to points in Cartesian coordinates and points in Parallel Coordinates are equivalent to lines in Cartesian coordinates, so it can be considered a dual space.

The visual structure of Parallel Coordinates make it easy to add additional dimensions by simply adding more axes to the view. The order of the axes is significant, as correlations are perceived most easily for neighboring axes, although the user can typically change this order interactively.

One difficulty of Parallel Coordinates is the cluttering of information when large data sets are displayed. Various approaches such as clustering and proximity-based coloring have been proposed to solve this problem[18]. A first step for large, but not huge data sets is to use transparency when drawing the polylines [31].

Generally, categorical data can not be visualized well using Parallel Coordinates as the categorical values first have to be transformed into numeric values. However, artificial patterns and errors in the interpretation of the visualization can easily be introduced and are sometimes inevitable [58]. As shown in Chapter 1.2.4 the user usually associates categorical data with a discrete model whereas the Parallel Coordinates implement a continuous model, which explains the bad utilization of available screen space. The interpretation of categorical values may be confusing and hard to understand. Moreover, Parallel Coordinates commonly do not display any meta information such as the names of the categories.

Rosario et al.[58] introduced an approach that handles ordering and spacing of categorical values in a Parallel Coordinates system. Before positioning the categories on the parallel axes, the categorical data is pre-processed using a Distance-Quantification-Classing (DQC) approach so that the calculated order considers similarity of the categories. Additionally a degree of similarity is used to space the categories along one axis leading to a more efficient and meaningful location of the data values. While the authors note that the mapping is distance and association-preserving Bendix [6] points out that categorical values naturally lack any distance.

Figure 2.6 shows an example of categorical data visualization using Parallel Coordinates comparing the traditional approach (categorical values are mapped using equal spacing and ordered arbitrarily) to the Distance-Quantification-Classing approach. The results are easier to interpret, but the discrepancy between expected and implemented models still make Parallel Coordinates not very suitable for categorical data.

2.1.3 Frequency-based Techniques

As stated above, categorical data lends itself to a discrete user model. Because most item-based visualization techniques are based on a continuous design model, they are usually not the optimal choice for categorical values. Frequency-based techniques on the other hand, implement a discrete model eliminating the discrepancy of user imagination and presented image [7]. Frequency-based methods use the frequency of each category, i.e., the number of entries within that particular category, to scale visual representations accordingly.

Friendly [15] also states that principles of perception, detection and comparison have suggested that areas are the best representations for frequencies.

Absolute and relative frequencies are often mistakenly interchanged. In this work the term “absolute frequency” refers to the actual item count and it is a discrete number. Relative frequencies on the other hand are normalized by the total number of occurrences. In most cases the relative frequencies are used to compare two statistics and if not stated otherwise the short term frequency is used for relative frequencies.

Bar Charts

Bar charts or bar graphs are a common method for visualizing the frequencies of categorical values or value ranges of continuous data. The frequencies are represented by

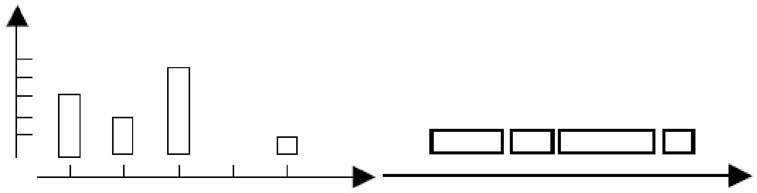


Figure 2.7: Comparison of histogram and bargram, both displaying the same data. Image courtesy of Wittenburg et al. [79]

the areas of the bars. While vertically oriented, equal-width bar charts (histograms) are more familiar to most people and allow for more exact comparisons, horizontally oriented, equal-height bar charts (bargrams) use the available screen space more efficiently.

Histograms are equal-width bar charts where the height of the bar is associated with a frequency. The number of bars (or bins) is either selected by the user (for continuous data) or given by the number of unique values (e.g. a categorical data dimension). Attempts to determine an optimal number of bars are available but make strong assumptions about the data distribution. Therefore it is usually better to give the user the option to change the number of bins according to their needs or the used data set. The frequency of a histogram bin is calculated by comparing each value of the data set to the bin boundaries and counting the ones that fall into the boundary.

1-D histograms are displayed in two dimensions on the screen while 2-D histograms (the boundary is a region and not just a single interval) are usually displayed in 3-D or as a 2-D "height fields". Kosara et al. [41] present several approaches to visualize histograms for time-varying data. Another variant is the cumulative histogram where the cumulative number of observations is counted up to the specified bin.

Generally, histograms are very useful since they give the user a quick and easy-to-understand overview of the distribution of a data dimension. They can also be very powerful in combination with other views (see brushing and view linking in Chapter 2.2.1).

Bargrams are equal-height bar charts and represent the frequencies of the bins by their relative widths. Any bins without data are ignored. As you can see in Figure 2.7, gaps are not shown and this information is therefore lost. Bargrams need less screen space than histograms and multiple bargrams can displayed at once. Wittenburg et al. proposed *parallel bargrams* which, similar to Parallel Coordinates, display multiple dimensions on uniformly spaced axes [79]. In contrast to Parallel Coordinates, the axes are oriented horizontally and no lines are drawn. Value bins are can be brushed revealing possible relationships in other dimensions and bins.

The commercial Information Visualization product *InfoZoom* uses parallel bargrams as the main view (see Figure 2.8) and also allows for the visual analysis of hierarchies and tree structures [64].

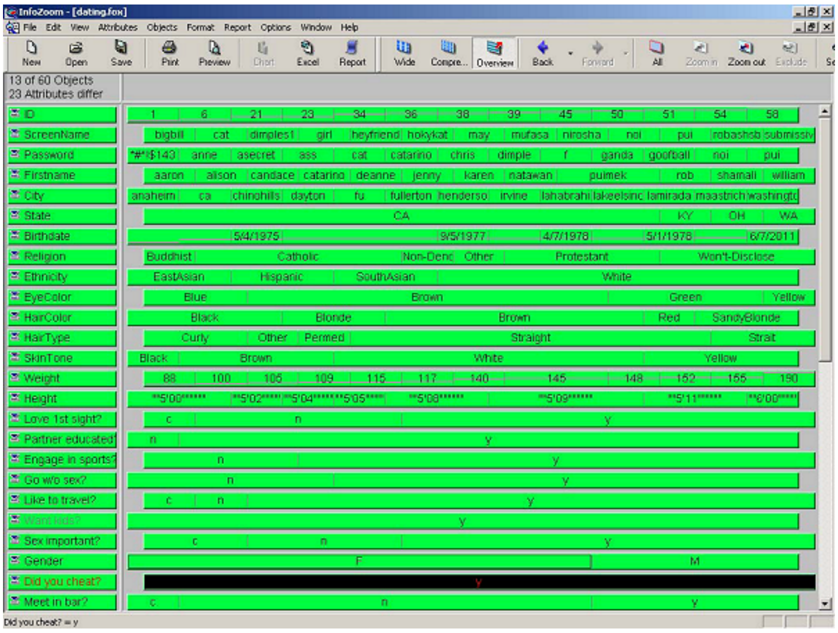

InfoZoom - [dating.fox]
File Edit View Attributes Objects Format Report Options Window Help
New Open Save Print Preview Chart Excel Report Wide Compre... Overview Back Forward All Zoom in Zoom out Exclude Se
13 of 60 Objects
23 Attributes differ
ID 1 6 21 23 34 36 38 39 45 50 51 54 58
ScreenName bigbill cat dimples1 girl heyfriend hokykat may mufasa nirosha noi pui robashsb submissiv
Password *#*!$143 anne asecret ass cat catarino chris dimple f ganda goofball noi pui
Firstname aaron alison candace catarino deanne jenny karen natawan puimek rob shamali william
City anaheim ca chinohills dayton fu fullerton henderso irvine lahabrahi lakeelsinc lamirada maastrich washingtc
State CA KY OH WA
Birthdate 5/4/1975 9/5/1977 4/7/1978 5/1/1978 6/7/2011
Religion Buddhist Catholic Non-Denc Other Protestant Won't-Disclose
Ethnicity EastAsian Hispanic SouthAsian White
EyeColor Blue Brown Green Yellow
HairColor Black Blonde Brown Red SandyBlonde
HairType Curly Other Permed Straight Strait
SkinTone Black Brown White Yellow
Weight 88 100 105 109 115 117 140 145 148 152 155 190
Height **5'00****** **5'02**** **5'04***** **5'05**** **5'08****** **5'09****** **5'11****** **6'00****
Love 1st sight? c n y
Partner educated? n y
Engage in sports? n y
Go w/o sex? n y
Like to travel? c n y
Want kids? y
Sex important? c n y
Gender F M
Did you cheat? y
Meet in bar? c n y
Did you cheat? = y

Figure 2.8: InfoZoom is using parallel bargrams to display multiple dimension on one screen. Image courtesy of Kobsa [38].

Pixel Bar Charts are derived from the bar charts as described before (histogram and bargram) but present data values directly instead of just using the frequency of a bin. The idea can be described as a combination of a bar chart and a scatterplot. The data is first partitioned using a categorical data dimension before single pixels are arranged within a bar using one or two numerical attributes. The pixels are colored according to the data values. Space filling pixel bars solve the problem of traditional bar charts where large portions of the screen space are not used because of differing heights of the bars. The space filling approach scales the whole area assigned to a category, instead of only adjusting height or width of a bar and therefore uses almost all pixels on the screen to display information. Keim et al. [35] extended this approach further to display hierarchical data which allows the user to drill down on selected bars.

Mosaic Display

Mosaic Displays implement a discrete design model and graphically represent the frequencies of the categorical values [17]. Using a recursive space-subdivision algorithm the width and height of the available space is divided alternately using the variables of the assigned dimensions. Friendly [15] enhanced the Mosaic Display by using coloring and shading to map additional information. Other implementations use the color to display the levels of one variable in order to make the categories more visually distinct. Mosaic Displays have further been extended with several interactive features [27].

Although the number of dimensions handled by the algorithm is only limited by the available screen space, Mosaic Displays become difficult to understand when the number of displayed dimensions is larger than three or four.

Dimensional Stacking

Dimensional Stacking is another technique based on a discrete design model making it suitable for multi-dimensional categorical data [5]. Similar to Mosaic Displays, a two-dimensional grid is recursively subdivided by the categories of the assigned dimensions until all dimensions are used. The space is split into small rectangles where the next pair of dimensions is embedded. Whereas Mosaic Displays use the frequency of a category to scale the respective area, the rectangles of Dimensional Stacking are of the same size but filled according to the data values.

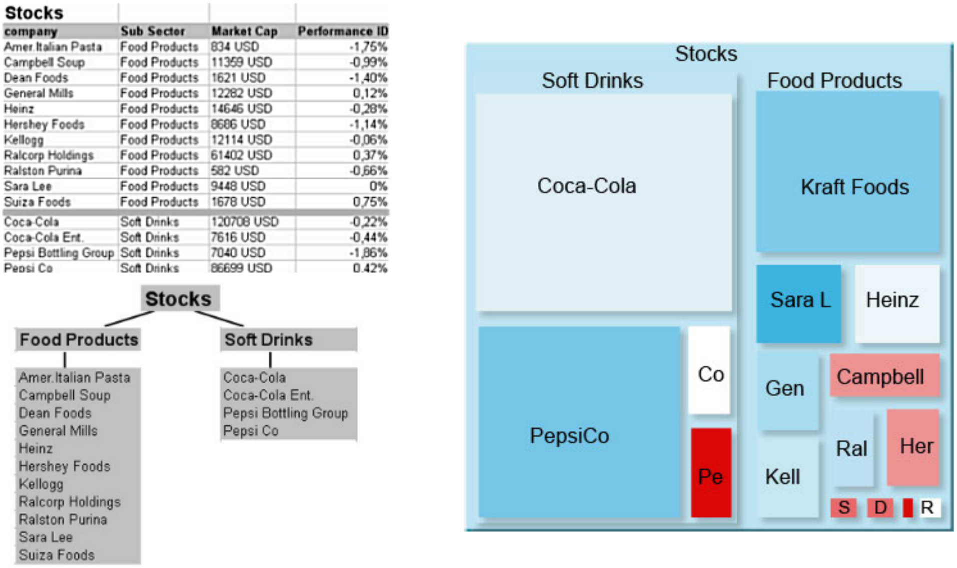
The technique preserves much of the spatial information and can therefore be used to find outliers or clusters in the data set. Dimensional Stacking is limited in the number of reasonably displayed dimensions as well as in interactive features, making it a seldom used technique.

TreeMaps

The most common way of representing hierarchies in a graphical form is a tree visualization. As with other visualization techniques, trees are limited by the size of the screen which is especially an issue when very large hierarchies are visualized. There has been much work on the visualization of large hierarchies resulting in a variety of different approaches. A comparison of several 2-D visualizations of hierarchies was written by Barlow and Neville [4]. Furthermore, a number of popular techniques has evaluated by Kobsa [39].

Whereas traditional tree visualizations represent the tree as a rooted, directed graph with connected nodes, the TreeMap is a space-filling approach where each node is represented by a rectangle whose size is proportional to an attribute of the node, such as the frequency count.

Proposed by Shneiderman [60], TreeMaps have since been a popular visualization technique. Given the discrete design model of TreeMaps they are especially useful for hierarchically structured data sets. The algorithm recursively subdivides the available screen space by traversing the hierarchy topdown and alternatingly splitting the area horizontally and vertically. The areas of the rectangles at one hierarchy level can depend on the frequency count of the respective categories but may also be calculated using a different data dimension (e.g. the total revenue generated). Each node must have total value of its subtree stored or otherwise this data must be aggregated first. Furthermore an additional dimension can be used to color the rectangles accordingly. See Figure 2.9 for an example.



Stocks

company	Sub Sector	Market Cap	Performance ID
Amer.Italian Pasta	Food Products	834 USD	-1,75%
Campbell Soup	Food Products	11359 USD	-0,99%
Dean Foods	Food Products	1621 USD	-1,40%
General Mills	Food Products	12282 USD	0,12%
Heinz	Food Products	14646 USD	-0,28%
Hershey Foods	Food Products	8686 USD	-1,14%
Kellogg	Food Products	12114 USD	-0,06%
Ralcorp Holdings	Food Products	61402 USD	0,37%
Ralston Purina	Food Products	582 USD	-0,66%
Sara Lee	Food Products	9448 USD	0%
Suiza Foods	Food Products	1678 USD	0,75%
Coca-Cola	Soft Drinks	120708 USD	-0,22%
Coca-Cola Ent.	Soft Drinks	7616 USD	-0,44%
Pepsi Bottling Group	Soft Drinks	7040 USD	-1,86%
Pepsi Co	Soft Drinks	86699 USD	0.42%

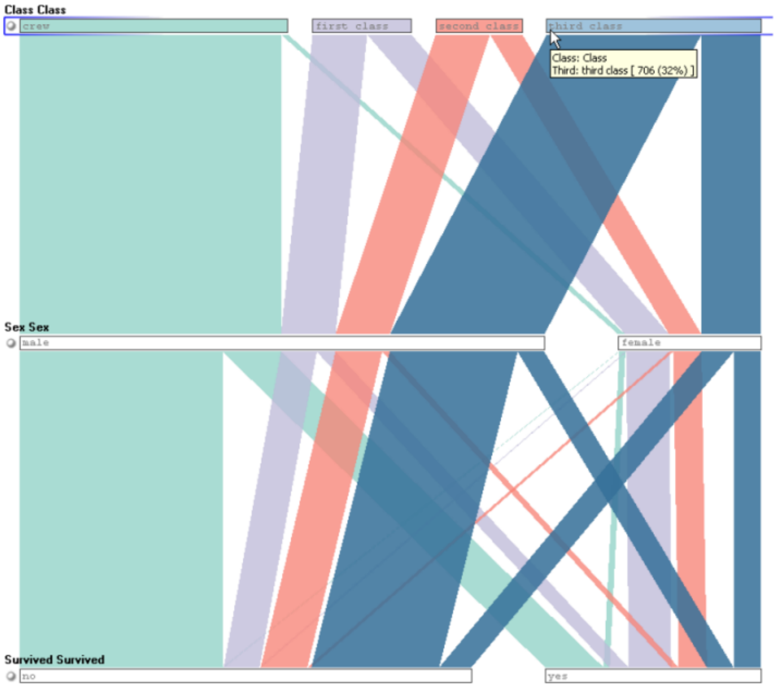
Figure 2.9: Displaying similar data in a spreadsheet, a tree visualization and a TreeMap. The areas of the TreeMap are sized according to the market cap and the performance ID is used for coloring (these values are not displayed in the tree example). Images courtesy of Panopticon Software [63].

The major advantage of TreeMaps is that they utilize the whole available space and provide a good overview of a hierarchical structure as well as information about the data distribution. However, as Schreck et al. [59] point out, the standard TreeMap algorithm may produce tessellations of many different rectangle aspect ratios complicating the comparison of two nodes by the user.

Various enhancements of the original slice-and-dice algorithm exist which are improved with respect to the the aspect ratio. Several algorithms implemented by Wattenberg and Bederson, and an interactive comparison, are available online [77]. Schreck et al. evaluated the space efficiency of different TreeMap techniques and proposed the Grid TreeMap which provides a higher degree of regularity than the standard techniques [59]. CatTrees [40] are another extension of TreeMaps that allows the manipulation of the hierarchy itself. After an initial hierarchy is loaded the user can change the order of the hierarchical structure. This can be useful if the data set is not inherently structured and little is known about the data (explorative analysis, see Chapter 1.1.2).

Parallel Sets

Similar to Parallel Coordinates (see Chapter 2.1.2) and bargrams (Chapter 2.1.3) Parallel Sets use a layout where the data dimensions are represented by parallel axes. Whereas the continuous design model of Parallel Coordinates makes them unsuitable for categorical variables, Parallel Sets like bargrams have a discrete design model, based on the



Class: Class
Third: third class [706 (32%)]

Figure 2.10: Screenshot of Parallel Sets in action. Three dimensions, *Class*, *Sex* and *Survived*, from the titanic data set are visualized. The active dimension, in this example *Class*, is used to assign the colors to the connections. Image courtesy of Bendix[6].

frequencies of categories which is tailored towards categorical data. Each dimension is represented by its categories and is aligned horizontally. The numeric axes of Parallel Coordinates are replaced by proportionally scaled boxes. As with bargrams the relative frequencies of the corresponding categories are used to scale these boxes.

Furthermore the lines of Parallel Coordinates are replaced by parallelograms which connect each pair of categories of adjacent dimensions. In the worst case, Parallel Coordinates may represent all data items with a single line. Parallel Sets solve this problem by again using the relative frequencies to scale the connections and enable the user to compare the width of the parallelograms to see how many observations are represented by the combination of categories. One active dimension, which can be selected by the user, is used to assign the color-coding of the connections. Additionally, the active dimension defines the visual ordering and segmentation of connections into sub-connections. This process, also dependent on the ordering of the displayed dimensions, starts at the active dimension and connections of neighboring dimensions are split into sub-connections according to their number of categories. Figure 2.10 shows an example, where three dimensions of a data set are displayed and connections are split as well as colored according to one active dimension.

To obtain the information represented by the Parallel Sets visualization, a crossabulation is used (see Figure 2.11 for an example). As shown in Chapter 2.1.1 crosstabula-

Class	Sex					
	female		male			
first	145	44.6%	180	55.4%	325	
	30.8%	6.6%	10.4%	8.2%		14.8%
second	106	37.2%	179	62.8%	285	
	22.6%	4.8%	10.4%	8.1%		12.9%
third	196	27.8%	510	72.2%	706	
	41.7%	8.9%	29.5%	23.2%		32.1%
crew	23	2.6%	862	97.4%	885	
	4.9%	1.1%	49.8%	39.1%		40.2%
	470		1731		2201	
		21.4%		78.6%		100%

$\mathbf{f}_{ij}$ 145	44.6% $\mathbf{c}_{ij}$
$\mathbf{r}_{ij}$ 30.8%	6.6% $\mathbf{p}_{ij}$

Figure 2.11: Crosstabulation of two dimensions (class and sex). Image courtesy of Bendix et al. [7]

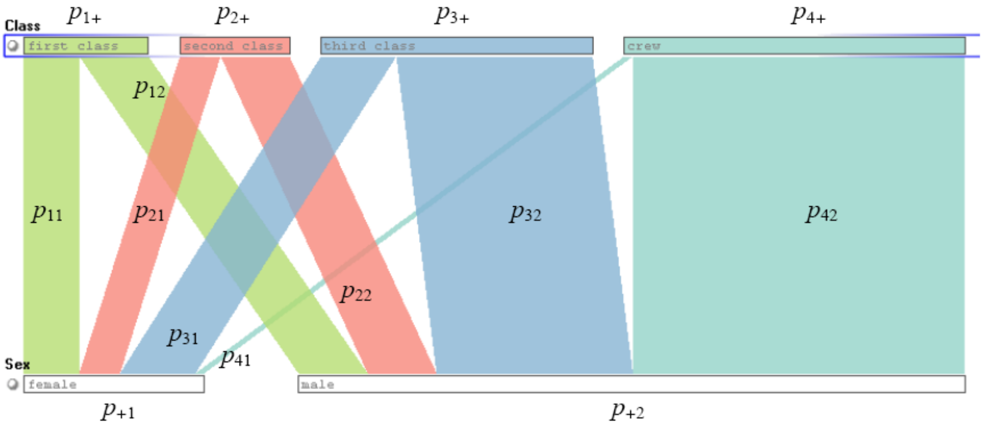


Figure 2.12: Using the information from the crosstabulation to scale the categories and connections. Image courtesy of Bendix [6].

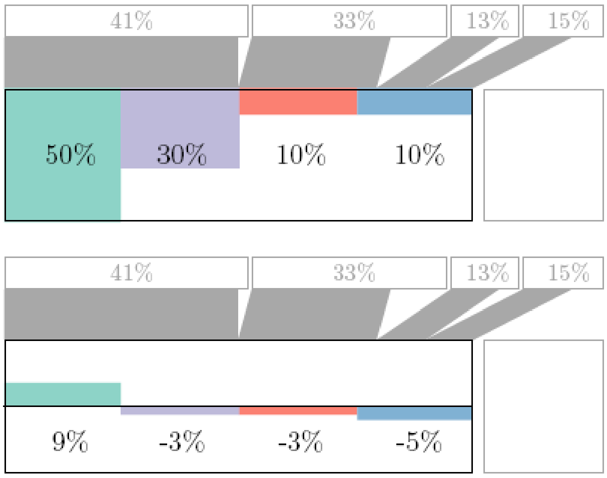


Figure 2.13: In the upper image histograms are used to display the conditional probabilities while the lower image shows the degree of independence. Images courtesy of Bendix et al. [7].

tion is a combination of frequency tables where each cell represents the combination of two variables. Crosstabulations are commonly used in statistics to identify relationships between variables.

For each attribute combination of the i-th row and the j-th column, the absolute frequency (item count) f_{ij}, the relative frequency $p_{ij} = f_{ij}/f_{++}$ ($f_{++} = \sum\sum f_{ij}$) and the individual row and column frequencies ($r_{ij} = f_{ij}/f_{i+}$ and $c_{ij} = f_{ij}/f_{+j}$ with $f_{i+} = \sum_j f_{ij}$, $f_{+j} = \sum_i f_{ij}$) are calculated. The marginal frequencies p_{i+} and p_{+j}, only affected by one dimension (i.e. they can be obtained for every dimension without using a crosstabulation), are used to scale the individual categories while the connections are scaled by the relative frequencies p_{ij}.

Obviously, Parallel Sets display all information which also crosstabulations delivers. But whereas the standard crosstabulation considers only two dimensions at once in a single table, Parallel Sets can display multiple dimensions and a user can examine relationships by comparing the colored and split-up connections.

While the marginal frequencies which can be seen as probabilities $P(A)$, the calculated frequencies of a crosstabulation are conditional probabilities $P(A|B)$. As proposed by Bendix et al. [7], the Parallel Sets visualization can be extended by histograms which are drawn inside each category box. These histograms among other display options can then be used to visualize the conditional probability or the ”degree of independence” which is the deviation of conditional to the unconditional probability:

$$doi(A, B) = P(A|B) - P(A)$$

The two categories A and B are independent, if $P(A|B)$ and $P(A)$ are equal. This

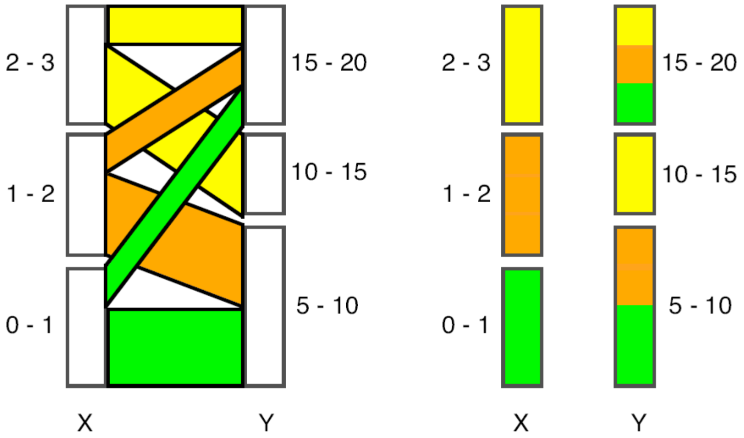


Figure 2.14: Comparison of Parallel Sets (left) and Parallel Trees (right) visualizing the same two dimensions. Image courtesy of Sifer et al. [62].

information has proven to be very valuable in finding relationships and patterns in the data set. See figure 2.13 for an example.

Parallel Sets are a highly interactive. Dimensions and categories may be rearranged as well as grouped together. Furthermore, dimension composition is an interactive approach to dimension reduction where the domain knowledge of the user is integrated in the process. Additionally, dimensions and categories can be highlighted revealing extra information (tooltip, association lines drawn in foreground).

Bendix et al. [7] also point out that Parallel Sets are not limited to categorical dimensions but also support continuous data which has been categorized using binning or clustering.

Parallel Trees

Parallel trees arrange and scale categories similar to Parallel Sets, but do not show any connections between the dimension. According to Sifer an increasing number of colored paths of varying thickness become difficult to read in case of many categories [62].

Instead, Parallel Trees implicitly link one active dimension with all others by coloring parts of the boxes (see Figure 2.15 for an example). Like the connections in Parallel Sets, the relative frequency p_{ij} (see Figure 2.11) is used to scale these colored parts. Figure 2.14 compares the Parallel Sets and Parallel Trees visualizations. Parallel Trees only show the relationship between the active dimension and all others rather than between adjacent dimensions (like Parallel Sets). Sifer feels that this reduced visual complexity provides a significant advantage if more than three dimensions are shown.

The main feature of Parallel Trees is the support for hierarchies of categorical data. As shown in Figure 2.15 multiple levels of a hierarchy can be shown and compared

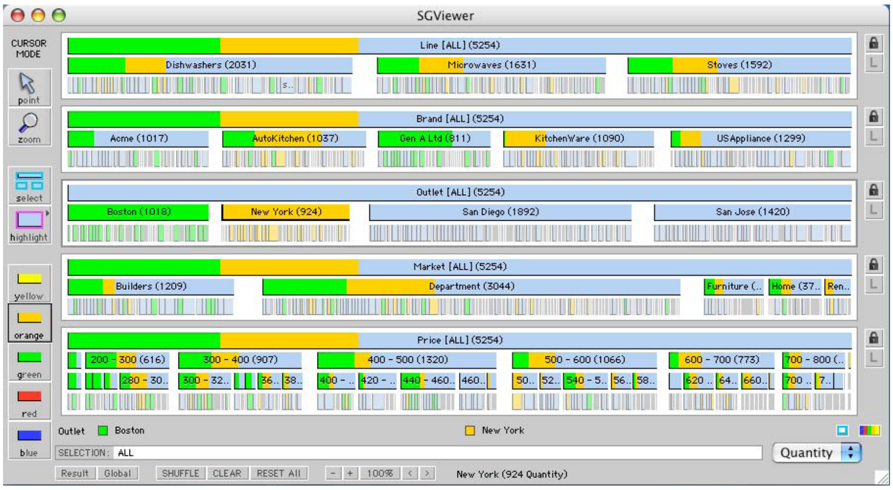


Figure 2.15: The categories ”Boston” and ”New York” are selected (green and yellow) coloring all other dimensions accordingly. Furthermore one can see the hierarchical structure of the dimensions. Image courtesy of Sifer [62].

simultaneously. Each hierarchy (also called Dimension Tree) is represented by a top level which aggregates all items, at least one intermediate level and a bottom level containing separate nodes for the highest possible detail level. Figure 2.15 shows a visualization of a sales data set, containing 365 orders for a total of 5254 items. The lowest level contains 365 nodes which width is proportional to its item quantity. All intermediate levels are scaled similarly.

In order to drill-down on data subsets or single categories, categories on any level can be selected, highlighted or filtered. All dimensions are then filtered accordingly, showing only a restricted view of the data, excluding currently irrelevant information. For example, a user could restrict a sales dataset to only show information about a certain product. Afterwards, additional categories (e.g. a specific month) can be selected and the remaining dimensions are filtered accordingly. Visual cues, such as the border color of a category, are used to show which dimensions are restricted. Though powerful, a student study by Sifer shows that users prefer coloring over filter selections to make comparisons [62]. He also states that it is possible to lose track of the query sequence when many dimensions are involved.

Interactive Sankey Diagrams

Traditional Sankey diagrams are static visualizations of dynamic processes. For example, a Sankey diagram can be used to visualize the flow of energy within a city. The diagrams display quantitative information about transport flows, their relationships as well as their transformation. Used since the 19th century, Sankey Diagrams represent weighted, directed graphs.

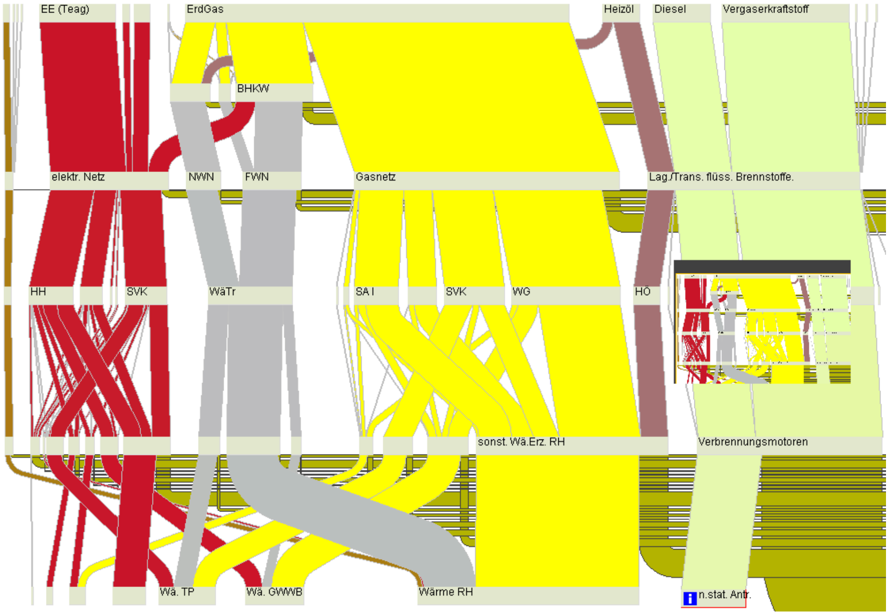


Figure 2.16: The energy distribution in a city visualized using an interactive Sankey diagram. Different types of energy are represented by the nodes, and the width of nodes as well as edges provides information about the quantitative flow of energy. Image courtesy of Riehman et al. [56]

Riehman et al. [56] describe a system which allows for an interactive analysis of such diagrams. The used visual metaphor is very similar to the one of Parallel Sets as scaled nodes are laid out on horizontally aligned parallel axes. Furthermore, the nodes can be grouped to form a hierarchy, which can be analyzed at different levels of detail. An important feature of the visualization is flow tracing. The user can select any node or edge, highlighting the contributions of all flows. Instead of simply highlighting all contributing edges, the authors suggest, that the actual contribution of each edge (i.e. quantitative information of the flow) is emphasized.

Using straight connections for the edges introduces the problem of non-constant line widths if the connections are not horizontal or vertical. Figure 2.16 shows how this can be solved by constructing curved edges using concentric circles and parallel lines. Besides that, the visualization sorts the edges and shows the tick edges, which usually represent the most important information, on top of thin edges. Alternatively, the sorting order can be switched by the user to avoid occluded edges.

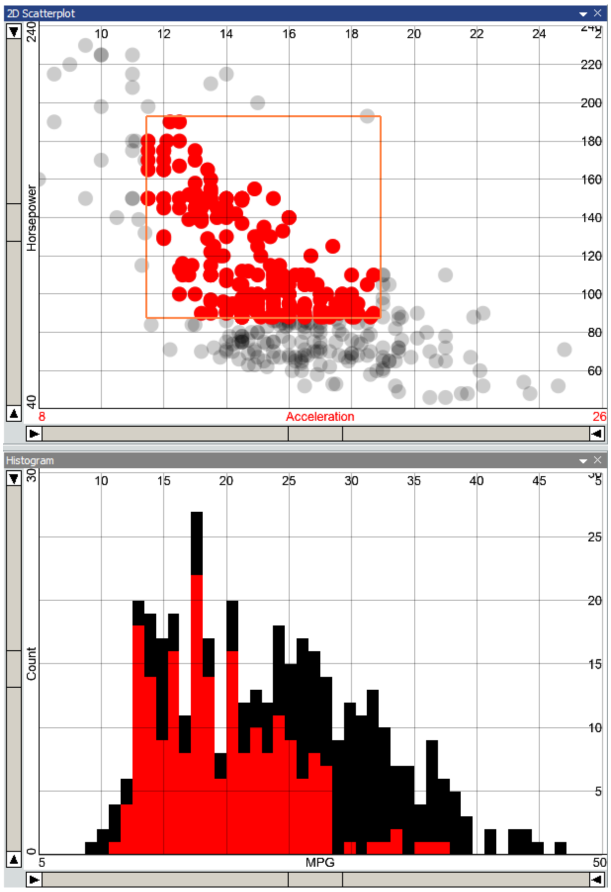


Figure 2.17: Brushing and linking in the BulkAnalyzer visualization system. A region of interest is brushed in the scatterplot (high horsepower, low acceleration) which is then linked to a histogram revealing the low mileage of those cars (red color).

2.2 Important Interaction Techniques

Interaction with the visualized data is key to the analysis process, and the recognition of relationships and patterns within the data. The following section describes some commonly used interaction techniques.

2.2.1 Brushing and View Linking

The idea behind linking and brushing is to combine different visualization views and techniques in order to overcome the issues of single views. Brushing can be employed by the user to interactively select regions or points of interest directly within views, where selected data items are then highlighted in all linked views [25]. Doleisch et al. point out that linking views in order to interactively update all changes in the different views

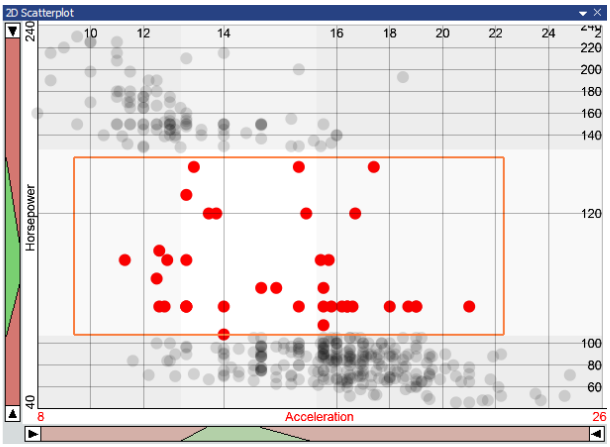


Figure 2.18: An example for Focus+Context visualization. The scatterplot view is zoomed revealing more information where needed.

simultaneously is crucial for any visualization system using multiple views [14].

Highlighting subsets can be achieved in many ways, a simple but effective way is to use the same coloring for selected points in all views (see Figure 2.17). Another solution is to use transparency where selected points are drawn opaque while the points outside the selected subset are half transparent. The interaction of brushing is highly dependent on the used view. For instance in a scatterplot, the user could brush multiple points by selecting a rectangular region. Categorical visualizations may give the user the option to select single categories by clicking on them.

Additionally brushes can often be modified later by either direct (using the mouse to alter the brush) or indirect manipulation (using separate widgets of the user interface to specify brush coverage). As one possible extension, smooth brushes allow for a non-binary transition between points inside and outside the selected subset. Brushes can also be combined by using logical operators like AND, OR and NOT making them a very powerful toolset in the visual data exploration process. A framework for flexible and interactive feature specification using smooth brushes is described by Doleisch et al. [14].

Structure based brushes are proposed by Fua et al. [19] which allow for brushing in hierarchies of data. The technique was implemented in the XmdvTool, a platform for visually exploring multidimensional data [75]. XmdvTool also introduced an n-dimensional brush where each of the brush's dimensions correspond to a data attribute (or range of values). Similar results can be achieved by combination of two- or one-dimensional brushes found in most other systems.

2.2.2 Focus and Context Approach

A major problem of large datasets is that it is hard to show the whole dataset simultaneously without losing information. In simple approaches the user has the choice between either viewing only an overview of the data or a zoomed visualization where specific details are revealed. Both, overview and details, are important in the visual data exploration (see the information seeking mantra in Chapter 1.1.2). Focus+Context techniques combine both types of information supporting the visualization of the entire data at once as well as specific details [10].

One of the first Focus+Context techniques was the fisheye strategy by Furnas [20]. This strategy enlarges the focus directly within the overview context, which is distorted to fit in the limited display space. Techniques using this strategy are also called distortion-oriented techniques. While most research work has been devoted to distortion techniques, Hauser points out that the idea of separating focus from context is more general [24]. The use of other visual dimensions, such as color or opacity, is suggested in addition to (or instead of) space distortion.

Interaction is very important for Focus+Context visualization since the user needs to be able to focus according to his or her goals. Therefore a Focus+Context system should give the user the ability to not only focus on specific regions of data, but also change those region interactively. Various approaches of focusing in different datasets are shown by Hauser [24]. In general a notion of which parts of the data are in the focus (and which are not) is required. This can either be a binary decision or a smooth one, as mentioned by Doleisch et al. [14].

2.3 Data Mining and OLAP

The combination of Information Visualization and data mining, often called visual data mining, is getting more and more important as data set sizes continue to increase. As Hauser and Kosara [25] point out, both approaches attain similar goals but apply different methods to do so.

2.3.1 Data Mining

Data mining or Knowledge-Discovery is the analytic process of finding patterns or relationships in usually very large data sets. The obtained models are typically validated by applying them to new subsets in order to predict future trends and behaviors. Thearling [71] defines data mining as *the extraction of hidden predictive information from large databases* and predictive data mining is an often used type with the goal of generating predictions [67].

Unlike to the visual data exploration techniques of InfoVis, data mining is an automated process usually involving little user interaction. Artificial neural networks,

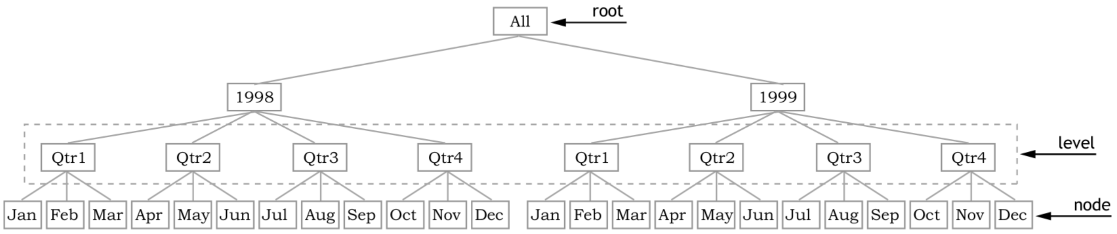


Figure 2.19: Hierarchical structure of the time dimension which consists of four levels: all, year, quarter, and month. Images courtesy of Stolte et al. [69].

decision trees and nearest neighbor methods are examples of common techniques in data mining.

Visual data mining is an approach to combine traditional data mining techniques and InfoVis visualizations to utilize the advantages of both, automated analysis methods and the human perception. Keim et al. [37] provide an extensive overview of visualization and interaction techniques, which include most of the methods described in the previous two sections as well as novel approaches to visualize data clusters, decision trees and text documents, to name a few. The work of Rongitsch [57] investigates the differences between data mining and InfoVis, also coming to the conclusion that the integration of both fields has to be the goal to enhance the data analysis process.

Association Rules The concept of association rule mining was introduced by Agrawal et at. [3]. Association rules provide predictions and patterns in the data set in the form of $X \rightarrow Y$ where X and Y are two disjunctive subsets of the data (”if-then” statements). An example for an association rule is the statement that 95% of all people living in Austria speak German fluently. Note that association rules are directed, e.g., not 95% of all fluent German speakers are living in Austria. Association rules can also be used with hierarchies, e.g., Germany, Austria and Switzerland can be combined in a single category. Certain constraints on measures of significance and interestingness have to be satisfied by the association rule (whether a rule is important enough to be worth consideration).

Drill-Down Analysis is another concept, which can also be applied to data mining [67]. Instead of the whole data set at once, only a few interesting variables are chosen and analyzed. The user can then *drill-down* on interesting subsets and adapt the data mining process to the gained insight. This more interactive approach has similarities to the explorative analysis in visualization (see Chapter 1.1.2) and is especially suitable for hierarchical structured data sets.

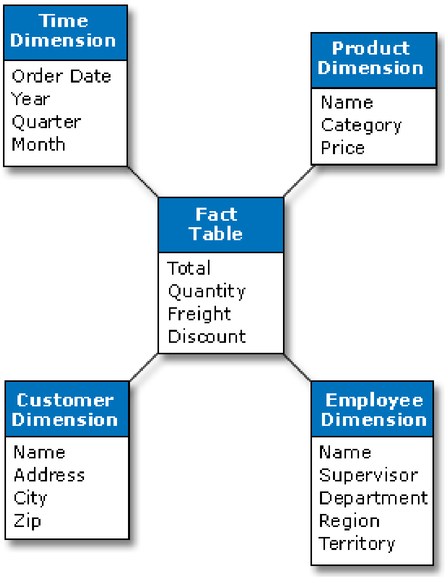


Figure 2.20: Star Data Model for OLAP. Images courtesy of Mailvaganam [44].

2.3.2 Online Analytical Processing

The term Online Analytical Processing (OLAP) was introduced by Codd in 1993 [13]. The goal of OLAP is to answer multi-dimensional analytical queries; typical applications are sales and financial reports. Rather than using a relational database, OLAP uses a multi-dimensional view of aggregate data. One could think of it as an n-dimensional spreadsheet. Multi-dimensionality is also one of the key requirements for any OLAP system because almost all business models are represented by at least four or five dimensions [49]. Additionally huge data sets with millions of stored transactions have to be supported and queries should be answered quickly whether a request is for the weekly sales of a single product or yearly sales across all products [62]. MDX is the most commonly used query language for OLAP systems and the output is typically in matrix form.

In general *OLAPs are designed to give an overview analysis of what has happened* [44].

Relational databases organize the data into relations (tables) which store multiple records (rows). OLAP data is often collected from relational data (which is used to process orders etc.) and organized in a star data model (Figure 2.20). The so called fact table in the center is surrounded by other tables, the dimensions. Each data attribute (such as product, consumer, employee and time period) is represented as a separate dimension. The fact table, on the other hand, stores numerical measurements (e.g. the total quantity of a product sold in a specific time period). While the fact table usually only has a small number of fields but is usually very long since it stores counts for all the possibly dimension combinations. The dimensions are often hierarchically

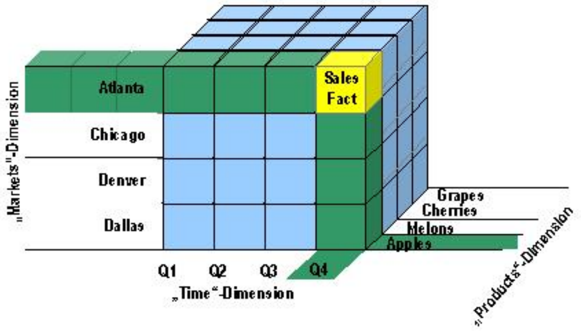


Figure 2.21: OLAP Cube with Time, Markets and Product Dimensions. Images courtesy of Mailvaganam [44].

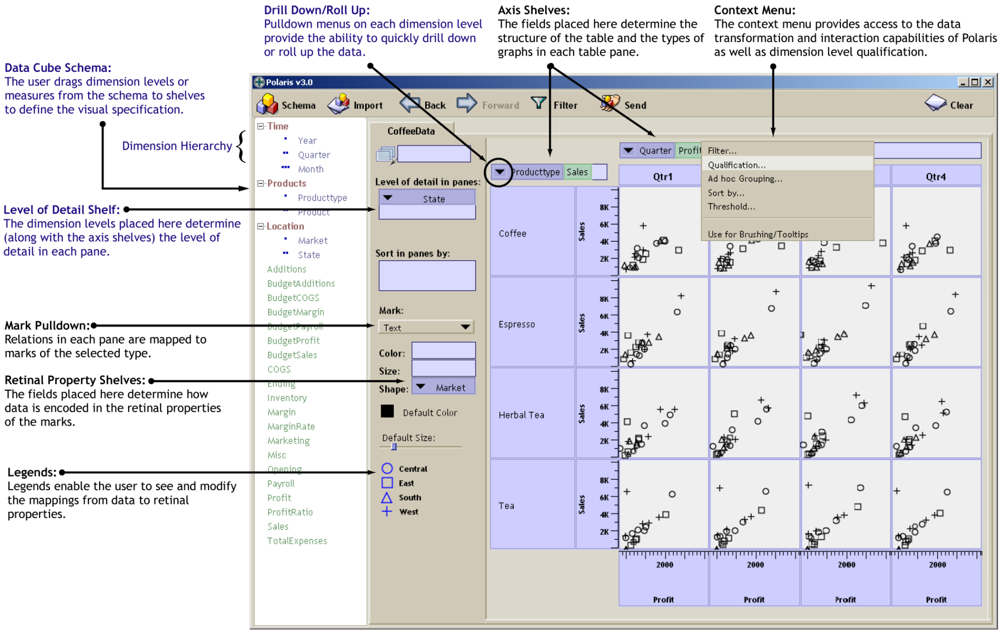
structured (Figure 2.19) and counts are aggregated in the fact table [69]. The major advantage of using this data model are the very fast response times to queries because measurements and counts do not need to be calculated when the data is queried, but are already available in the fact table.

The example from Figure 2.20 could be used to answer queries like "which employee sold the most units of a specific product in the last year". Using the star schema *OLAP cubes* (Figure 2.21) are created where the values of the fact table are stored in the cube's cells. The cubes can have any number of dimensions and are not restricted to three dimensions. OLAP operations can then be applied on the cube, reducing the whole data set to a subset by slice-and-dice, performing aggregations with drill-down and roll-up or rotating the cube to view different dimensions. The most common used interface for exploring OLAP cubes is the pivot table [62] (see Chapter 2.1.1).

OLAP and data mining have similar goals, but are conceptionally different, as OLAP stresses as user-driven, interactive approach, while data mining mostly operates automatically on the data. OLAP usually deals with aggregations in highly dimensional hierarchically structured data sets and is used to extract information on different granularity levels. Whereas data mining provides methods for finding patterns and relationships in data sets, OLAP does not. Both, the more interactive and explorative analysis of OLAP and the automated tools of data mining can be used in conjunction and complement each other well. For example association rules can be mined using OLAP cubes as Messaoud et al. propose [47].

2.3.3 OLAP and Information Visualization

As Vinnik and Mansmann state, the ultimate benefit of applying OLAP technology depends on the "intelligence" and usability of visual tools available to end-users [74]. Visualization is often used to help users understand the results generated by OLAP queries or data mining processes. Furthermore, InfoVis techniques can also be used to gain some initial insight into the data set before data mining is used on interesting


Drill Down/Roll Up: Pulldown menus on each dimension level provide the ability to quickly drill down or roll up the data.

Axis Shelves: The fields placed here determine the structure of the table and the types of graphs in each table pane.

Context Menu: The context menu provides access to the data transformation and interaction capabilities of Polaris as well as dimension level qualification.

Data Cube Schema: The user drags dimension levels or measures from the schema to shelves to define the visual specification.

Dimension Hierarchy

Level of Detail Shelf: The dimension levels placed here determine (along with the axis shelves) the level of detail in each pane.

Mark Pulldown: Relations in each pane are mapped to marks of the selected type.

Retinal Property Shelves: The fields placed here determine how data is encoded in the retinal properties of the marks.

Legends: Legends enable the user to see and modify the mappings from data to retinal properties.

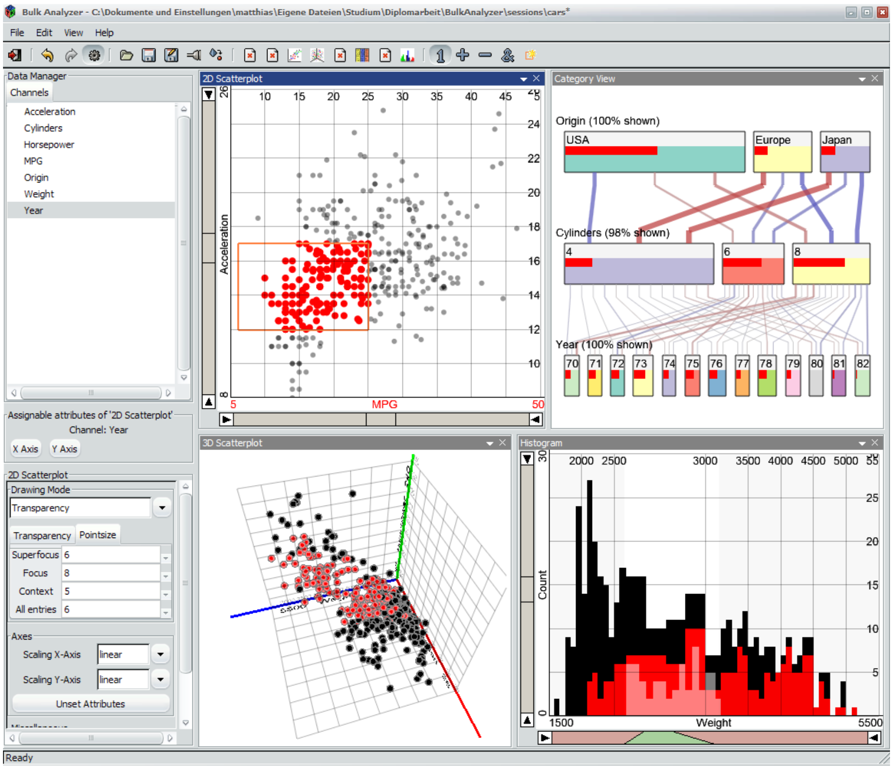
Figure 2.22: The Polaris user interface displaying a hierarchical data set. Image courtesy of Stolte et al. [69].

subsets. On the other hand, data mining or OLAP may be used as a first step to find interesting patterns or relationships which are further explored later or confirmed by the use of InfoVis methods.

Most research and commercial OLAP systems include visualization methods. One example is Polaris [68] and its commercial successor Tableau Software [70]. Polaris extends the interface concept of Pivot Tables and makes use of tabular layouts to display multiple different graphics and views at once. Polaris is built upon a formalism for describing table based visualizations which defines the mapping of data sources to layers, the mapping of dimensions to rows and columns and aggregation of data values. While these parameters affect the "outer layout", the formalism further describes how data within a single view cell is represented. This includes the type of graphical display, the mapping of values to graphical attributes (e.g. color) etc.

Table configurations can either be specified visually (see Figure 2.22 for an example of the Polaris interface) or by means of an XML specification. Using the interface, users can construct and refine visual queries which then update the visualizations. Polaris supports hierarchies, making it a suitable tool for visually exploring OLAP data cubes [69] and also allows the reordering of hierarchical structures. The available visualization techniques (scatterplots, bar charts, glyph based displays etc.) adapt automatically to display continuous or categorical data.

Another visualization system, ADVIZOR, implements a relational model but also allows the exploration of OLAP cubes. ADVIZOR supports hierarchical data and imple-



Bulk Analyzer - C:\Dokumente und Einstellungen\matthias\Eigene Dateien\Studium\Diplomarbeit\BulkAnalyzer\sessions\cars*

Assignable attributes of '2D Scatterplot'
Channel: Year

Drawing Mode
Transparency

Superfocus 6
Focus 8
Context 5
All entries 6

Scaling X-Axis linear
Scaling Y-Axis linear

Ready

Figure 2.23: An example screen of the Bulk Analyzer system displaying a data set using multiple linked views and brushing to highlight data across visualizations.

ments various different visualization techniques including bar charts, histograms, scatterplots or more specific views such as the time table or the data sheet. ADVIZOR and many other InfoVis systems are described and compared in a survey by Hansen [23]. In 2001 Kobsa [38] compared three multi-dimensional InfoVis systems (Eureka, Spotfire and InfoZoom) empirically and described advantages and disadvantages of the different approaches.

2.4 Bulk Analyzer

The Bulk Analyzer is a visual analytics software for high dimensional, large data sets which is currently in development at the VRVis Research Center [2]. Originally developed for the analysis of engine simulations, Bulk Analyzer supports various input formats and visualization techniques and may be, to a certain extent, seen as a general Information Visualization toolkit. The Bulk Analyzer system is based on several basic methodologies, which are applied to all parts of the system. Some of these methodologies, which are also important for the visualization of categorical data, are described in the following section.

2.4.1 Multiple Linked Views

The Bulk Analyzer system is a multi view approach (see Figure 2.23), providing the user with various different visualizations. A large number of views can be displayed simultaneously and views may be arranged on the screen by the user. All views are bidirectionally linked to each other over a feature specification language similar to the approaches described in Section 2.2.1. Every user interaction in one view immediately updates all other views, making the visual exploration and analysis of large, mostly inhomogeneous data sets possible. The interactive feature specification process (called brushing) is dependent on the specific view, enabling the view designer to choose adequate techniques for this purpose.

The Bulk Analyzer is furthermore based on a Focus+Context system (shown in Figure 2.18), which is directly connected to the interactive brushing. Using the provided methods, a user can brush, and therefore select, subsets of the data using one or multiple views. The selected subsets are then highlighted in all views and drawn in a more prominent way than the rest of the data. The Bulk Analyzer offers three of such subsets, called the *focus*, *context*, and *super focus* layers. Focus and context layers can be set similarly using combined brushing and are especially useful for comparisons of subsets of the data. It is important to note that focus and context layers are not disjunctive subsets but may overlap. Whereas the brushes of these two layers can be quite complex using the feature specification language, the super focus layer is selected simply by hovering the mouse pointer over interesting regions of the data (e.g., a bar of a histogram, a point of a 2D scatterplot) and can be used to quickly explore the data in multiple views. Another interesting aspect of this system is, that the structure of the focus and context layers, i.e., the set of brushes defining the subset, may be altered by the user at any time. Every brush is associated with a specific view and the user may choose to modify the brush interactively, even if the view used to create the brush at first is not shown.

Additionally, the Bulk Analyzer allows the user to zoom in on parts of the data and use distortion techniques in order to keep an overview of the data while showing a smaller subset in greater detail. Zoom and distortion is handled separately in every view, allowing the user to focus on different regions of interest in multiple views.

2.4.2 Categorical Data Support

The Bulk Analyzer system is built upon the *InfoVis Library*, which was designed to provide a common framework for storing structured data for InfoVis applications. Data is stored in table structures, similar to relational databases and the InfoVis Library offers a set of means to work with this data. The library supports various data types, including categorical data, which is handled differently than other types. More precisely, categories are subsets of the data, independent from the underlying data types or origin of the data. The framework furthermore allows the hierarchical structuring of categories which associates a hierarchy with one dimension of the data set. Details on this subject can be found in the next chapter.

Chapter 3

Data Model

In order to incorporate hierarchically structured categorical data into an Information Visualization tool like the *Bulk Analyzer* the data model has to support this kind of data. This chapter describes the underlying data model as well as operations working on the data model. But first, categorical and hierarchical data are defined.

3.1 Categorical Data

In current research, a list of characteristics is used to clarify the term categorical data [7, 43, 58]. First and foremost, a categorical variable is usually defined by a rather small number of possible values. Independent of the data type, whether it is non-numeric (e.g. String) or numeric (e.g. Integer), any data with a small number of possible values can be seen as categorical.

As mentioned in Section 1.2.2, categorical data may not have a natural ordering. Even numerical values do not necessarily imply an inherent ordering of the values, since they are often used to code data (e.g., ”male” = 1, ”female” = 2). Similarly, they need not have natural numeric differences and interpolation between two values may be impossible or meaningless.

On the other hand, some categorical data may have an ordering as well as numeric differences, which makes identifying reliable characteristics for categorical data difficult. In this thesis, categorical data is only characterized by the limited number of distinct values and no other properties are expected.

3.2 Hierarchical Structure

A hierarchy is a structured system of asymmetrical and acyclic relationships. Each entity of the system is subordinate to exactly one other entity of the hierarchy. The entities are ordered or ranked in a way where one of the related nodes is the *parent*

Continent	Country	Region	Population
Europe	Austria	Wien	1.668.737
Europe	Austria	Burgenland	278.215
Europe	Austria	Salzburg	529.033
Europe	Germany	Bayern	12.488.392
Europe	Germany	Sachsen	4.249.774
...	...	...	

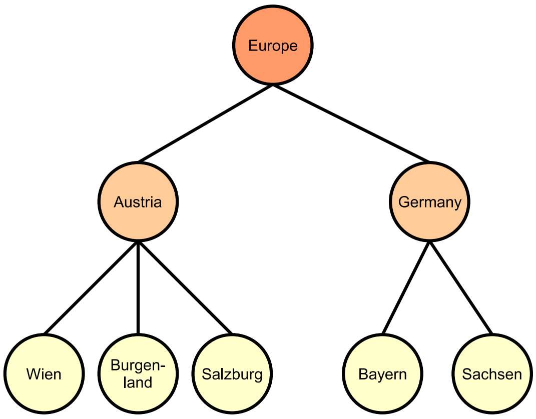


Figure 3.1: Hierarchical structure of a subset of the data from above, visualized as a tree.

and the other the *child*. Examples are relationships, such as ”B is part of A” or ”A contains B” where A would be the parent or also called ”superior” and B the child or ”subordinate”. Each parent in a hierarchy can have many children while a child can have only one parent, also known as one-to-many relationship or 1:N mapping. Special types of hierarchies may always have a fixed number of child nodes for each parent (e.g., full binary trees).

Multiple individual entities or nodes which have the same ”distance” to the root node, are called a *level*. Therefore a hierarchy can also be described as a system of nested levels in which one level can be appropriately regarded as nested within another level, introducing an ordering of levels. This allows a description in terms of higher and lower levels, where lower levels are nested within higher levels. In a system of 1:N mappings, the number of nodes generally increases with the distance of a level from the root node, resulting into fewer nodes at higher levels and more nodes at lower levels. Since higher levels contain the lower levels, hierarchies can be used to look at the same data set at different granularity levels. In some types of hierarchies, all leave nodes (i.e., nodes without any children) are at the same depth.

In graph theory, hierarchies are called trees. A tree is a connected acyclic graph and got its name because the traditional approach to draw the root node at the top of the page and the children below looks like an upside down tree. Trees can be ordered or unordered. Whereas nodes in an unordered tree arbitrarily placed below their parent node, the most common form are ordered trees where some order is imposed on the children of each node.

Hierarchies are often a natural way to structure data and can be found in a lot of different areas, e.g., evolutionary trees in biology, organizational structures in management, or directories in file systems. Furthermore, data which is not inherently structured can be put into a hierarchy by aggregating elements of lower levels. This is often used in financial applications and in OLAP, see Chapter 2.3.2 for details. Another example is shown in Figure 3.1. The data set, which contains population numbers of several different regions of the the world, consists of four dimensions, three categorical ones and one numeric attribute. The continent, country and region dimensions are hierarchically structured.

Hierarchies of categories can greatly enhance the understanding of a dataset by allowing the user to view the data at different granularity levels. This perfectly fits with the information seeking mantra (see Section 1.1.2)), *overview first, zoom and filter, then details-on-demand.*

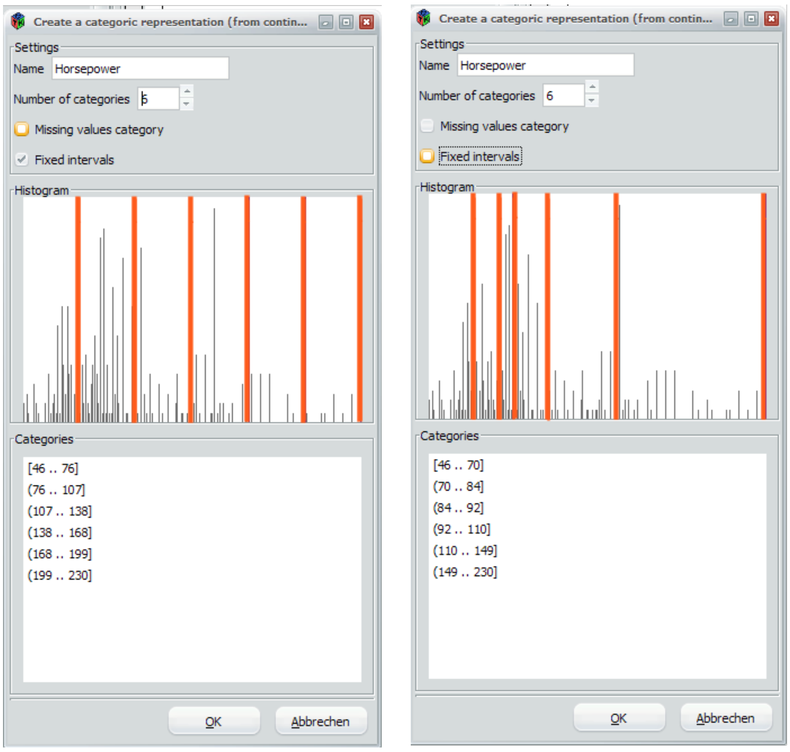
3.3 Categorical Hierarchies

After the term categorical data has been explained above, the subsequent section explains how categories can be identified in data sets. A description of how categorical data can be structured hierarchically follows and several operations on these hierarchies are presented.

3.3.1 Identifying Categories

Basically, categories can be identified in any data dimension but it depends on the data types as well as the number of unique values how this is reasonably achieved. First, the dimension is analyzed whether the data is inherently categoric or continuous. A dimension is considered inherently categorical, if the data type is "Boolean", or if the number of distinct values lies below a certain threshold, which depends on the data. Therefore, the choice of the threshold is left to the user, who may change the default setting (20 proved to be a reasonable number for most cases but may be to little for others, e.g., a data set containing categories for all 50 United States of America) depending on the task.

If the dimension is continuous, intervals are necessary to group (or bin) the data into categories. The intervals can be either uniformly spaced across the dimension's value range or automatically adjusted to contain equal frequencies. Additionally, the user can interactively change the number of intervals and their boundaries. To assist the user in the process of identifying categories, the values of one data dimension are sorted and displayed together with the interval boundaries in the form of a histogram. Figure 3.2 shows an example.


Create a categoric representation (from contin...
Settings
Histogram
Categories
OK
Abbrechen
Create a categoric representation (from contin...
Settings
Histogram
Categories
OK
Abbrechen

Name: Horsepower
Number of categories: 6
□ Missing values category
☑ Fixed intervals

[46 .. 76]
(76 .. 107]
(107 .. 138]
(138 .. 168]
(168 .. 199]
(199 .. 230]

Name: Horsepower
Number of categories: 6
□ Missing values category
□ Fixed intervals

[46 .. 70]
(70 .. 84]
(84 .. 92]
(92 .. 110]
(110 .. 149]
(149 .. 230]

(a) Uniform intervals. (b) Non-uniform intervals.

Figure 3.2: Comparing uniform and non-uniform intervals when creating categories from a continuous data dimension. The interval boundaries are highlighted in orange.

Missing Data: Data sets are often incomplete and dimensions are missing data entries, e.g., because a respondent did not answer a question of a survey. The handling of missing data is a challenging problem in visualization. One solution is to not display any data items with missing entries at all. While simple to implement, this also means, that potentially interesting data is not visible to the user because of a missing entry (e.g., in a data set with 100 dimensions, only one data value may be missing, possibly invalidating all 99 others). Therefore, the proposed approach allows the creation of a category containing all missing values. This solution, which works for both, inherently categoric and continuous data dimensions, enables the user to work with the missing data as it can be displayed and handled similar to valid data categories.

3.3.2 Building Hierarchies

After identifying suitable categories for a data dimension, a hierarchy containing the categories is created. Initially, the hierarchy is trivial, consisting of one root node (the name of the hierarchy, often the name of the data dimension) and the categories as its child nodes. The categories are disjunctive subsets of the data dimension, no data entry is part of more than one category in this hierarchy. Together, the categories represent the entire data dimension.

The initial hierarchy can be modified to introduce a stronger classification (specialization) or a less precise classification (generalization) by extending the hierarchy with additional levels. Both concepts, generalization and specialization, are important mechanisms to follow the ”overview first, details on demand” approach [61] and enable the user to view the data from varying degrees of detail (see Section 3.3.4).

In any hierarchy, all categories of each hierarchy level are disjunctive subsets of the data. Therefore, each hierarchy level represents the same data, but with a different classification.

Combining Categories

The combination of categories is a simple way to explicitly specify connections between categories, which are somehow (e.g. semantically) related. As shown in Figure 3.3 by combining the two categories A and B, a new parent category A+B is introduced which functions as a parent node for both A and B. While the degree of detail on the first level of the hierarchy has been reduced, the overall information of the hierarchy has been increased by extending its structure based on some meta-information about the categories.

Since the initial hierarchies only have one level of categories after they have been created, combinations can therefore be very useful to introduce additional levels of detail. An example is a country dimensions which stores each country separately. Without any other dimension or information, the user can combine the countries into regions or continents. Currently, combining categories is a manual process but may be adapted to an automated approach in the future. For example, single weeks of a time dimension could automatically be combined into months, months into quarters, quarters into years and so on.

Refining Hierarchies

While a combination introduces a more general category derived from specific ones, a refinement splits a general category and creates more specific ones below. This is done by using a second hierarchy which then refines each leaf-node of the first one, adding an additional level of detail to the hierarchy. Figure 3.4 shows an example, where ”Hierarchy 1” gets refined by ”Hierarchy 2”. While the first level of the hierarchy remains unchanged, an more specific level is added to the structure by adding categories below each former leaf-node. As one can see, refining is not commutative and the result differs whether ”Hierarchy 1” gets refined by ”Hierarchy 2” or vice versa (though the resulting leaf-nodes will be equal).

As different hierarchies may be derived from different dimensions, a refinement step thus means an integration of information from multiple dimensions into a single hier-

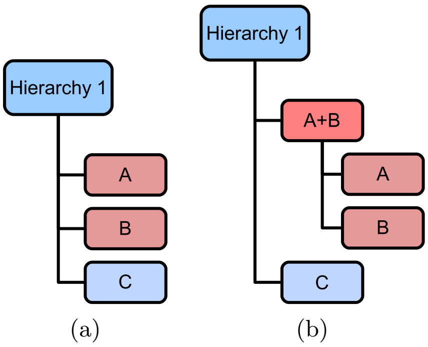


Figure 3.3: The left picture shows the hierarchy before the combination, the right one afterwards.

archy. Therefore refining can also be considered a kind of dimension reduction. For example, a user could refine a dimension ”sex” with ”marriage status” if the relationship of ”married man” and ”married women” in regard to other dimensions should be measured.

Hierarchies can be refined by an arbitrary number of other hierarchies and it is also possible to only refine parts of the hierarchy (single leaf-nodes or sub-hierarchies). Furthermore, a hierarchy can be ”unrefined”, which restores the original structure.

3.3.3 Other Operations

Move: In addition to combining and refining, the structure of a hierarchy can also be changed by moving categories (and their attached child categories) to different parent nodes. To maintain the criterion of disjunctive subsets, old and new parent node have to be at the same hierarchy level. Consequently, moved categories stay at the same level too.

Reorder: Each node imposes an order of its children. Therefore, reordering allows to specify a meaningful order in cases where such an order of the categories exists (e.g. intervals, days of the week, etc.). Every level and sub-hierarchy can be ordered individually.

Delete: Because a hierarchy may represent only a subset of the data, any category can be deleted. Even though this means that not the whole data set is represented any more, the criterion of disjunctive subsets is still fulfilled. Removing a category will further delete all sub-categories if it has any.

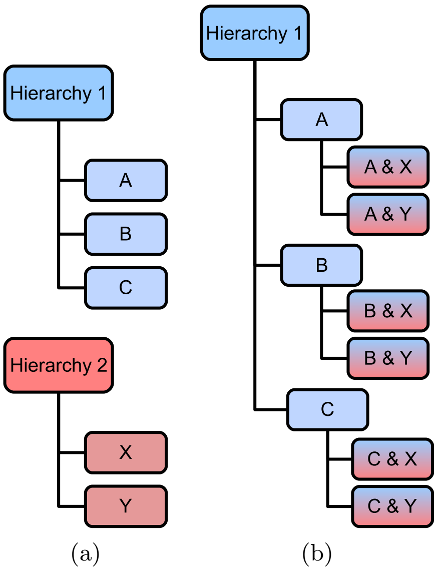


Figure 3.4: The left picture shows the hierarchy before the refinement, the right one afterward.

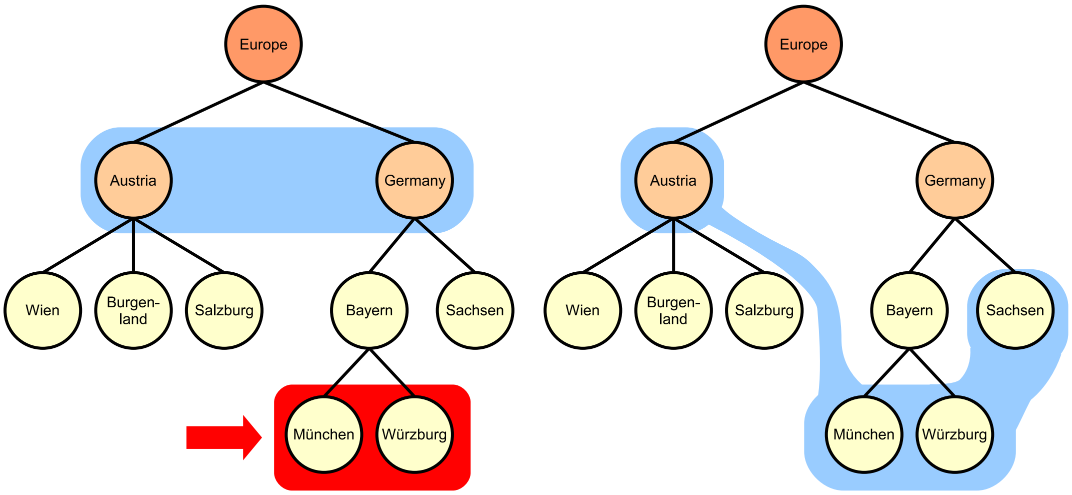
Rename: The application tries to find suitable names for categories (e.g. the interval boundaries) but especially after the structure of the hierarchy has been changed by the above described methods, it may be helpful to rename certain categories.

3.3.4 Navigation with Hierarchies

It is necessary to distinguish between modifications as mentioned above, and the way, how hierarchies and categories are locally used by individual views, most importantly the navigation within hierarchies.

In order to examine the data at different levels of abstraction, the user must have the ability to move down the hierarchy (viewing data with increased detail) and up the hierarchy (viewing data with decreased detail). These two operations, *drill-down* and *roll-up*, are the two basic navigational functions used in hierarchies with multiple levels of aggregations.

By using selective drill-down and roll-up operations, the user sets the *cut*, which is a disjunctive and complete subset of nodes of the hierarchy (i.e. it refers to all the data contained within the hierarchy itself). The purpose of the cut is to represent a certain state of navigation with respect to the level of detail and is used to visualize the data at different levels of abstraction. Instead of simply choosing a single level of the hierarchy, the cut can consist of categories in various levels, providing detailed information as well as context where needed.



(a) (b)

Figure 3.5: The left picture shows the initial state of the hierarchy with the top level in the cut (blue colored). The user then selects the category "München" to be represented in the cut, which adopts the cut accordingly and still maintains a disjunctive subset of nodes.

Drill-down and roll-up are possible directly within the visualizations (see the next chapters) which then automatically update the displayed data. These operations operate on nodes currently in the cut (e.g. a drill-down on a node will put all its children in the cut), but may also be used to modify the cut to include certain categories, which are currently outside the cut, as shown in Figure 3.5.

Chapter 4

Parallel Hierarchies

In this Chapter, a new visualization technique for hierarchically structured categorical data is proposed. The approach, which uses a parallel axis layout, similar to Parallel Sets and Parallel Trees, is able to display multiple categorical dimensions at once. Extending the idea of Parallel Sets, Parallel Hierarchies emphasize the visualization relationships between categories and allow for interactive brushing, as they are designed to be easily integrated within the concept of a system for visual data analysis.

Before describing the statistical background of the visualization, the motivation for the Parallel Sets approach is going to be explained. Afterwards, the Parallel Hierarchies visualization and its features are described in detail.

4.1 Motivation

Data sets as resulting from surveys or financial applications (see OLAP in Chapter 2.3.2) are often high-dimensional and contain a large number of categorical dimensions. The "Identity II" survey, for example, carried out by the International Social Survey Programme [53] contains 44170 entries (respondents) and 241 dimensions (questions asked) of which almost all are categorical. In order to analyze such data sets, traditional visualization approaches are often not sufficient since they are usually not designed to handle categorical data. On the other hand, solutions designed for categorical data normally do not visualize continuous data as well as techniques catered toward this type of data. Therefore, one of the main motivations of Parallel Hierarchies was to integrate them into an existing visual data analysis system based on multiple linked views. Using brushing across views, the user is able to take advantage of different visualization techniques.

As explained in Section 3.2, categorical data is often inherently structured or may be structured by the user to find the right level of aggregation for a certain task. Considering this hierarchical structure, the visualization facilitates an appropriate analysis of the data.

Shortcomings of Existing Techniques Parallel Sets as described in Section 2.1.3 are a visualization geared towards categorical dimensions, which is motivated by Parallel Coordinates, and works very well with multiple categorical dimensions. Parallel Trees are a similar approach, but also support hierarchies. Current implementations of both techniques are based on a single view and not integrated in a visual data analysis system.

Parallel Sets and Parallel Trees use the frequencies of categories to scale their visual representations (in both cases rectangles) and the categorical dimensions are aligned horizontally. Whereas Parallel Sets use parallelograms to connect categories of adjacent dimensions to visualize the data relations between categories, Parallel Trees do not show any connections between categories or dimensions. Instead, categories can be brushed to reveal the distribution of data in other categories. Sifer et al. [62] note, that the connections of Parallel Sets are very hard to read when many categories are displayed. As a matter of fact, the smaller the parallelograms are, the harder comparisons are. Parallel Trees try to solve this problems by displaying a hierarchical structure and different levels of aggregations can be used to compare subsets of the data.

Parallel Sets and Parallel Trees highlight the relations between different categorical attributes of the data by using the item count of categories. Called *support* in statistics [22]), this emphasizes items which occur frequently in the data set (i.e., categories with a high item count), but important patterns between smaller subsets of the day may be missed completely. Called the *rare item problem* [22], this disadvantage is very relevant in data sets with uneven distribution for individual items (e.g. financial transactions where a few items are used very often while most others are rarely used). Therefore, other measurements such as the *degree of independence* (see the histograms in Parallel Sets [7]) are often more interesting for analysts.

These shortcomings led to the major goals of the Parallel Hierarchies approach: the visualization has to be able to display a large number of categories as well as hierarchies, and simplify the pattern finding process by highlighting interesting information about relationships.

4.2 Statistical Background

The most common technique of analyzing categorical data statistically is a crosstabulation or short *crosstab*. Crosstabs provide information about combinations of categories of multiple (usually two) dimensions and are also used as the statistical background of Parallel Hierarchies (see Section 2.1.1 for details). An example of a crosstab with two dimensions is given in Table 4.1.

The first information of the crosstab is the distribution of categories in regard to their own dimension. For example, it shows that 1325 or 32.5% of the 4077 people that were asked are very proud of their country's achievements in sports (absolute and relative frequencies of the first column in 4.1a). Furthermore, 939 or 23% (frequencies of the fourth row) of the people are living in Austria.

Each cell gives information about the combination of crosstabulated values. The absolute frequency or count is the number describing how many observations fall into a specific combination (e.g., 463 Austrians say that they are very proud). Relative frequencies are absolute counts normalized by the total number of occurrences (463 very proud Austrians are 11.4% of the total, 4077). The relative frequency of an event (in this case a subset of the data) is also a probability estimate of the event and may be used to predict future occurrences. The quality of this estimation depends on the number of entries. In the following sections probability and relative frequency are going to be used interchangeable.

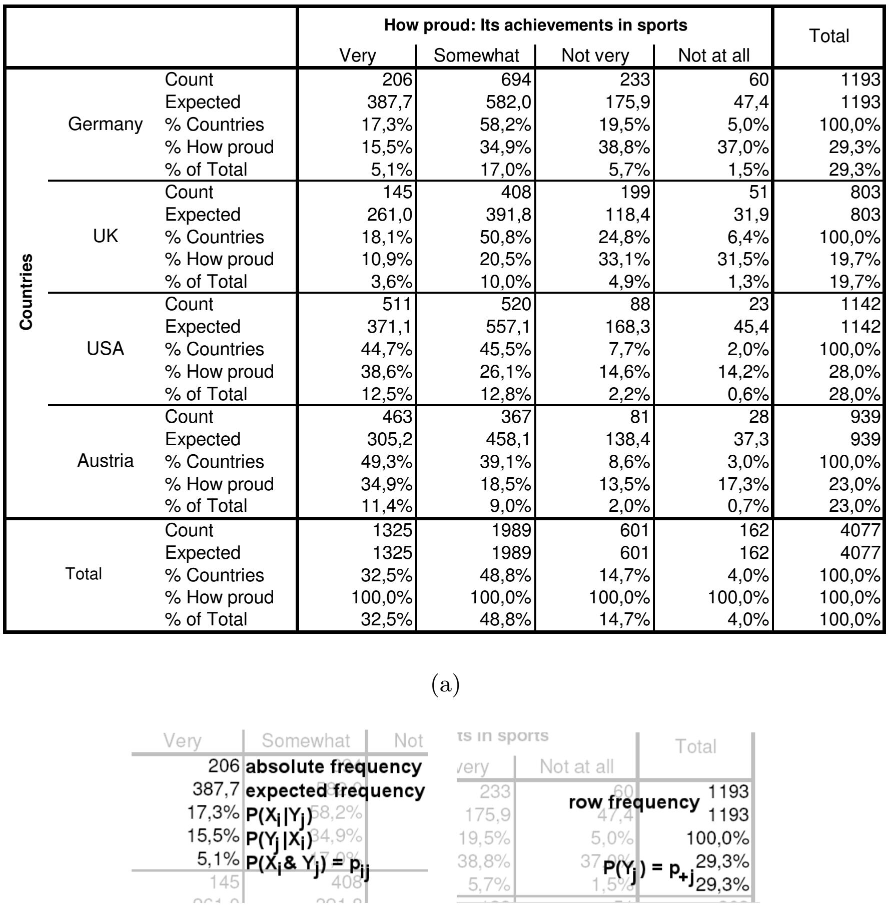
The expected count measures how many occurrences are expected by taking the distributions within the dimensions into account. Given 939 Austrians (23% of the total) and 1325 very proud respondents (32.5%), 305.2 Austrians are expected (32.5% of 939, 23% of 1325) to be very proud.

Using conditional probabilities (or conditional relative frequencies), the relative frequency of one category combination with regard to one category (rows or columns) can be seen. For example, 49.3% of the Austrians are very proud (463 of 939) and 34.9% of the total amount of very proud people are from Austria.

One very valuable piece of information in the pattern finding process is the discrepancy of expected and actual frequency. One can see in the above example, that Austrians are proportionally more proud of achievements in sports. Similarly, the conditional probabilities can be compared to individual category probabilities (e.g. 49.3% of the Austrians are very proud while only 32.5% of the total are very proud).

The classifications for which crosstabs can be used are not limited to the categories of an attribute of the source data, e.g. in a system allowing for interactive selection, the information "is selected/is not selected" is a highly relevant classification and including it in the crosstab is a powerful way of characterizing the selection by statistical means.

The results of a crosstab can simply be aggregated in a hierarchy. As a result, selective drill-down and roll-up operation only affect a subtree and the entire hierarchy.



			How proud: Its achievements in sports				Total
			Very	Somewhat	Not very	Not at all	
Countries	Germany	Count	206	694	233	60	1193
		Expected	387,7	582,0	175,9	47,4	1193
		% Countries	17,3%	58,2%	19,5%	5,0%	100,0%
		% How proud	15,5%	34,9%	38,8%	37,0%	29,3%
		% of Total	5,1%	17,0%	5,7%	1,5%	29,3%
	UK	Count	145	408	199	51	803
		Expected	261,0	391,8	118,4	31,9	803
		% Countries	18,1%	50,8%	24,8%	6,4%	100,0%
		% How proud	10,9%	20,5%	33,1%	31,5%	19,7%
		% of Total	3,6%	10,0%	4,9%	1,3%	19,7%
	USA	Count	511	520	88	23	1142
		Expected	371,1	557,1	168,3	45,4	1142
		% Countries	44,7%	45,5%	7,7%	2,0%	100,0%
		% How proud	38,6%	26,1%	14,6%	14,2%	28,0%
		% of Total	12,5%	12,8%	2,2%	0,6%	28,0%
	Austria	Count	463	367	81	28	939
		Expected	305,2	458,1	138,4	37,3	939
		% Countries	49,3%	39,1%	8,6%	3,0%	100,0%
		% How proud	34,9%	18,5%	13,5%	17,3%	23,0%
		% of Total	11,4%	9,0%	2,0%	0,7%	23,0%
Total		Count	1325	1989	601	162	4077
		Expected	1325	1989	601	162	4077
		% Countries	32,5%	48,8%	14,7%	4,0%	100,0%
		% How proud	100,0%	100,0%	100,0%	100,0%	100,0%
		% of Total	32,5%	48,8%	14,7%	4,0%	100,0%

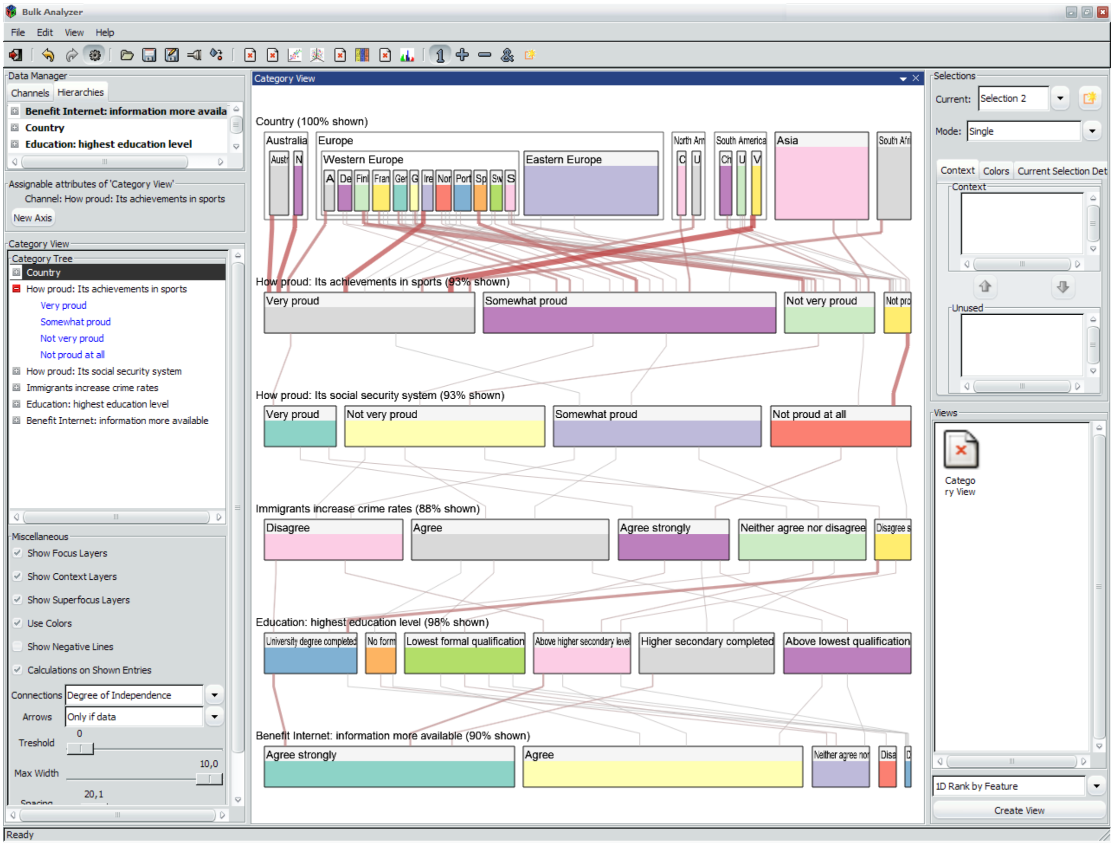
(a)

(b) (c)

Table 4.1: The table a) shows a crosstab as produced by statistical software like SPSS [66]. b) and c) explain the different values. See below for details.

Given the two dimensions X = "How proud ..." and Y = "Countries", each cell of the crosstab provides the following information:

- Absolute frequency f_{ij} ("count"): the actual item count of the two combined categories.
- Expected absolute frequency e_{ij} ("expected"): calculated by $f_{+j} * p_{i+}$ or $f_{i+} * p_{+j}$ (where f_{i+} is the marginal row count calculated by $\sum_{j=1}^{n} f_{ij}$, p_{+i} the row frequency, f_{+j} the marginal column count and p_{+i} the column frequency).
- $P(X_i|Y_i)$ or $confidence(Y_i \rightarrow X_i)$ ("% Countries"): the conditional probability of X_i under the condition Y_i, calculated by $P(X_i \cap Y_i)/P(Y_i)$.
- $P(Y_i|X_i)$ or $confidence(X_i \rightarrow Y_i)$ ("% How proud"): the conditional probability of Y_i under the condition X_i, calculated by $P(X_i \cap Y_i)/P(X)$.
- Relative frequency p_{ij} or $P(X_i \cap Y_i)$ ("% of Total"): The quantum of the two combined categories in relation to the overall frequency.


Bulk Analyzer
File Edit View Help
Data Manager
Channels Hierarchies
Benefit Internet: information more availa
Country
Education: highest education level
Assignable attributes of 'Category View'
Channel: How proud: Its achievements in sports
New Axis
Category View
Category Tree
Country
How proud: Its achievements in sports
Very proud
Somewhat proud
Not very proud
Not proud at all
How proud: Its social security system
Immigrants increase crime rates
Education: highest education level
Benefit Internet: information more available
Miscellaneous
Category View
Country (100% shown)
Australia
Europe
Western Europe
Eastern Europe
North Am
South America
Asia
South Afri
How proud: Its achievements in sports (93% shown)
Very proud
Somewhat proud
Not very proud
Not pro
How proud: Its social security system (93% shown)
Very proud
Not very proud
Somewhat proud
Not proud at all
Immigrants increase crime rates (88% shown)
Disagree
Agree
Agree strongly
Neither agree nor disagree
Disagree s
Education: highest education level (98% shown)
University degree completed
No form
Lowest formal qualification
Above higher secondary level
Higher secondary completed
Above lowest qualification
Benefit Internet: information more available (90% shown)
Agree strongly
Agree
Neither agree nor
Disa
Selections
Context Colors Current Selection Det
Context
Unused
Views
Category View
Create View
Ready

Current: Selection 2

Mode: Single

☑ Show Focus Layers

☑ Show Context Layers

☑ Show Superfocus Layers

☑ Use Colors

□ Show Negative Lines

☑ Calculations on Shown Entries

Connections: Degree of Independence

Arrows: Only if data

Treshold: 0

Max Width: 10,0

Spacing: 20,1

1D Rank by Feature

Figure 4.1: The screenshot shows the Bulk Analyzer system using the Parallel Hierarchies to visualize six hierarchies of categorical dimensions of the ISSP [53] survey data set. *Association lines* between adjacent dimensions are graphical representations for statistical measurements to identify relationships, as explained below.

4.3 Visualization

Figure 4.1 shows a screenshot displaying a typical setup of the Parallel Hierarchies visualization in the *Bulk Analyzer* system, where many featurs are turned on. The various features will be explained in detail in the subsequent sections (initially, no hierarchies are assigned to the view and the screen is white). An arbitrary number of dimensions can be visualized and arranged in any order. The layout of the hierarchies is similar to Parallel Sets and a horizontal alignment is chosen because of the horizontal alignment of labels, which are crucial to associate the graphical representation with a category. Furthermore, a larger number of categories may be displayed since the available screen space is better used (given that most resolutions display more information horizontally than vertically).

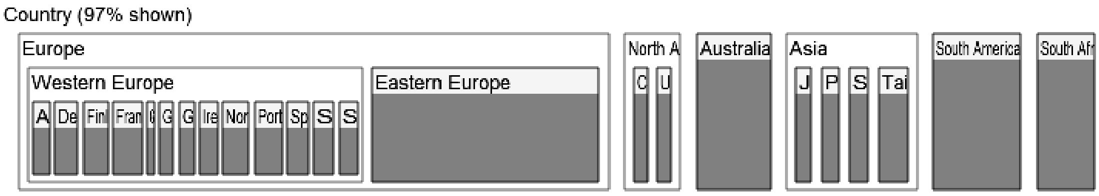


Figure 4.2: Hierarchical structures are displayed using nested boxes. The currently selected cut is visualized, while parent nodes are displayed by borders and labels.

4.3.1 Categories and Hierarchies

Categories are represented by adjacent boxes and relative frequencies are used to determine their widths, where the full width of the view (not including margins) means 100% of the shown data. Categories with higher item counts are therefore wider than categories with smaller counts. Thus, the sizes of the boxes is an immediate clue regarding the distribution and structure of one hierarchy. If categories are hidden, the shown categories still fill the entire width of the visualization, in order to allow for focusing on important categories. Because differently sized hierarchies can be misleading when being compared, a number displayed next to the dimension label shows the percentage of the displayed data in respect to the entire data set.

The concept navigating a hierarchy based on selective drill-down and roll-up operations was described in Chapter 3.3.4. Initially the cut is placed at the highest level of the hierarchy and the categories of this level are visualized as explained before. Drilling-down on a category with sub-categories modifies the cut and the visualization is updated. In order to represent the hierarchical structure, the visualization of the sub-categories are recursively nested within the box of their common parent node, which is still displayed (see Figure 4.2) by the bounding rectangle.

With an increasing number of hierarchical levels being displayed, the height of the dimension increases too. The lowest level (e.g. the individual Western European states in the above example) is displayed at a fixed height and the height of the other boxes are scaled accordingly. The horizontal spacing between categories decreases with any level to facilitate the visual discrimination of the various levels.

4.3.2 Layers

As described in Section 2.4, the *Bulk Analyzer* system allows the brushing of three independent subsets or *layers* of the data. The data in these layers is highlighted in all views, allowing the user to efficiently compare subsets of the data across views. In order to passively integrate the layers in the Parallel Hierarchies visualization, individual bars are displayed within each category box of the cut (this information is not shown for nodes above the cut). The layers have global colors assigned, making them distinguishable across visualizations. Similar to the width of each box, the bars are scaled according

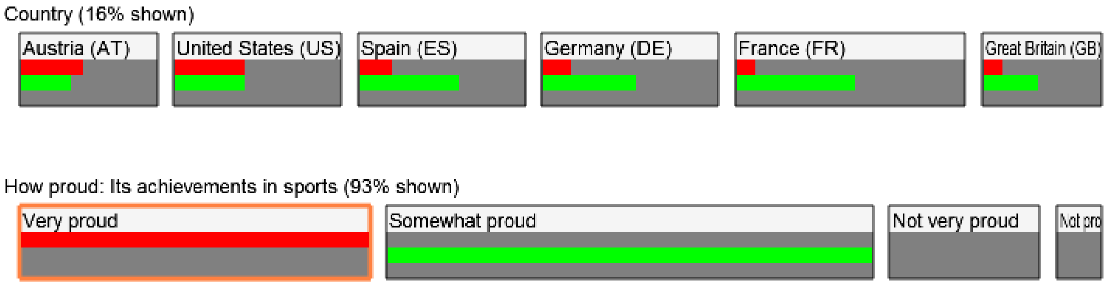


Figure 4.3: Visualizing the dimensions *country* and *how proud: achievements in sports?* reveals that most people are somewhat proud, while a lot of people from Austria and the USA are very proud.

to the relative frequencies of the entries being selected by the layer with respect to the according category (i.e., if the item count of one category is 200 and 100 of them are in one layer, the associated bar will fill half the width of the category). As information is shared between all views, layers link the Parallel Hierarchies visualization with other views and are also very powerful way when multiple dimensions are explored at once.

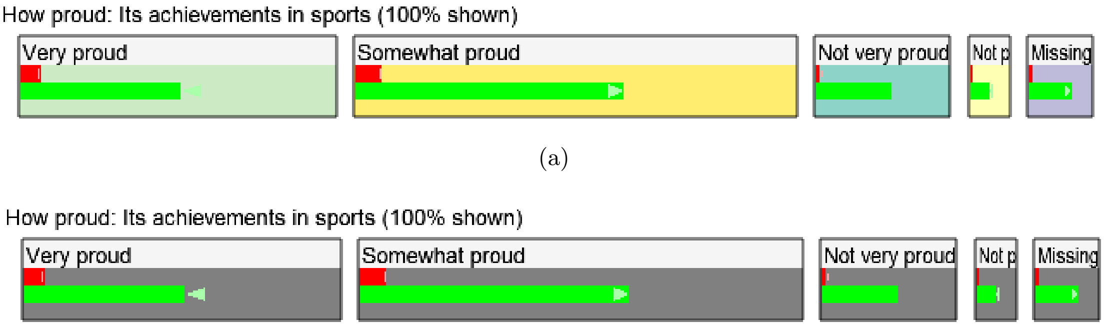
Using the Parallel Hierarchies approach, the crosstab example from before can now be visualized, as seen in Figure 4.3. A few things can be immediately seen from the graphical representations. First, the interviewees seem to be quite equally distributed over the countries, though France stands out with almost twice as many respondents as Great Britain. Furthermore, most people stated, that they are either "very proud" or "somewhat proud" of their country's achievements in sports. Only a small number are not "very proud" and people being "not proud at all" are the minority. This gives the user a basic overview of the distribution of categories within the single dimensions.

Categories can be selected by left-clicking, which was used to select the categories "very proud" (red), "somewhat proud" (green) and "not proud at all" (blue), revealing the distribution of these categories within the country categories. Details about brushing can be found in Section 4.4.3.

The percentage of Austrians and US Americans being very proud of sports achievements seems to be considerably higher than of the other countries. There are no such significant differences in the category "somewhat proud", but it can be seen that Austria is the only country where the amount of respondents being "very proud" is larger than "somewhat proud". The category "not proud at all" is quite small, which makes the analysis of its distribution in other categories hard.

4.3.3 Colors

Disabled by default, a qualitative color scheme of different colors may be used to color the categories to make the differentiation of categories more visually appealing. The scheme uses colors of which the hue is as dissimilar as possible but have similar brightness and


How proud: Its achievements in sports (100% shown)
Very proud
Somewhat proud
Not very proud
Not p
Missing
How proud: Its achievements in sports (100% shown)
Very proud
Somewhat proud
Not very proud
Not p
Missing

(a)

(b)

Figure 4.4: The same dimension, colored in the top picture and without colors below. While the colors make the appearance more visually appealing, they could also be misleading in that they implicate unwanted associations and make the colored layers bars harder to read.

luminance [8]. Different hues for successive categories help to prevent wrong implications of magnitude differences, but the use of colors may still be misleading for some users since each hierarchy uses the same color scheme (i.e, categories in different hierarchies use identical colors). Experiments have shown, that users often assumes a relationship between similarly colored categories even though there may be no notable relationship between them. Additionally, the colors may implicate unwanted semantical associations (e.g., blue = cold). Besides that, the colored layer bars are usually easier to see and compare using gray background instead of colors (see Figure 4.4).

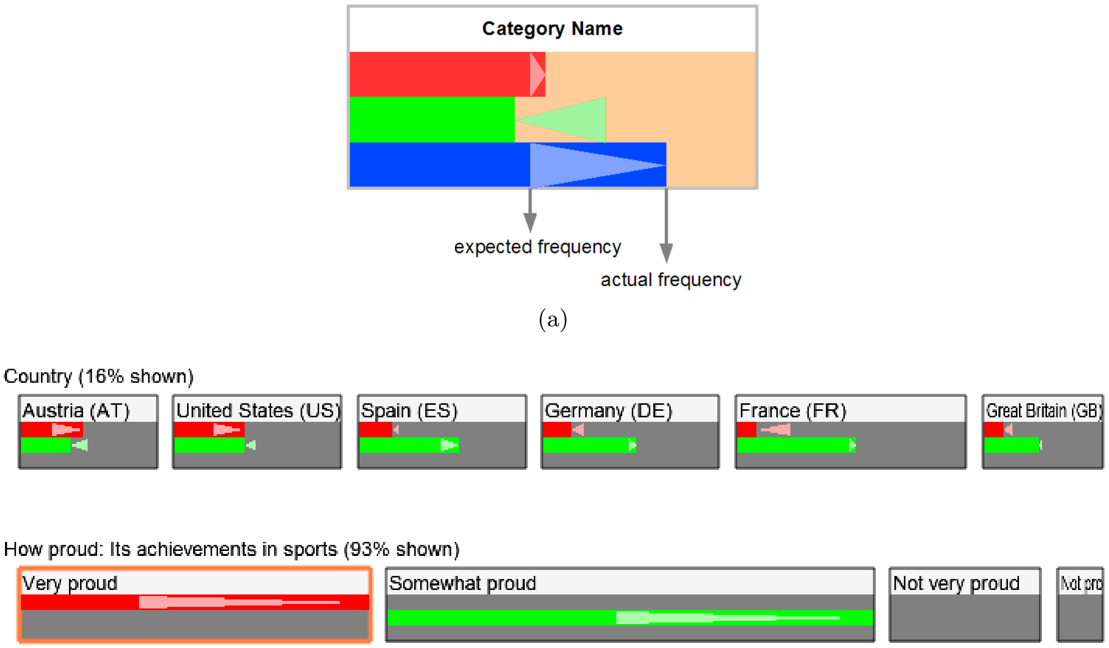
Of course, coloring could also enhance the user's understanding of the data, if used correctly. For example, associations such as "blue = cold" may be used for a "temperature" dimension. Although different color schemes for different hierarchies are currently not supported, it may be an interesting addition in future implementations.

4.4 Additional Features and Optimizations

The following section illustrates further features and optimizations of Parallel Hierarchies, which enhance the basic approach presented above and allow the user to analyze data sets more efficiently. Furthermore, important interactive capabilities of the view are described.

4.4.1 Expected frequencies

The expected frequencies and their deviation to the actual values are two important results of the crosstab (see Section 4.2). The expected relative frequency of a combination



(a)

(b)

Figure 4.5: Arrows are used to visualize the discrepancy between actual and expected frequency, which can enhance the pattern finding process significantly. Image a) shows the idea and b) an example of the expected frequency display integrated in the Parallel Hierarchies visualization.

of two subsets is the product of the individual relative frequencies: $P_{expected}(X \cap Y) = P(X) * P(Y)$.

An example: the data set consists of 1000 items, subset X contains 100 (10%) while subset Y contains 250 (25%). The expected absolute frequency of $X \cap Y$ is therefore 25 while the expected relative frequency is 2.5% ($10\% * 25\%$). In other words, without any prior knowledge about the datset or its distribution, 25 items would be expected to be included in subset X and in subset Y.

Visualization

The first implemented approach uses the categorical information about the membership to the global layers to calculate the expected frequency of entries selected in each layer for each of the displayed categories, i.e. for every category and every layer one expected value is calculated. Typically, the deviation of expected frequencies from actual frequencies is of high interest, as it expresses to which amount the category is over- or under-proportionally related to the selection criterion underlying the various layers.

To visualize the expected frequencies together with their deviation from the actual values, a half-transparent arrow is drawn which has its base at the expected value and

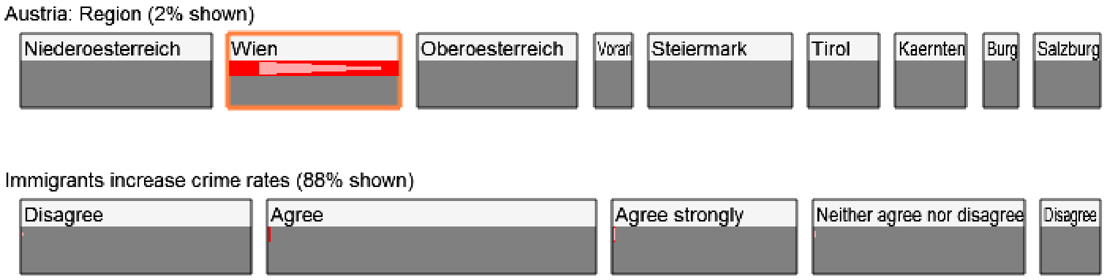


Figure 4.6: This example shows how differently sized hierarchies may be misleading. The "Austria: Region" hierarchy consists of about 900 entries (2% of the entire data set) while the above hierarchy includes all entries (almost 50.000). The category "Wien" is selected but its distribution in the other dimension can not be seen since the actual and expected frequencies are relatively small (with regards to the entire category).

its top at the actual value (see Figure 4.5a for a sketch). Experience shows that this visualization facilitates the perception of relationships between layers and categories significantly.

Figure 4.5b shows the same example as Figure 4.3, but includes the expected frequency arrows. Without looking at any other category, it can be seen that the actual number of very proud Austrians is considerably higher than the expected number. On the other hand, less than half of the expected number of French respondents are very proud of their country's achievements in sports.

Rare item problem

Similar to the layer bars themselves, the expected frequency display can become increasingly hard to read for small categories or subsets. For example, if the category has an item count of 10.000 and the subset only includes 100 items, the expected and actual frequency will be too small to see the deviation between the two, even though it may be significant. Since dimensions may be of different sizes if missing values are not included (e.g., some questions in a survey may be asked only to people from a certain country while others are asked everywhere; see Section 3.3.1 for details on missing data) and categories may be hidden, the graphical representations of categories may be large, even though the associated subset is very small. See Figure 4.6 for an example.

One solution is the use of association measurements, which are independent of the dimension sizes. Here, relationships and patterns are highlighted even if the analyzed subsets of the data have varying sizes. This approach, which is described in more detail in Section 4.4.2, significantly enhances the analysis of differently sized hierarchies. However, it does not solve the problem of hard-to-read layer bars. A possible solution, which is not yet implemented, would be to filter all displayed hierarchies to include only a smaller subset (e.g., the whole survey data set may be filtered to only include the

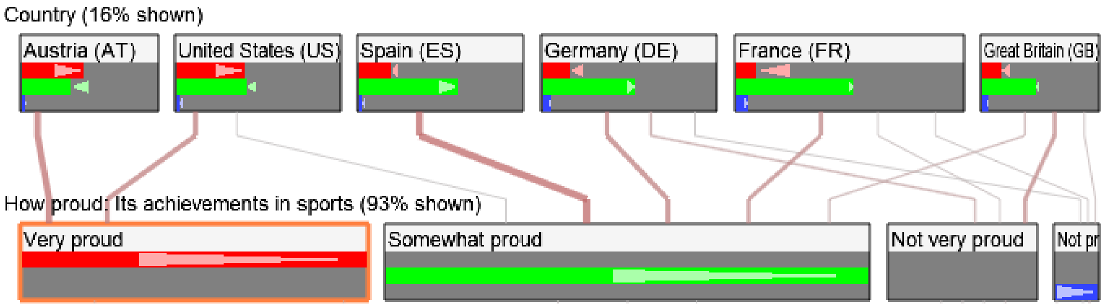


Figure 4.7: Visualizing associations highlights relationships between categories and considerably simplifies the pattern finding process. In this example, the *degree of independence* (the deviation of conditional probability $P(Y|X)$ to the probability $P(Y)$) is used and negative correlations are disabled.

answers of Austrians). Although this prohibits the explorative analysis of the whole data set at once, it may be used to analyze previously found patterns in more detail.

4.4.2 Displaying Trends and Relations

Parallel Hierarchies are able to visualize statistical measurements, similar to the ones found in association rules mining, to assist the user in discovering relationships between categories of adjacent hierarchies. Association rules are basically "if-then" statements (e.g., "if category respondent is from Austria then he or she is likely to be proud of achievements in sports"), details of the concept were described in Chapter 2.3.1. Using Parallel Hierarchies, these statements are represented visually by the lines connecting categories of adjacent hierarchies (Figure 4.7). These "association lines" can be used to display positive as well as negative correlations. In addition to facilitating spotting of relationships between each two particular categories, they also convey an approximate impression whether two hierarchies themselves are strongly related (many bold lines) or not (hardly any lines).

Measurements

Parallel Hierarchies currently include a number of commonly used interest measurements for association rules [22]. The user may change the active measurement at any time and changes are reflected immediately in the visualization.

Support is the relative frequency of the conjunction of two categories X and Y.

$$support(X \to Y) = support(Y \to X) = P(X \cap Y)$$

As explained by Agrawal et al. [3] it is used as a measurement of significance of a subset. Possible values range from 0% (no items in the subset) to 100% (the subset contains the entire data set). If the support is over a user-defined threshold, the subset is called frequent or large. Basically, the support represents the proportion of occurrences which contain categories X and Y. As we have seen before, the disadvantage of this measurement is the rare item problem. Rare items will always have a low support even though they may include interesting patterns.

Example: The categories X and Y are each part of a hierarchy with a total of 100 items and contain 20 and 10 items respectively. 5 items are in both categories, which is 5% of the whole data set. Therefore, the $support(X \rightarrow Y)$ is 0.05 or 5%.

Confidence (sometimes also called strength) is the ratio of the relative frequency of $X \cap Y$ to the relative frequency of X.

$$confidence(X \rightarrow Y) = support(X \rightarrow Y)/support(X) = P(X \cap Y)/P(X) = P(Y|X)$$

Confidence may also be seen as the conditional probability $P(Y|X)$, with X as the given condition or antecedent and Y as the consequent [3]. In other words, confidence is the relative frequency of Y when X is given (e.g. the respondent is known to be Austrian and the relative frequency of "very proud of achievements in sports" under that condition is searched for). Like other probability measurements, values range from 0% to 100%. It is important to notice, that confidence is directed and gives different results for $X \rightarrow Y$ and $Y \rightarrow X$.

The confidence is often used in conjunction with the support. In a first step, significant data subsets are detected by using the support measurement. Then, the confidence is used to extract possible relationships within these frequent subsets. The drawback of the confidence is its sensitivity to P(Y) (the relative frequency of Y). A high probability of Y will likely result in a high conditional probability $P(Y|X)$, even if there is no important association or relationship between X and Y.

Example continued: For categories X and Y the $confidence(X \rightarrow Y)$ is 0.25 or 25% (i.e., 25% of the items of category X also belong to category Y). $Confidence(Y \rightarrow X)$ on the other hand would be 50%.

Lift or interest as proposed by Brin et al. [9] is the ratio of confidence to expected confidence.

$$\begin{aligned} lift(X \rightarrow Y) = lift(Y \rightarrow X) = confidence(X \rightarrow Y)/support(Y) = \\ confidence(Y \rightarrow X)/support(X) = P(X \cap Y)/(P(X) * P(Y)) \end{aligned}$$

The expected confidence($X \rightarrow Y$) equals the *support*(Y) if X and Y are statistically independent. Therefore the lift may be seen as a measurement of how many times more often X and Y occur together than if they were independent. It shows the probability increase of Y, given X. Lifts smaller than 1 indicate a negative association, positive associations are implied by values above 1 (no value cap).

Especially rare item sets can produce very high lift values (as (P(X)*P(Y) becomes very small) and comparing two very small categories may give misleading results, the lift is not directed, i.e. lift($X \rightarrow Y$) and lift($Y \rightarrow X$) produce the same results as minor changes have an over-proportional effect on the lift.

Example continued: The *lift* for categories X and Y is 2.5, which means that X and Y occur 2.5 times more often than if they were independent. The value of 2.5 implies a positive association of the two categories.

Degree of Independence In contrast to the lift, which measures the ratio between confidence and expected confidence, the degree of independence measures the deviation of these two values.

$$DOI(X \rightarrow Y) = confidence(X \rightarrow Y) - support(Y) = P(Y|X) - P(Y) = \\ P(X \cap Y)/P(X) - P(Y)$$

The degree of independence (DOI) is directed and results can be quite different for $X \rightarrow Y$ and $Y \rightarrow X$. Results range from -100% to +100%, with a DOI of zero indicating that both categories are independent. Positive and negative associations are suggested by values above and below zero.

Whereas the lift may produce misleading results for small subsets, the DOI works well for all sizes (e.g. given a confidence of 3% and an expected confidence of 0.5%, the lift is 6.0 while the DOI is +2.5%). Since associations between larger categories may be more important to the user, the support is commonly used to identify significant subsets first.

Example continued: The $DOI(X \rightarrow Y)$ is 15% (= 25% - 10%), indicating a small positive association. For $Y \rightarrow X$ the *DOI* is significantly higher with 30% (in this example 50% of category Y's items are also part of category X, which 30% more than the expected 20%).

Visualization

The measurements described above are calculated for all pairs of categories between adjacent dimensions. The view is read top to bottom, making X the higher category and Y the lower one. Whereas the results of support, confidence and degree of independence can be directly visualized, the result of the lift, which does not have an upper value cap, must first be clamped (4.0 as an indication for a strong positive association is used as an upper value cap) and then scaled between -100 and +100:

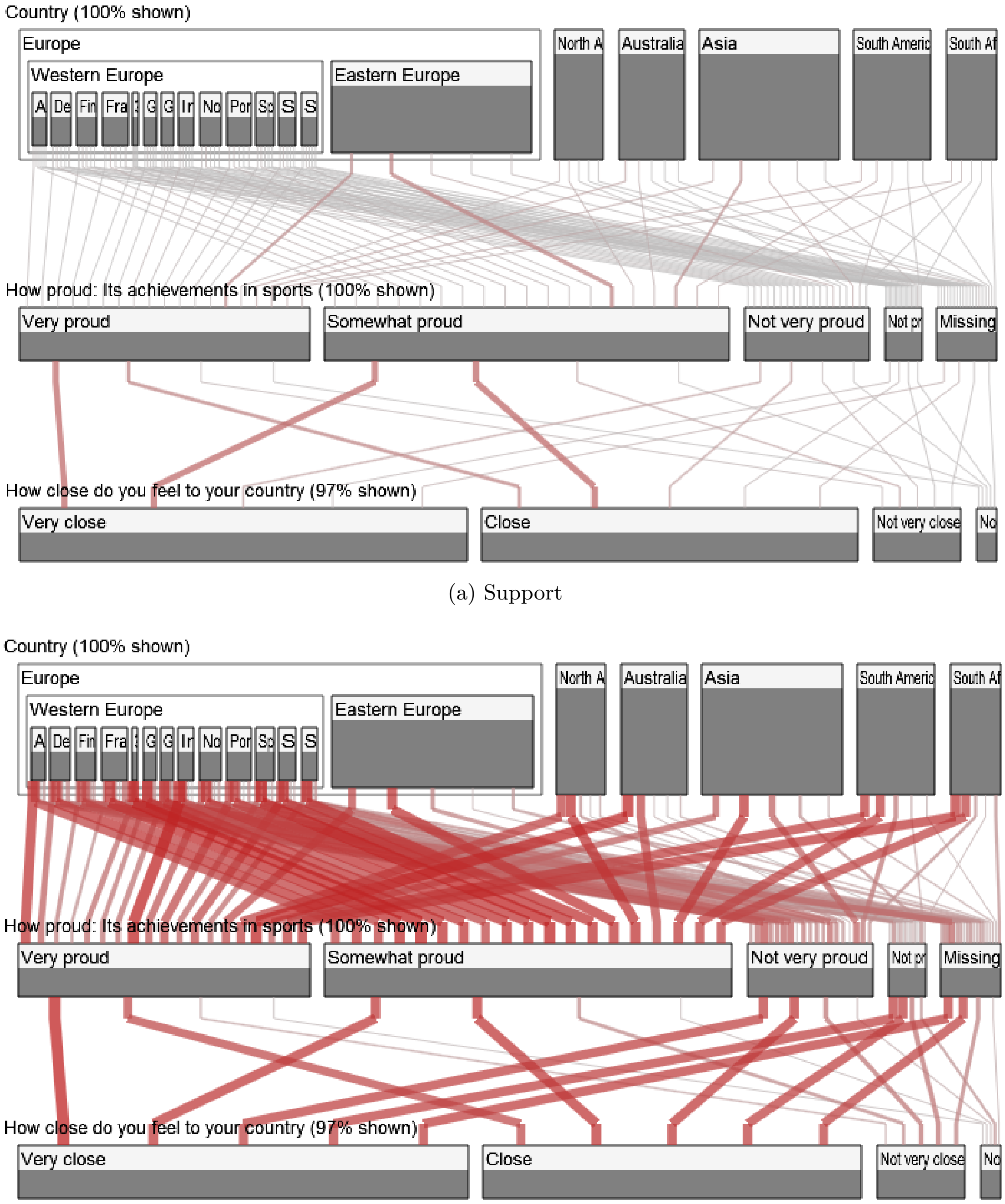
$$Lift_{scaled} = \begin{cases} (Lift - 1) * 100 & \text{if Lift} \leq 1 \\ (\frac{Lift-1}{3}) * 100 & \text{if } 1 < \text{Lift} < 4 \\ 100 & \text{if Lift} \geq 4 \end{cases}$$

To visualize these results, a line is drawn for each pair of categories of adjacent hierarchies and the value of the selected measurement is used to determine the width and color of the line. Positive associations are visualized by red lines and negative associations by blue lines. The saturation of the color and the width of the lines depend on the actual value (e.g. a value of 100% would result in a bright red colored line).

Figures 4.8 a) to d) show a comparison of the four different association measurements using the same data set (negative associations are disabled in this example). Lift and degree of independence produce the most significant visual results, clearly highlighting similar associations. As mentioned before, the lift suffers from the rare item problem, which is also evident in the example (a strong association is shown between the two smallest categories of the two lowest hierarchies). On the other hand, the confidence is clearly affected by the larger categories.

As mentioned above, a threshold may prove very useful, as shown in the Figures 4.9 a) to c). The threshold is a user-defined value between 0% and 100% and association lines not exceeding this values are hidden, strongly improving the perception of the more distinct relationships. The user may change the threshold interactively and results are updated immediately.

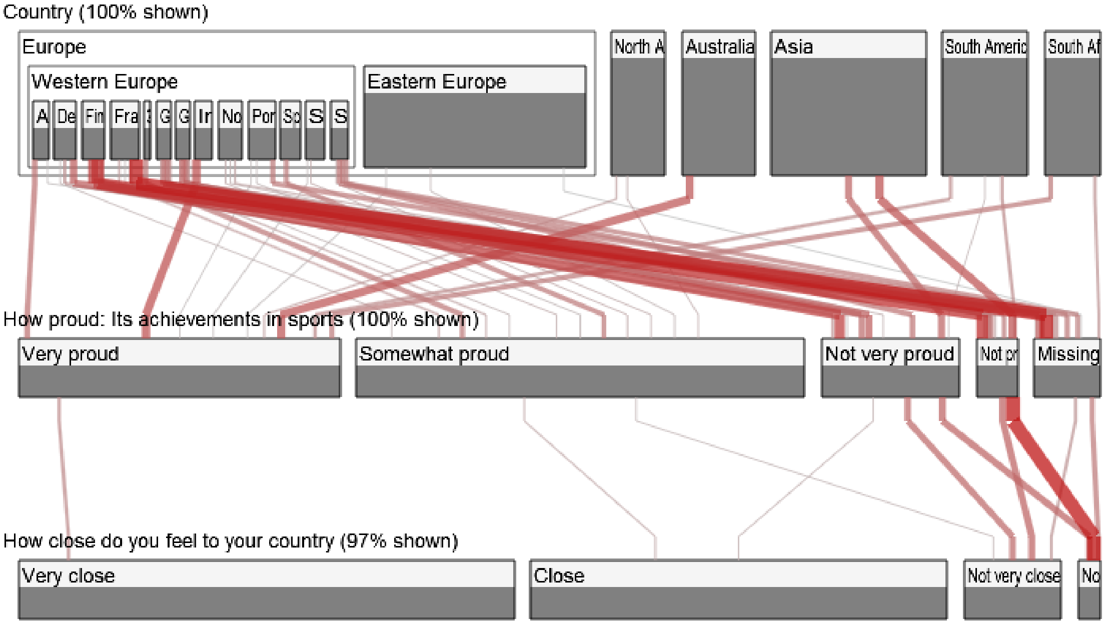
Because positive associations are usually more interesting to the analyst since negative correlations often do not convey any additional information, but are a consequence of the unequal distribution already represented by the positive correlations, negative associations may be disabled. Figure 4.9 shows an example of the degree of independence with negative associations enabled. It furthermore displays, how a large number of connections is visualized simultaneously. By sorting the connections in ascending order of their assigned association measurement results, the most important information is shown in front of the less important connections.



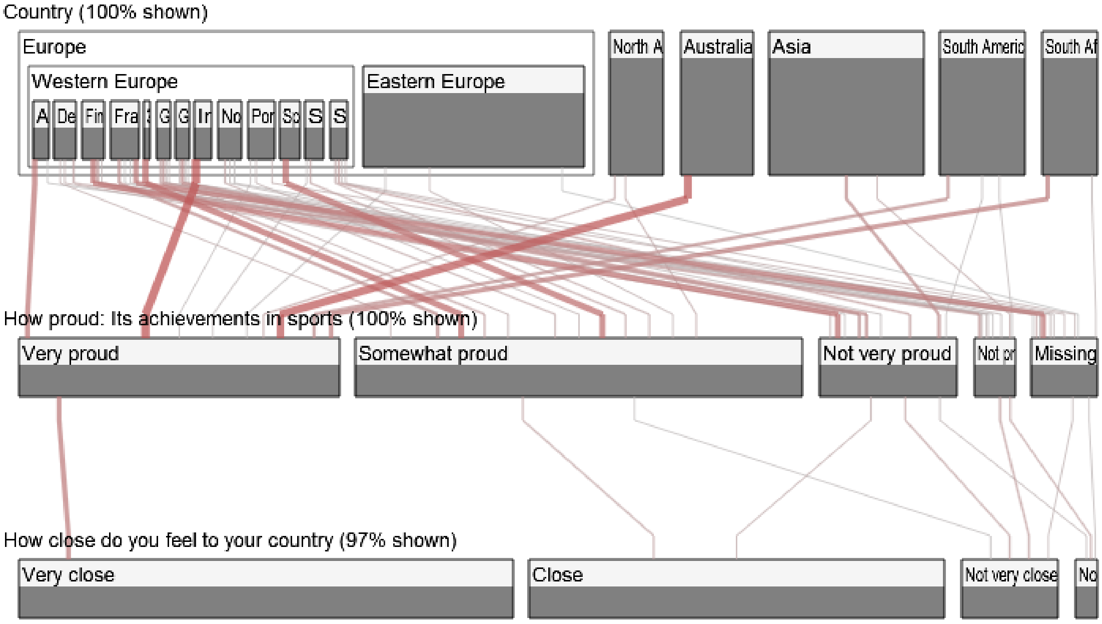
(a) Support

(b) Confidence

Figure 4.8: The different association measurements.

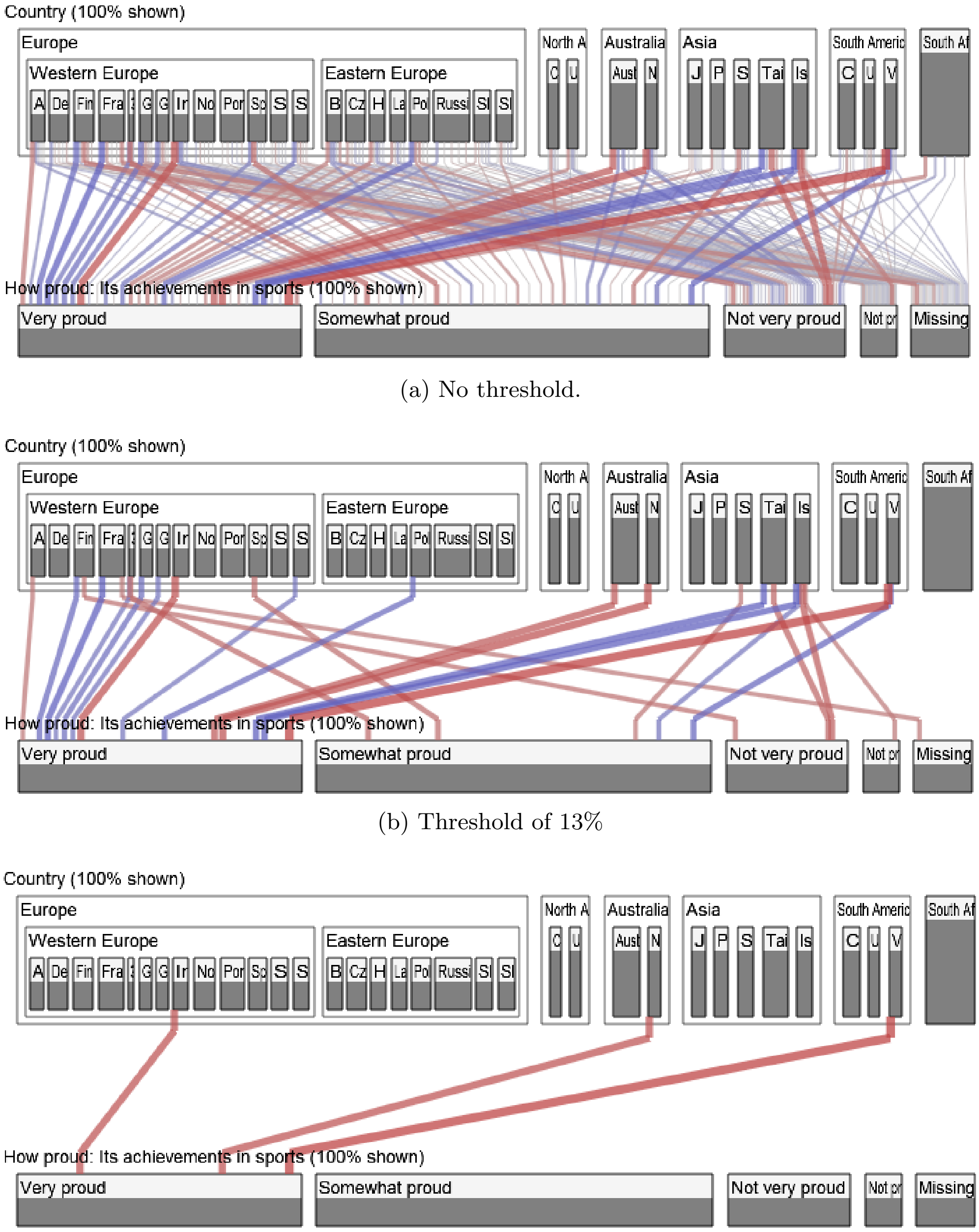


(c) Lift



(d) Degree of independence

Figure 4.8: The different association measurements.



(a) No threshold.

(b) Threshold of 13%

(c) Threshold of 25%

Figure 4.9: Thresholds are very useful to decrease the number of displayed lines, highlighting the most important associations. The pictures show three different threshold settings using the ”degree of independence” as measurement.

Differently Sized Hierarchies

The problem of differently sized hierarchies, as mentioned in the last section, is also evident in the calculation of association measurements. Figure 4.10a shows an example where a 2% subset of the data, the Austrian regions, is compared to a dimension with 50 times more entries. Without any adjustment, the values of support and confidence are almost zero, whereas lift and degree of independence indicate a negative association between the categories, simply because the frequency of one Austrian region is very small in a data set with more than 40 nations. Although mathematically correct, this behavior is usually not expected nor wanted by the user. Therefore, an additional subset, called *shown mask*, which contains an union of shown entries from both hierarchies, is calculated for each pair of adjacent hierarchies. Using the shown mask as a basis to determine the probabilities of each category, the association measurements are adjusted.

This leads to the expected results, as one can see in Figure 4.10b, and associations between differently sized hierarchies can be measured. Unfortunately this introduces a new problem. Whenever categories are hidden or shown again, the shown mask and the association measurements change, which may be confusing for the user. Therefore, this feature is optional and may be enabled or disabled at any time.

4.4.3 Interaction

Besides the already mentioned interaction possibilities, the Parallel Hierarchies visualization offers several features to interactively work with the presented data.

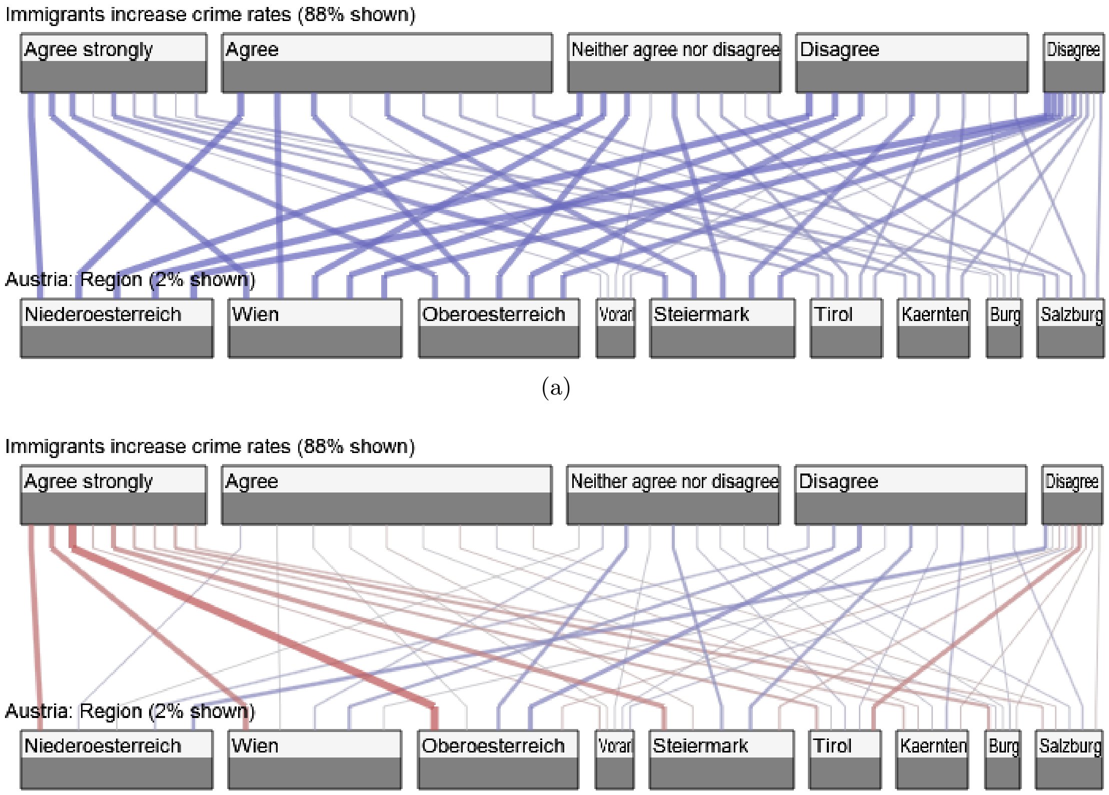
Hierarchy Navigation

The concept of selecting a cut within a hierarchy with selective drill-down and roll-up operations was explained in section 3.3.4.

Using the right mouse button, a category can be selected and a pop-up menu appears. The menu gives the user the possibility to drill-down and roll-up in the displayed hierarchical structure. Additionally, double clicking on a category will drill-down to its sub categories, if it has any. Besides navigating the hierarchy directly within the visualization (which is especially convenient if the hierarchical structure is know to the user), the tree view in the control panel may be used for the same purpose.

Brushing Categories

Categories can be brushed simply by left-clicking on their graphical representation. By selecting a category, the layers of the Bulk Analyzer system are updated based on the complex combinations of brushes the framework offers. Logical operations such as AND or OR, make it possible to select multiple categories in different hierarchies (e.g. one could brush all "very proud Austrians") which makes the view appropriate as a kind



(a)

(b)

Figure 4.10: The top picture demonstrates the problem when associations of two differently sized hierarchies are calculated. The proposed solution shown in b) bases all calculations on the shown mask, which is the overlapping part of the currently visible subsets of both hierarchies.

of filter for other selections. As mentioned in Section 2.4, brushing is not limited to a single view, but may be done across various visualizations.

Brushed categories are highlighted in the visualization by a colored border and can also be de-selected again. The selection of categories is not restricted to categories of the hierarchy's cut, any displayed parent node of the hierarchy may be brushed as well. Figure 4.11 shows how two categories can be selected to analyze their distribution in regard to other hierarchies and categories.

Reordering Categories

The data model facilitates a number of temporary operations on the hierarchy, such as the reordering of categories. For example, it eases the visual analysis, if categories, which show some association or relationship, are located close to each other. Using drag and drop, a category can be picked up and dropped at the desired position. In a hierarchy,

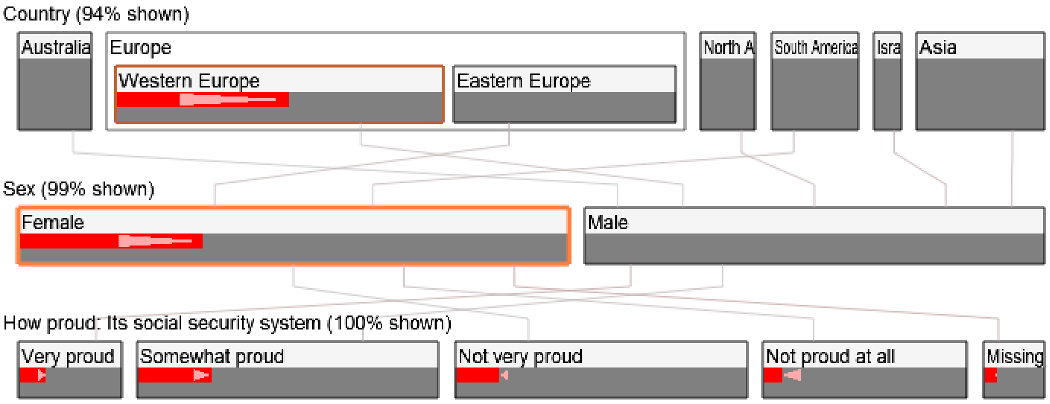


Figure 4.11: The categories "Western Europe" and "Female" have been brushed and combined using the *Refine* Operator, putting all Western European women in the current selection. The active selection (Female) is highlighted by a bright orange border, while the other selection (Western Europe) is shown by a dark orange border.

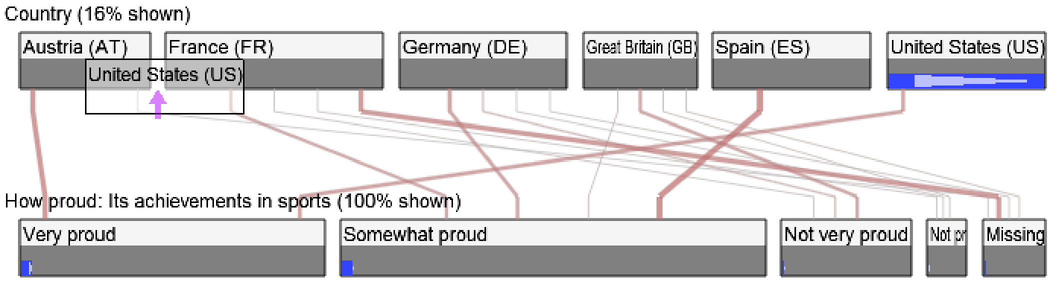


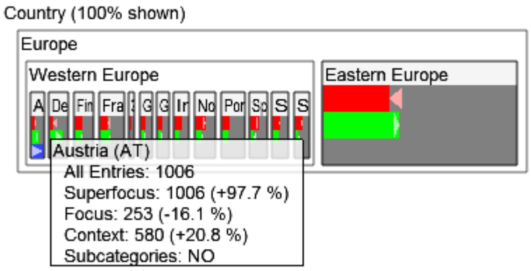
Figure 4.12: The United States category is moved closer to "very proud" since there seems to be some relationship between the two categories.

categories can not be moved outside their parent categories. Therefore, categories of the same subtree will always be located near each other.

Reordering Hierarchies

Similar to categories, hierarchies can also be reordered by using drag and drop. Because association measurements are only calculated and shown between adjacent hierarchies, this is often a necessary and commonly used task.

If little is know about the data, one may assign multiple dimensions to the view before reordering them until an interesting pattern is found. Another way is to first use brushing to detect interesting occurrences in the data set (the layers are used to display a "global" distribution of the data), before hierarchies are reordered to investigate further.



Austria (AT)
All Entries: 1006
Superfocus: 1006 (+97.7 %)
Focus: 253 (-16.1 %)
Context: 580 (+20.8 %)
Subcategories: NO

Figure 4.13: A tooltip can be used to display additional information.

Hiding Categories and Hierarchies

Categories and hierarchies can be hidden from the current display by dragging them out of the screen or using the pop-up menu. Sometimes the number of hierarchies and categories is too large and filtering the displayed data is a key issue in order to work with the visualization. Hidden categories and hierarchies can be shown again at any time using the tree view of the control panel.

Tooltips

While the graphical representations of frequencies, expected frequencies, or associations provide a very good overview of possible patterns and relationships in the data set, often exact information is ultimately needed, i.e., plain numbers to draw the necessary conclusions. This data is presented by means in form of a tooltip, which is shown whenever the mouse cursor is pointed over a category. The tooltip displays information about the absolute frequencies of the layers as well as the deviation from the actual frequency. Furthermore, the tooltip displays if and how many sub categories the selected category contains. Finally, the full label of the category is shown by the tooltip, which is often not possible in the visualization due to spatial constraints.

Chapter 5

Aggregate-based Hierarchical Scatterplot

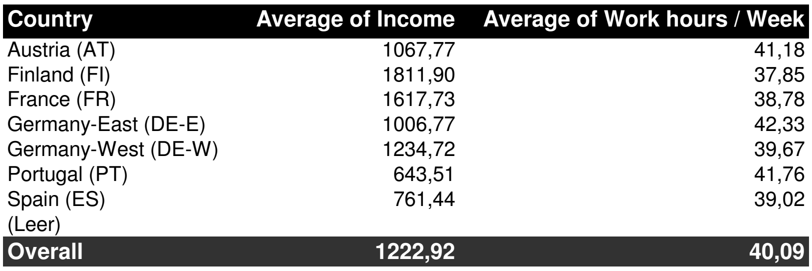
Whereas the concept of Parallel Hierarchies is an approach to visualize multiple hierarchically structured categorical data dimensions at once, the technique proposed in the following chapter presents a detailed view of a single hierarchy in regard to two numerical dimensions, allowing for a unique analysis of data sets. The hierarchy itself is visualized similar to its visual representation in Parallel Hierarchies, but categories are additionally placed on a scatterplot-like view with two numerical axes. In the scatterplot, the categories are represented by glyphs and placed using aggregates, similar to those found in pivot tables (see Section 2.1.1). The visualization is very interactive and aggregates may be changed at any time, making the approach very flexible and useful for a wide range of different applications.

After the motivation of the approach is given, details of the visualization are explained. Additional feature of the technique are presented last.

5.1 Motivation

Most datasets in real-world applications are not strictly numerical nor are they exclusively categorical. Instead, datasets usually consist of both kinds of dimensions (and other, more complex data types, see Section 1.2.2. In Chapter 3 it is demonstrated how a numerical data dimension can be used to identify categories by setting intervals. This enables the user to analyze numerical data with categorical data visualization techniques such as the Parallel Hierarchies.

However, this approach has a few drawbacks. Most importantly, the placement of the interval borders may be quite arbitrary and even small changes may affect the results significantly. Values being close to each other may end up in different categories while more distant values may be combined in the same category. Furthermore, characteristics



Country	Average of Income	Average of Work hours / Week
Austria (AT)	1067,77	41,18
Finland (FI)	1811,90	37,85
France (FR)	1617,73	38,78
Germany-East (DE-E)	1006,77	42,33
Germany-West (DE-W)	1234,72	39,67
Portugal (PT)	643,51	41,76
Spain (ES)	761,44	39,02
(Leer)		
Overall	**1222,92**	**40,09**

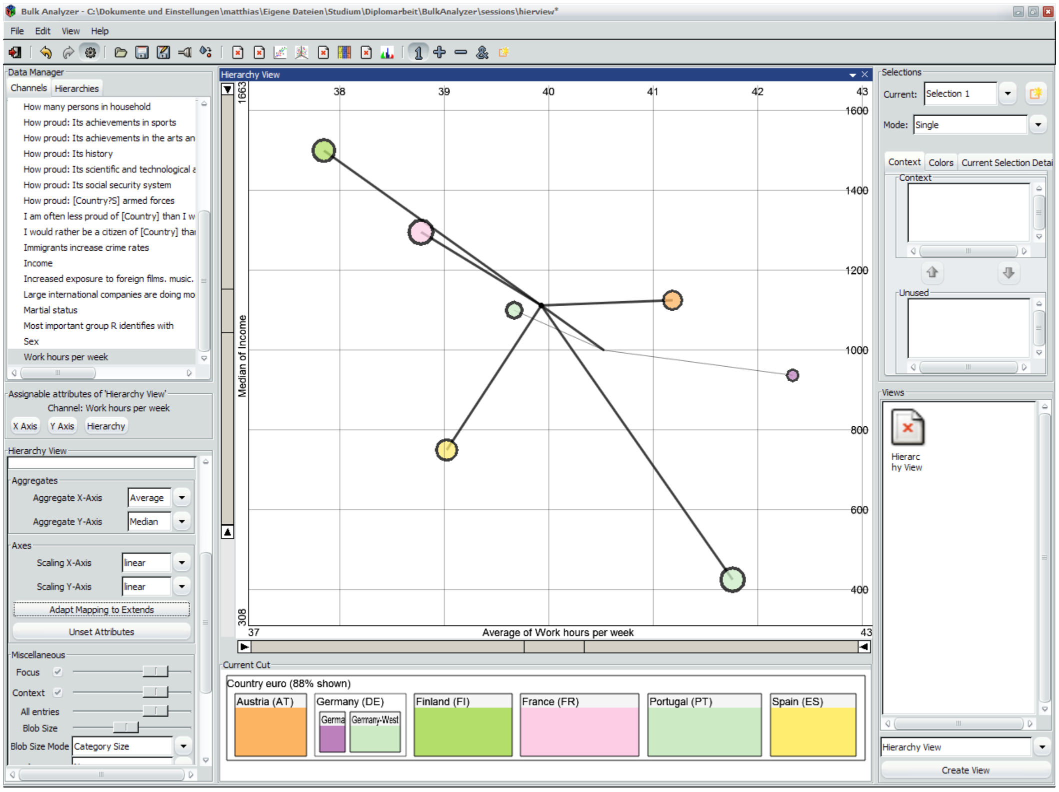
Figure 5.1: Using a pivot table to display the averages of two aggregated continuous (income, work hours per week) dimensions in regard to one categorical dimension (country), which is used to subdivide the entire data set.

of numerical values are lost by the transformation. For example, calculations possible with numerical values are usually not realizable with categories.

Therefore, in order to take advantage of the characteristics of numerical data, the visualization has to support numerical values directly. Various techniques to visualize numerical data, e.g., the scatterplot, were presented in Chapter 2. Most of these approaches usually do not work well for categorical data. The Trendanalyzer visual analysis system [21] (see Section 2.1.2), on the other hand, is an example, how both data types can be successfully combined in a single visualization. However, it is rather limited in other aspects (e.g., no support for hierarchical data, no complex brushing of data, no linking of multiple views, etc.) and the does not allow the aggregation of arbitrary data subsets data to position the circles (i.e., the positioning information is stored within the data set).

Figure 5.1 shows an example of a pivot table, a tool which can be found in most spreadsheet applications (see Section 2.1.1 for details). A pivot table subdivides the whole data set by (multiple) categorical data dimensions and utilizes these partitions to calculate aggregates of (multiple) numerical dimensions. In the above example, the dimension "Country" is used to divide the whole data set into disjunctive subsets. The numerical dimensions are then aggregated for each subset, e.g., to calculate the average income for each country. Various aggregates (e.g., average, maximum, minimum, sum, etc.) may be chosen. Pivot tables furthermore support hierarchical structures, allowing drill-down and roll-up operations in the data set.

The general concept of using numerical aggregates in conjunction with a categorical dimension is very powerful. Hence, the main motivation was to design an appropriate visualization for this task. Additionally, and similar to Parallel Hierarchies, the view should be able to operate with hierarchically structured data and give the user the possibility to analyze the data at different levels of granularity. Another main motivation was to integrate the visualization into the existing Bulk Analyzer data analysis system. Therefore, brushing and the linking of arbitrary views had to be supported by the technique.



Bulk Analyzer - C:\Dokumente und Einstellungen\matthias\Eigene Dateien\Studium\Diplomarbeit\BulkAnalyzer\sessions\hierview*

Assignable attributes of 'Hierarchy View'
Channel: Work hours per week

Aggregate X-Axis Average
Aggregate Y-Axis Median
Scaling X-Axis linear
Scaling Y-Axis linear
Blob Size Mode Category Size

Current: Selection 1
Mode: Single

Current Cut
Country euro (88% shown)

Figure 5.2: This screenshot shows the *Bulk Analyzer* application with the hierarchical scatterplot view displaying the ISSP [53] survey data set. On the left is the control panel of the view, while the controls on the right are global Bulk Analyzer settings. The bottom shows the current cut of the assigned hierarchy, in this case countries. The main visualization is shown in the middle, displaying a filled circle for each category. Similar to a 2D scatterplot, the view has two numeric dimensions assigned. Aggregates are used to determine the positions of the circles, in this case the *average of work hours per week* (X-axis) and the *median of income* (Y-axis). Lines are used to display the hierarchical structure by showing the position of parent nodes.

5.2 Visualization

Figure 5.2 shows the hierarchical scatterplot in action. The basic layout of the visualization is similar to the 2D scatterplot, hence the name of the technique. However, instead of displaying each single entry of the data, a glyph (a filled circle) represents each category, an approach comparable to Gapminder's Trendanalyzer [21]. By default, glyphs are scaled by the frequency of their associated category, i.e., the more entries a category has, the larger is the size of the respective glyph of the category. Optionally, the user may choose to use the level of the category in the hierarchy to determine the glyphs size, which proved useful for complex hierarchies.

In addition to the scatterplot, the hierarchy is also displayed in a separate part of the view below the main visualization. This display is similar to the visualization of a single hierarchy in the Parallel Hierarchies view, as described in Section 4.3.1. On one hand, the cut display is used to navigate the hierarchy and the user may change the cut by drill-down or roll-up operations, reorder or hide categories and define brushes. On the other hand, the cut display provides a second visualization of the hierarchy, also displaying the actual and expected frequencies of the subsets defined by the various layers as in the Parallel Hierarchies view (see Section 4.4.1 for details), making the cut display a very valuable tool for certain tasks, such as comparing the sizes of categories, which is often more intuitive using the cut display instead of the scatterplot, or the analysis of expected frequencies without using a Parallel Hierarchies view. To increase the readability of labels for a larger number of categories, this part of the view can optionally be displayed using an extended area with a scroll bar.

Categories are colored using the same color scheme as in Parallel Hierarchies, see Section 4.3.3 for details. Whereas coloring is an optional feature in Parallel Hierarchies, it is very important in the Hierarchical Scatterplot to match categories in the cut display and scatterplot parts.

5.2.1 Glyph Placement

The most important aspect of the visualization is the glyph placement. By aggregation of data items of the subsets defined by the categories, two values are calculated and used to position the glyph on the two axes. The aggregates are calculated using two numerical dimensions, which are assigned to X and Y axis respectively. For example, a glyph may be placed according to the category's average of one numeric dimension and its maximum value of the second dimension. The assigned dimensions can be exchanged at any time, resulting in a new placement of all categories. Similarly, the active aggregates (see Section 5.3.2 for a list of possible choices) may be changed.

5.2.2 Hierarchical Structure

Whereas only categories of the hierarchy's cut are represented by glyphs, the hierarchical structure above the cut may optionally be visualized by lines connecting each category with its parent. Parent categories are displayed by small black circles, also placed according to their respective aggregates. Figure 5.3 shows how different line widths are used to indicate various levels of the hierarchy. The lower the level, the thinner the lines are drawn, revealing the hierarchical structure.

This approach allows for quick comparisons of the displayed categories to their parent categories, without the need of drill-down or roll-up in the hierarchy. Additionally, the lines connection each category to its parent can be interpreted as distance vectors. Analysis and comparison of these vectors provides further information about the data. For example, it may be very significant if a category lies close to entire hierarchy (which

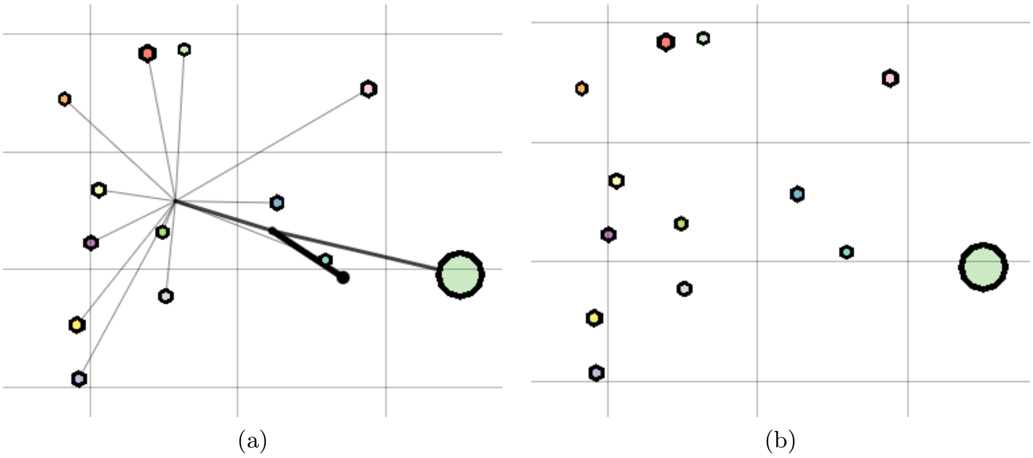

(a) (b)

Figure 5.3: On the left the hierarchical structure is displayed using lines, which connect categories to their parent nodes. The connecting lines are disabled on the right.

is also represented as a parent node) on one axis, but far from the overall result on the second axis. The vectors also help revealing patterns between the categorical hierarchy and the numerical dimensions.

5.2.3 Example

Using the Hierarchical Scatterplot approach, the data from the pivot table (Figure 5.1) can be graphically visualized, as illustrated by Figure 5.4. One can see at first glance, that Finland is the country with the highest average income (about 1800 euros a month). Surprisingly, it is also the country with the lowest average number of working hours per week (under 38). Looking at the sizes of the categories it can be seen that the number of respondents varies between the different countries with most people coming from France.

The overall averages of the entire data set is shown by the black circle all categories are connected to (in this example, the overall average is located behind the glyph of Germany). Concluding from this data set, the average income in France and Finland is higher than the overall average. While Finland's average age is clearly below the overall average, France has the highest average age of all compared countries. In general, there seems to be no correlation between average income and age.

5.3 Additional Features and Optimizations

Based on the basic idea, which was presented above, several features were implemented to enhance the Hierarchical Scatterplot approach. Important features, such as the layer integration and the placement of glyphs using aggregates, as well as optimizations are described in the subsequent sections.

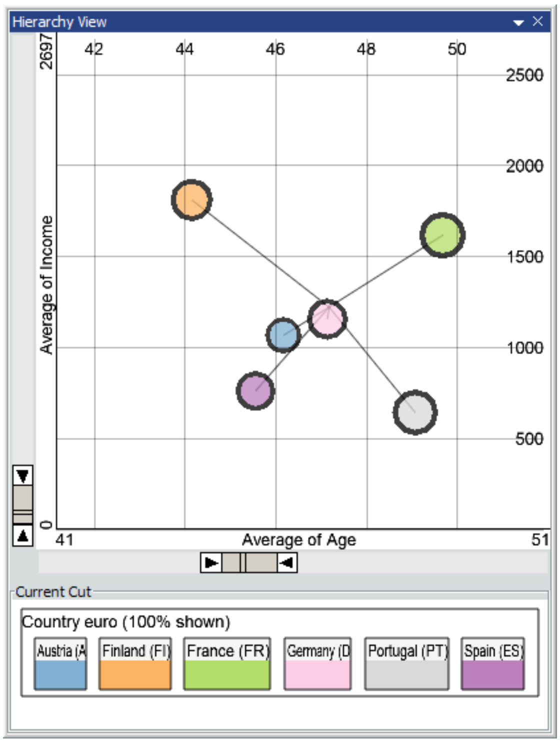


Figure 5.4: The picture shows the cut displayed on the bottom, with seven countries currently within the cut. In this example, one can compare average income and average age of different countries.

5.3.1 Layer Integration

The two parts of the view integrate the layers of the Bulk Analyzer system differently: The display using nested bars handles subsets as described for the Parallel Hierarchies view (see Chapter 4), thus highlighting the frequencies (both actual and expected and the difference between them). The Hierarchical Scatterplot display on the other hand visualizes the subsets defined by the layers in a similar way as for child nodes within the hierarchy, thereby stressing the deviation of the selected subsets from the entire categories with respect to the current aggregates of the attributes mapped to the X- and Y- axes. For this reason, the Hierarchical Scatterplot uses additional glyphs.

For each category and layer, a glyph is placed, scaled and displayed similarly to the default glyphs. Additionally, lines are drawn connecting these glyphs to the glyphs of the entire categories. To distinguish the different subsets, the border of the additional glyphs are colored using the layer's color (Bulk Analyzer's three layers are red, green and blue by default).

Figure 5.5 depicts how this feature can be used in conjunction with linked brushing. The example again compares the average income and average working hours per week

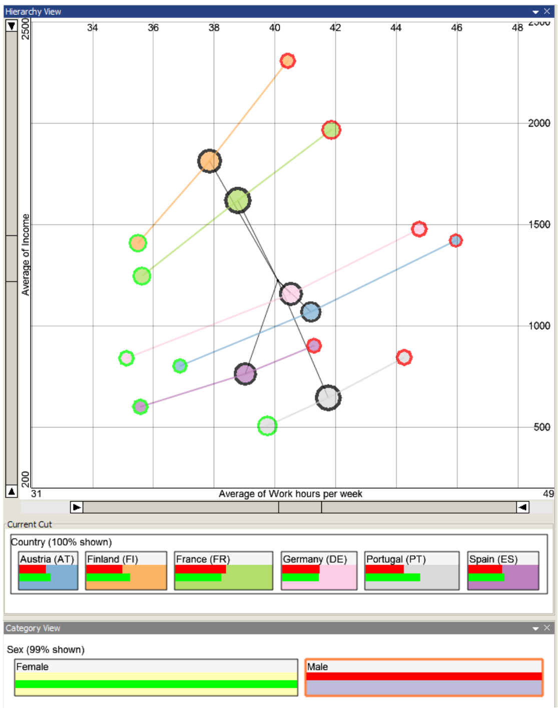


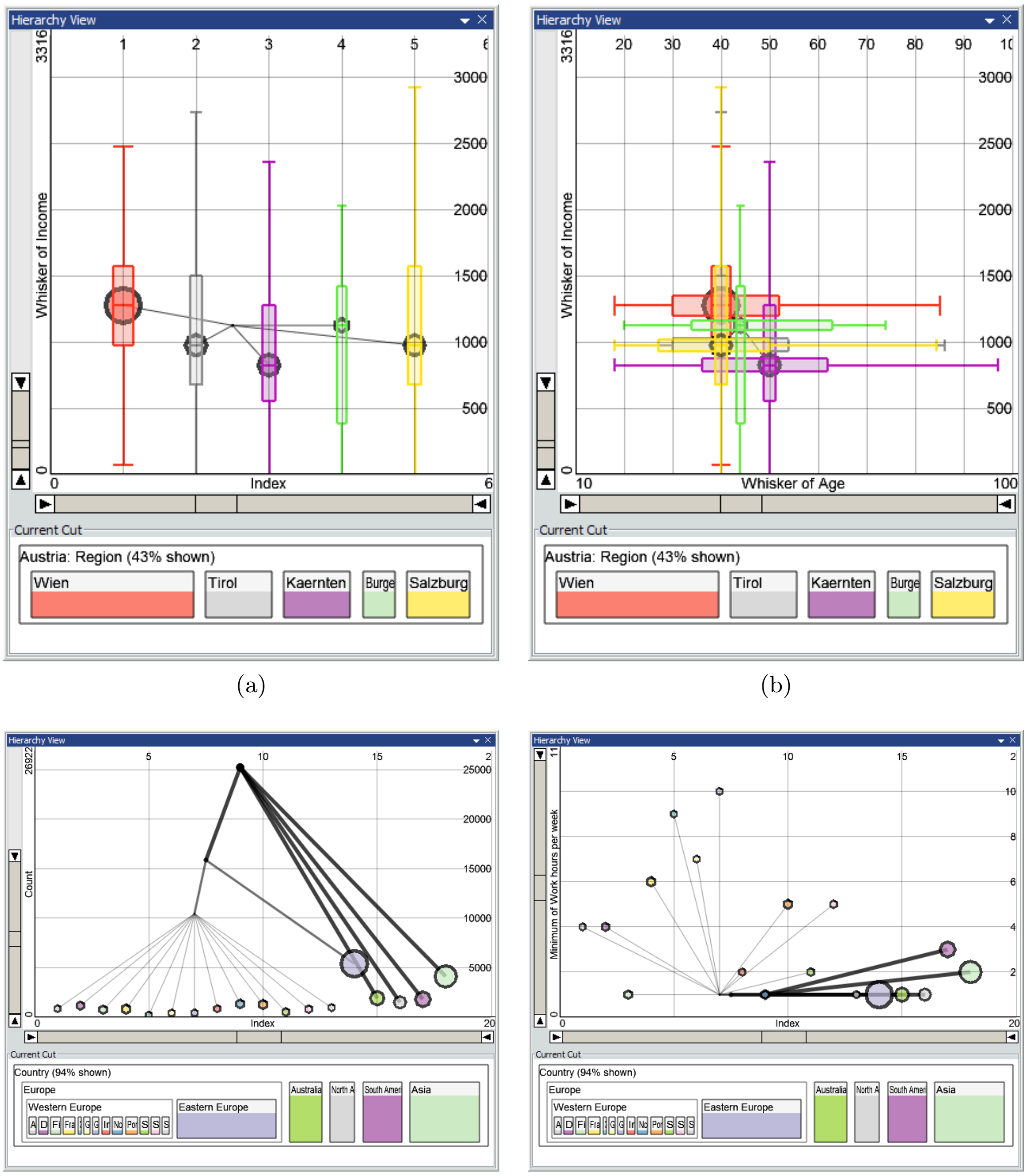
Figure 5.5: Using a Parallel Hierarchies view, female (green) and male (red) subsets were brushed using *Bulk Analyzer's* focus and context system. The Hierarchical Scatterplot then shows how these subsets affect each category by positioning additional glyphs. One can easily see how average income and working hours per week differ for men and women in the displayed countries.

in various countries. Another view - a Parallel Hierarchies visualization - was used to define two layers by brushing two categories of another dimension, in this case male and female, respectively of the sex dimension. In addition to comparisons between countries, one can now also analyze how male and female results vary. The visualization clearly shows that female respondents earned and worked less than men. Furthermore, the subsets can also be compared to other categories.

5.3.2 Aggregates

The Hierarchical Scatterplot provides various aggregates, which can be chosen independently for each axis. All aggregates calculate values to position the corresponding glyphs, but may also provide additional information. For example, each aggregate defines the value range needed for the data mapping process.

Average/Median: The average aggregate and the median aggregate calculate the average and median respectively of their assigned subsets. The result is used to place



(a)

(b)

(c)

(d)

Figure 5.6: Four examples of how the various aggregates may be combined to display different aspects of information. In picture a) a whisker aggregate is used for the Y-axis while the index is used for X to show the distribution of income in the individual Austrian regions. In b) the box plots for income and working hours are shown for five regions. c) shows how a combination of index and count aggregates can be used to reveal the hierarchical structure of a hierarchy with respect to the sizes of various branches. The count was replaced by the minimum aggregate in d).

the glyph in the shown coordinate system. The value range equals the value range of the assigned numerical dimension. An example of the average aggregate is shown in Figure 5.4.

Sum: The sum aggregate adds up all values of a given subset, e.g., the total revenue of a specific store. The value range is calculated by summing up all negative values for the minimum and all positive values for the maximum.

Minimum/Maximum: These aggregates yield the minimum and maximum values of a subset. The value range is defined by the smallest possible minimum, which is the minimum of the entire dimension and the largest possible maximum, which is the maximum of the dimension.

Index: This "aggregate" merely returns the position of the category in the current cut, i.e., it does not aggregate any data and is thus independent of the data dimension mapped to the axis. It is nevertheless useful to equally distribute the glyphs on one axis, which allows for more exact 1D comparisons. Minimum and maximum values are determined by the number of categories in the cut.

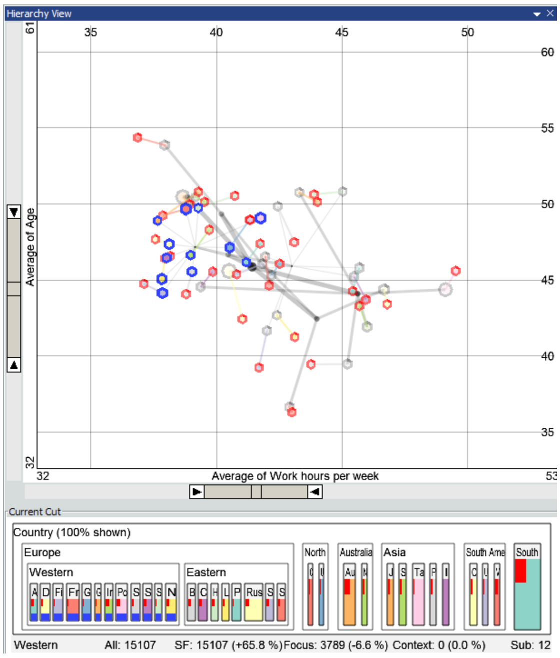
Count: The count aggregate calculates the absolute frequency of a subset, i.e., the item count. Like the *index*, this aggregate does not depend on the data of the assigned numerical dimension. Possible values range from zero to the total number of entries in the data set. The count aggregate can be used to display the hierarchical structure of a hierarchy in tree form (Figure 5.6).

Whisker: The whisker aggregate calculates all values needed to draw a box plot (see Section 2.1.2 for details). Basically, the box plot is used to interpret the distribution of data based on the median, the interquartile range and the 2,5% and 97,5% percentiles. When choosing the whisker aggregate, a box plot is drawn over the filled circle, which is used as by all other aggregates. The median is used to place the circle and box plot on the screen.

The examples in Figure 5.6 illustrates some of the possible ways to combine aggregates and demonstrates how versatile a categorical hierarchy can be visualized with this approach.

5.3.3 Transparency

Whenever many categories are displayed simultaneously, the visualization can get clustered and difficult to use. The display of brushed subsets makes the problem even more severe and important information becomes increasingly hard to see. One reason is, that



Western All: 15107 SF: 15107 (+65.8 %) Focus: 3789 (-6.6 %) Context: 0 (0.0 %) Sub: 12

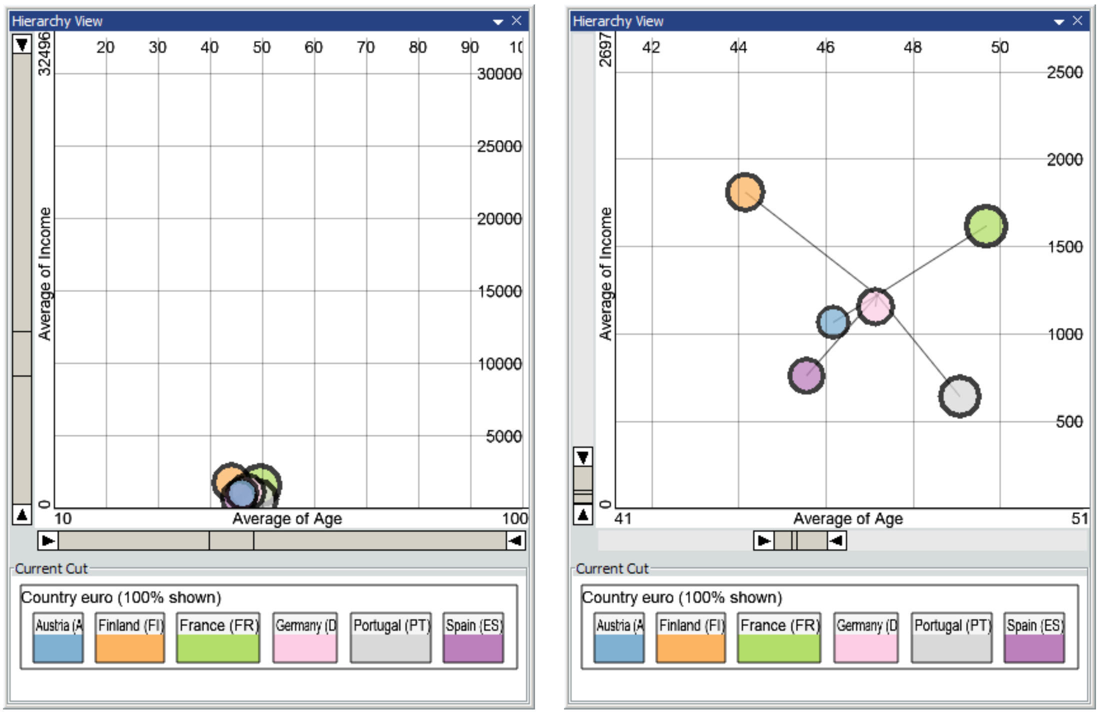
Figure 5.7: This example shows 33 categories being visualized using the Hierarchical Scatterplot. 45 additional glyphs are drawn for brushed subsets of the data. Using transparency these subsets are highlighted.

glyphs may overlap and larger glyphs often completely hide smaller ones. In order to improve on this, the glyphs are first sorted according to their size and consequently, smaller glyphs are drawn in front of the larger glyphs.

Besides that, each layer has an individual opacity/transparency setting attached, which can be set by the user. Using these settings, a large number of glyphs can be displayed better, though it is almost impossible to visually match the representations as bars and as glyphs (Figure 5.7). As the Hierarchical Scatterplot becomes more and more similar to a non-hierarchical scatterplot as the number of displayed glyph increases, the view still shows important aspects of the data, even if a clear identification of all shown categories is not possible simultaneously due to their number: clusters and correlations stay visible well and trends of their deviation of brushed parts (e.g., ”do all glyphs deviate into the same direction?”) are still perceivable. In the example, the brushed parts are highlighted using a higher opacity setting than the main category glyphs.

5.3.4 Interaction

Like Parallel Hierarchies, the Hierarchical Scatterplot visualization is a highly interactive technique. Besides the cut navigation, which works similar to the Parallel Hierarchies view, the Hierarchical Scatterplot provides the user with a set of interaction capabilities.



(a) (b)

Figure 5.8: Picture a) illustrates how categories are often crowded in one spot. Adopting the mapping to the extents of the glyphs results in a much better visualization as seen in b).

Brushing categories

Categories can be brushed in the cut display (which works equally to Parallel Hierarchies, see Section 4.4.3 for details) as well as directly in the Hierarchical Scatterplot. Any category glyph can be clicked to brush the associated subset in the data set. If more than one glyph is displayed under the mouse cursor, all categories will be selected. The user can furthermore draw a selection rectangle to select multiple categories at once. Besides that, the user can point the mouse cursor over any glyph or category which puts the category in the super focus. This highlights the category in both parts of the view, the scatterplot and the cut display, allowing for an easy identification and matching of categories.

Again, Bulk Analyzer's framework allows the combination of multiple brushes and links the Hierarchical Scatterplot with other views. Together with the opacity settings mentioned above, brushing and selecting categories proved also useful to highlight interesting areas in the Hierarchical Scatterplot.

Data Mapping Adjustments

The Hierarchical Scatterplot uses the data mapping controls provided by the Bulk Analyzer system to for zooming into certain areas, moving these areas or using distortion (Figure 2.18).

Initially, the entire value range as determined by the aggregates is shown. Because glyphs are often crowded in a small area (e.g., all countries have an average income between 500 and 2000 euros, but the possible values range from 0 to 10.000), the view provides an option to automatically adopt the mapping to the extents of the displayed glyphs (Figure 5.8).

Tooltips

Similar to Parallel Hierarchies, tooltips can be used to display detailed information. The user may point the mouse cursor over any category in the cut display as well as over any glyph in the visualization to see a tooltip. The most interesting information of the tooltip, besides the name of the category itself, is probably the exact result of the assigned aggregates.

Chapter 6

Case Study

The following chapter illustrates the usefulness of the Parallel Hierarchies and Hierarchical Scatterplot visualizations by analyzing a real-world data set. Both techniques are used together to explore a survey data set with more than 44.000 entries and 30 dimensions, of which 25 are categorical and 5 are numerical. The goal of the analysis is to discover unexpected relations and interesting patterns.

6.1 The Data

The analyzed dataset originates from a survey about national consciousness and identity. The survey was conducted by the International Social Survey Program (ISSP) in 33 countries between February 2003 and January 2005. The questionnaire consists of 104 general questions, plus several questions specific to particular countries. All in all, the data set consists of 241 dimensions and the number of respondents (data entries) is 44.170. The data set is divided into demographic values (sex, age, martial status, income, etc.), a wide range of question groups (identification with the town/region/country/continent, perceived pride in several fields, immigration, globalization, etc.) and dimensions specific to particular countries (party affiliation, region, etc.). Most values in the dataset are coded, i.e., the answer *very proud* is represented by the number 1 to reduce the file size. For some groups of questions, the numbers also provide an order of the categories (e.g., *very close* to n*ot close at all*), whereas there is no inherent ordering for others. For an easier understanding, the numbers were replaced by the names of the categories. Most questions allow for only a limited number of possible answers, which explains why the majority of dimensions is categorical.

For this case study, a selected subset of all dimensions has been taken into consideration. In particular, the case study focuses on the question groups of national pride and identity. Therefore, 30 interesting dimensions were extracted from the original data set. Table 6.1 lists and describes some groups of questions and the possible values of categorical dimensions, numerical dimensions are listed in Table 6.2. Besides several demographic variables (e.g., *country, sex, martial status, highest education level*), dimensions specific to Austria have been included to further explore patterns in regard to Austrian regions or political parties. These dimensions are listed in Table 6.3. If a respondent did not answer a certain question or the answer was invalid, the entry was marked *missing*. Most of the time, these entries are hidden in the visualizations.

There is no inherent hierarchical structure, but certain dimensions lend itself to such a structure (e.g., the countries can be structured based on the geographical location or mother tongue). Furthermore, hierarchies may be refined by using the generalization and specialization operations explained in Chapter 3.

Group	Example	Description
Proudness	How proud are you of: countries achievement in sports.	Various other questions regarding pride in specific areas such as sports, arts and literature, or armed forces. Answers categorized in very proud, somewhat proud, not very proud and not proud at all.
Immigrants	Immigrants increase crime rates.	How much the respondent agrees to statements about negative effects of immigrants. Answers: agree strongly, agree, neither agree nor disagree, disagree, disagree strongly.
Identity	Most important group you identify with.	Groups the respondent identifies with. Possible answers: occupation, race/ethnicity, gender, age group, religion, political party, nationality, family, social class, part of country.

Figure 6.1: A selection of question groups with categorical values as answers.

Variable	Description
Age	Age of respondent, ranging from 15 years to 98. In some countries the minimum age is 18.
Income	Respondent earnings in local currency. This variable is country specific, even if it is integrated in the same variable. Usually the values are monthly incomes, but some countries state annual incomes.
Working hours per week	The number of hours (usually) worked weekly. Ranging from 1 to 96 hours (which includes all values above).
Years of schooling	The number of years the respondent spent in schools.

Figure 6.2: The numerical dimensions.

Variable	Description
Region	The Austrian province the respondent lives in: Wien, Niederösterreich ...
Party	Party affiliation, possible answers: SPÖ, ÖVP, Grüne, FPÖ, other.
Size of community	Coded size of community: more than 1 million, 50.001 to 1 million ... less than 2.000.

Figure 6.3: The dimensions specific to Austria.

6.2 Goals

As stated before, this case study focuses on a subset of the question asked by the International Social Survey Program in their survey about national identity. Specifically, national pride across countries and possible influences are analyzed in detail. Furthermore, a closer look at Austria is taken.

At the beginning of this case study, little was known about the survey results. Various questions guided the research:

- Are the specific areas of national pride related?
- How does national pride compare across countries?
- Is pride influenced by sex, age or education?
- How influential is the party affiliation in Austria on questions asked?

Both previously proposed visualization techniques, Parallel Hierarchies and the Hierarchical Scatterplot, were used to answer these questions, as well as explore the data set to reveal other potential patterns within the survey data, while, at the same time, demonstrating their usefulness.

6.3 Categorization and Hierarchical Structuring

Because a majority of dimensions in the survey dataset are inherently categorical (i.e., they have a small number of possible values), the process of categorization is straight forward.

Initially, most dimensions do not directly lend itself toward a hierarchical structure. Only for the dimension "Countries" exists an inherent structure by combining countries of the same region and regions of the continent. The result (Figure 6.4) illustrates that a majority of respondents came from Europe, most of them from Western European states. Africa is the only continent only merely represented by a single country, namely

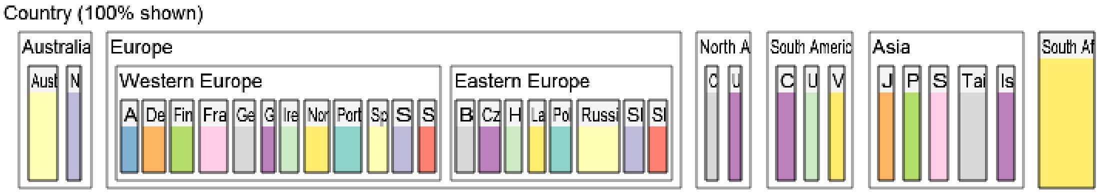


Figure 6.4: The structure and representation of the countries dimension.

South Africa. But, as can be seen, the number of respondents in South Africa was larger than in any other country.

In order to improve the confidence of the results, it has proved useful during the exploration to combine similar categories (e.g., *not proud* and *not proud at all*) with small numbers of entries into hierarchies to obtain less yet more significant categories. In contrast, the refinement operation has been used to derive more specific categories and integrate information of other dimensions into a single hierarchy. For example, the hierarchy ”Austrian Party Affiliation” has been refined by the dimension ”Sex”, resulting in an additional level of categories in the hierarchy (”SPÖ + Male”, ”SPÖ + Female”, ”¨OVP + Male”, etc.).

Later in this case study, it can be seen, how useful the support for hierarchical structuring in both views is. The possibility to analyze the data at different levels of detail greatly enhances the user’s understanding of the data (and its structure). Compared to other categorical data visualization techniques, like Parallel Sets or the Mosaic Display (see Chapter 2), the use of hierarchies also proved to be time-saving. A lot of tasks do not require the analysis of the highest level of detail (e.g., compare the survey results of Austria with the overall results of Asia) and interactive selection of the displayed cut (see Section 3.3.4 for details) allows for a more flexible analysis than Parallel Trees (Section 2.1.3).

6.4 National Pride in Comparison

The first goal was to compare the national pride of different countries. In order to obtain this goal, six dimensions, all regarding pride in specific areas, have been added to a Parallel Hierarchies visualization and the connection measurements of lift and degree of independence (see 4.4.2) have been used.

Figure 6.5 shows at a glance that pride in *achievements in sports*, *history*, *arts* and *literature*, and *science* is generally very large. Sports is the most frequently cited source of national pride, with more than 75% respondents indicating it makes them proud of their country.

Comparing only the *very proud* categories, the *achievements in history* stands out as the one with the most entries. People are distinctly less proud of their country’s

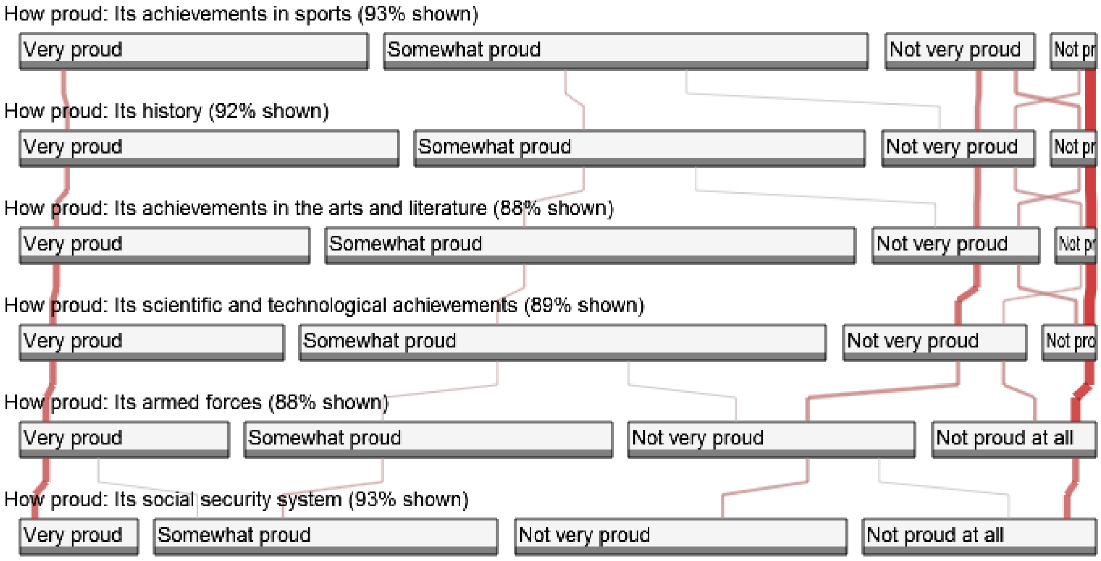


Figure 6.5: Displaying the *lift* (a symmetrical measurement of how many times more often two categories X and Y occur together than if they were independent) of adjacent dimensions reveals a pattern between similar answers. The colored red lines indicate, that the degree of pride correlates across the several dimensions (e.g., if one is very proud in one area, he or she is likely to be very proud in others as well).

armed forces and less than 50% are proud of their social security system. The latter is also the area with the least amount of *very proud* people. Democracy and global political influence (not shown in the picture) also rank low, indicating that people tend to be more proud of achievements in non-political areas than of achievements of their government.

Whereas the above analysis could have been done with similar approaches, like Parallel Sets or Parallel Trees, the Parallel Hierarchies view introduces several statistical measurements to highlight associations in a unique and efficient way. These measurements significantly enhance the visualization in several ways. First, they provide a quick and time-saving overview of potentially interesting relationships without the need of brushing. Secondly, proficient users are provided with additional information, which is not available elsewhere.

Using the association lines, it is possible to quickly identify several relationships between the different dimensions. Obviously, the degree of pride is strongly correlated across the several areas, which means that a person stating to be very proud in one area, is more likely to be very proud in other areas as well. The same holds true for *not very proud*, *not proud at all* and, to some extent, *somewhat proud*. Since the association lines only show patterns between adjacent dimensions, moving around the dimensions to prove this pattern for all combinations. Brushing and enabling the layer bars may also be used to assist this task.

At this point, it is important to repeat again that the lift measurement favors small categories and may produce large results for two categories with a low confidence value. The result produced by lift and degree of independence measurements basically tells us, how more likely something occurs compared to the expected value. Therefore, one has to be careful to draw conclusions such as "if a person is not proud of achievements in sports, he or she is also *most* likely not proud of achievements in science" without taking the *support* and *confidence* into account. Although the support between the *not proud at all* categories is low in the above example, the confidence validates the findings and further prove of their relevance can be found by the *degree of independence*, which shows similar results to the *lift*.

6.4.1 Pride in Sports Achievements

After comparing multiple question concerning pride, the next objective was to analyze these dimensions with regard to the various countries. Figure 6.6 shows the process starting at the highest level of the countries hierarchy. At first, only a pattern between Australia and *very proud* as well as Asia and *not proud* can be seen by using the *degree of independence* measurement. After drilling down one level, the individual countries are displayed. Drill-down operations are especially important if the grouped categories are quite inhomogeneous, as can be seen in Europe, where the situation is very different for geographically close countries. Here, it can also be useful to drill down only on selected categories (e.g., to compare the results of each Western European country with the overall results of Asia), which is a unique feature of both views, Parallel Hierarchies and the Hierarchical Scatterplot.

By only displaying positive associations and using a threshold to filter out weak connections, it is easy to identify dominating relationships. Once again, this demonstrates the usefulness of statistics in conjunction with the visualization. As a result of this step, people in Australia, New Zealand, Austria, Ireland and Venezuela tend to be very proud of their nation's achievements in sports, whereas Finland, Great Britain, Switzerland, Poland, Chile, Thailand and Israel are considerably less proud of the achievements of their countries in sports. The *lift* measurement highlights similar relationships and the findings can be verified by using the *confidence* measurement, which shows a high confidence between the above described relationships. Because each country only represents approximately 2.2% of the whole data set, the *support* is naturally small (however, 2.2% still represent about 1000 occurrences in the data set).

In the next step, an emphasis was put on Western European countries (Figure 6.6d) by hiding all other regions. Then, the *very proud*, *not very proud* and *not proud at all* categories are brushed. Using these layer and the expected frequency display, it is now easy to compare these categories for each category. Austria and Ireland are again the two countries with the highest percentage of very proud respondents (about 50%). It is also interesting that the number of Irish people stating not to be proud of achievements in sports is almost zero.

This procedure also showed, how the association lines can be efficiently used together with brushing. First, the lines were used to identify several interesting relationships between categories. Then, brushing was used to confirm these patterns as well as provide additional information about the data distribution. By using the tool tip function (i.e., hovering the mouse over a specific category), details about the relationships can be accessed (e.g., to see how many Austrians stated to be very proud of their country's sport achievements). Additionally, the expected frequency arrows, a novel feature of Parallel Hierarchies, greatly help the understanding of the patterns within the data set.

6.4.2 Pride in Other Domains

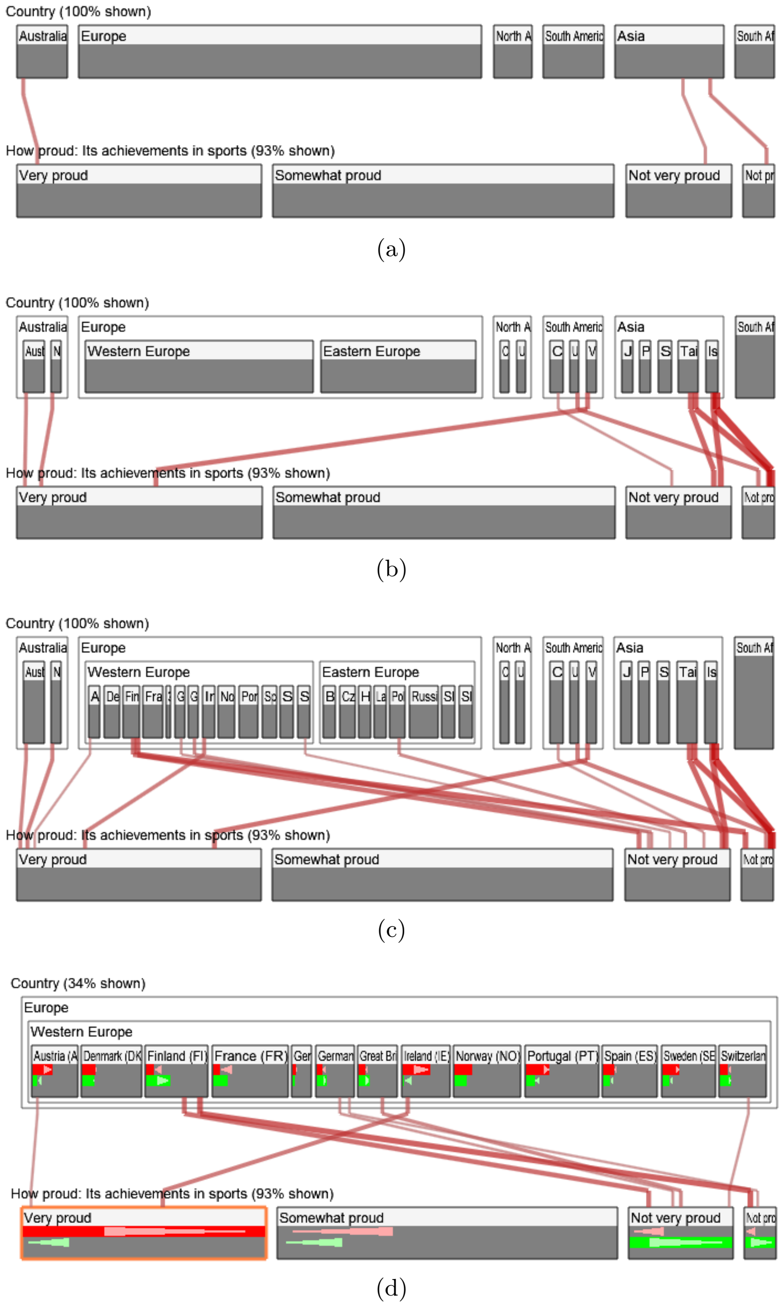
Similar to *achievements in sports*, other questions regarding pride have been analyzed as well. Analogically, the structure of the countries hierarchy has been used to identify trends in various levels of detail, as illustrated by Figure 6.7. Western European countries generally tend to be proud of their social security system, while the majority of Eastern Europeans is not proud of it at all. Drilling down in the Western European nations has revealed, that the four nations with the highest degree of very proud people are Austria, Denmark, Finland and France. Further analysis (hiding countries and using brushing) has shown that Austria is leading in the pride on social welfare.

The United States, Great Britain and Israel have the greatest pride in their armed forces, followed by Australia, New Zealand and Venezuela. Western European nations rank average with an equal distribution of the four possible answers. Eastern Europe, with the exception of Poland, has the lowest pride in that area.

In general, national pride seems to be greatest in stable, developed democracies and lowest in former communist states, especially in domains related to politics or the economy. All these findings can be verified using other statistical measurements, although the *support* remains small due to the relatively small number of items in each country (approx. 2.2% = 1000 items).

6.4.3 Influence of Age and Sex

Little evidence of any gender difference regarding pride can be found when comparing the various hierarchies, as no significant distinction is visible for any question. Before using the age dimension in the Parallel Hierarchies visualization, the inherently numerical dimension has had to be categorized. Using intervals as described in Section 3.3.2, six categories (15-27, 27-36, 36-45, 45-54, 54-65, 65+) with equal frequencies have been identified (i.e., each category consists of an equal number of items). The connections indicate a slight correlation between people *older than 65* and *very proud* category in most questions asked. Brushing has been used in Figure 6.8 to confirm this and has shown, that the relationship between age and pride is strongest in the domain of armed forces.



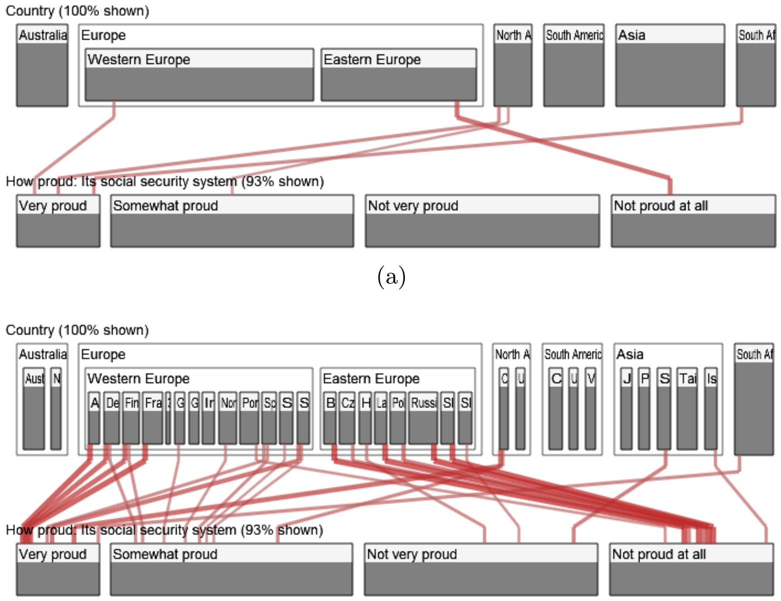
(a)

(b)

(c)

(d)

Figure 6.6: Starting at the highest level of the hierarchy (a) drill-down operations are used to reveal more and more details (b and c). In the bottom picture, Western Europe is shown in more detail and all other countries are hidden. Furthermore, brushing is used to compare the frequencies of *very proud* and *not (very) proud* regarding each country. All examples use the *degree of independence* (the deviation of confidence from expected confidence) as connection measurement and a threshold of 15%.



(a)

(b)

Figure 6.7: Comparing pride regarding the social welfare system already reveals a trend at a lower detail level by the *degree of independence*. In the top picture (a) the difference between Western and Eastern Europe is obvious and drilling down one level shows additional information about the distribution within these regions (b).

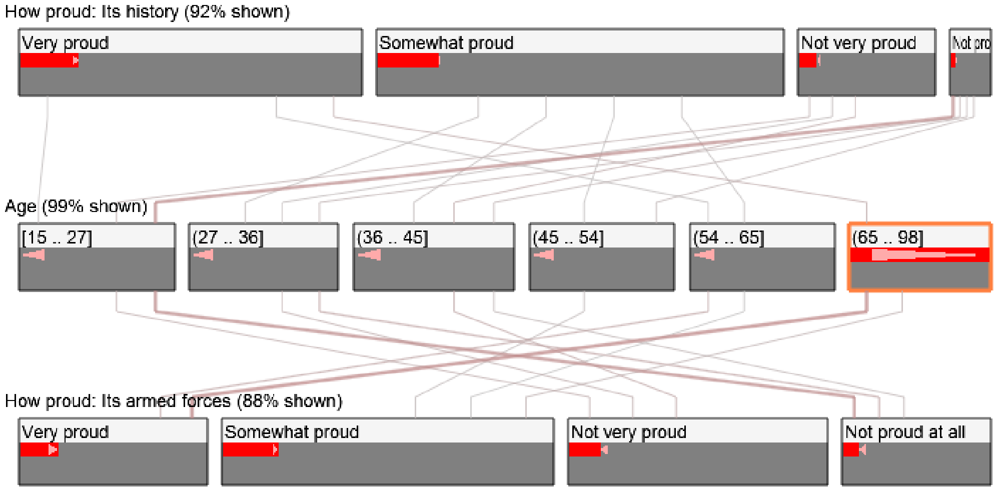


Figure 6.8: Using the (symmetrical) *lift* measurement to highlight patterns, a correlation between the *oldest age group* and *very proud of armed forces* can be found.

6.4.4 Influence of Education

Another interesting issue is the relationship between educational level and national pride. In order to analyze all domains of national pride in conjunction with other interesting dimensions of the survey on a single screen, the spacing between hierarchies has been reduced and association lines have been disabled. Brushing people with respect to their educational level in different layers has been used to find interesting patterns (Figure 6.9).

The first observation is, that more respondents (about 60%) have a higher education (*higher secondary completed, above higher secondary, university degree*). Similarly the lower education categories (*no formal qualification, lowest formal qualification, above lowest qualification*) are grouped. Brushing both groups reveals one interesting general pattern: the higher the education, the less proud people are in regard to specific achievements of their country. This relation is significant in the domains of sports and armed forces, and to a smaller degree also in the social welfare system. To a lesser extent, this trend surprisingly even holds for the fields of art, literature or science.

Concerning the additional variables which have been added to the view, the first one, "*general speaking, [country] is a better country than most others*" is a more general question to indicate national pride. Here, the influence of education on pride is even stronger, supporting the pattern found before. Furthermore, the visualization also shows that lower educated people generally feel closer to their country. They also rather tend to agree to the statement *immigrants increase crime rates.*

By getting rid of the association lines, a large number of hierarchies can be displayed simultaneously and the visualization has similarities to Parallel Trees (Section 2.1.3), where one active dimensions is linked with all others. Sifer states that this reduced visual complexity provides a significant advantage if more than three dimensions are shown [62]. Whereas Parallel Trees only display the data in this reduced form, Parallel Hierarchies allow the user to adapt the view according to the current task or the number of displayed dimensions. Additionally, this is another great example of how the expected frequency display improves the visualization. Without the arrows, only the distribution of the brushed data subsets could be observed, whereas the arrows highlight the patterns in a prominent way. This time-saving feature also enhances the user's understanding, as interesting deviations from the expected values are pointed out and easy to understand.

However, one has to be careful when using the deviation of expected frequency to actual frequency (displayed as arrows in the layer bars) to find patterns. In small data sets, slight variations in the data (e.g., caused by noise) may result in different results. In this case study, the used data set consists of over 44.000 items and even rather small deviations of 1% represent more than 400 occurences, making the significant influence of (random) variations rather unlikely. To prove this, two random subsets of the whole data set were analyzed using the same methods as above. Both times the results were similar to the patterns found before.

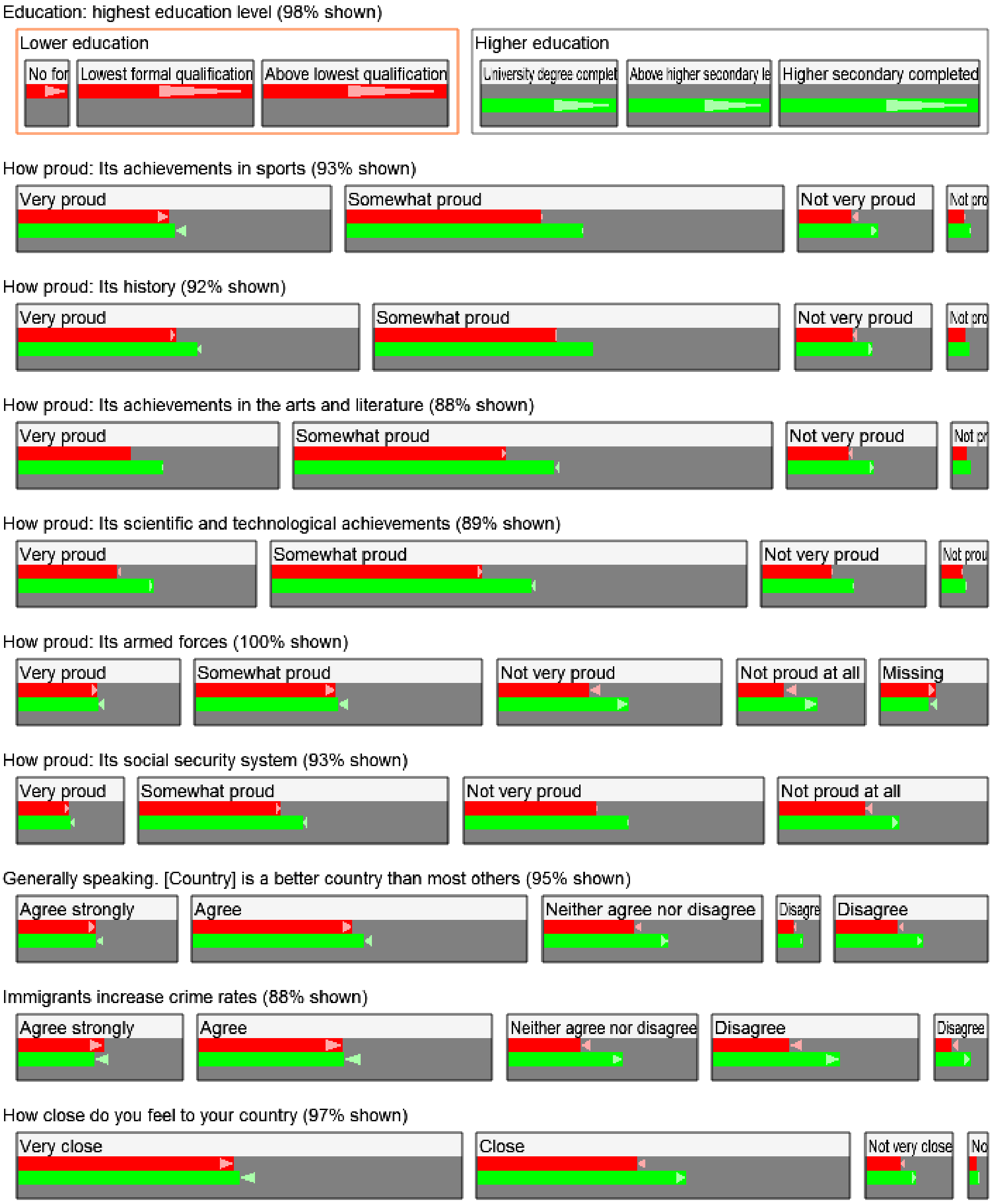


Figure 6.9: Brushing is used to visualize the influence of education on national pride, which reveals that lower educated people tend to be more proud. The arrows indicating the deviation of expected frequency to actual frequency allow for an efficient identification of patterns. For example, the red and green arrows in the *how close do you feel to your country* dimension (lowest in the picture) each represent approximately 400 respondents. Colors were disabled to highlight the red and green bars.

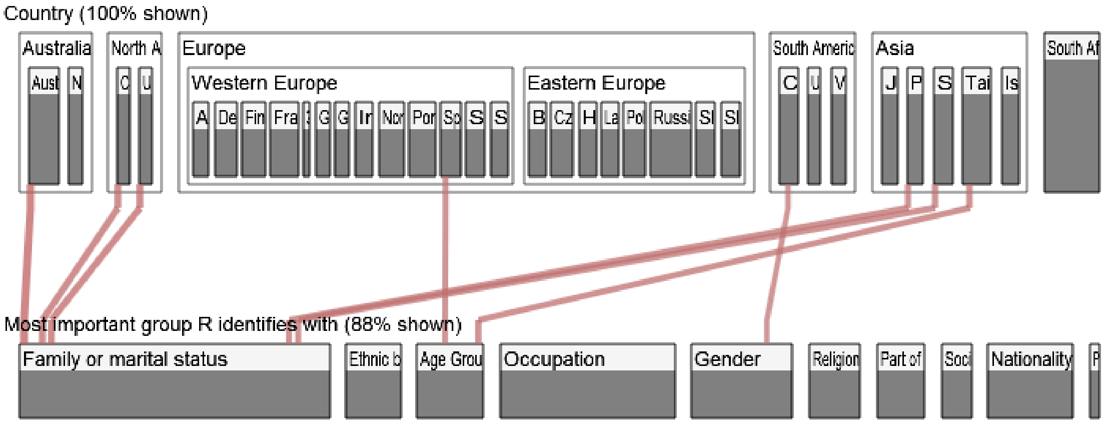


Figure 6.10: Family or martial status and occupation are the two most mentioned groups a person identifies with. Interesting are the relations of Spain and Taiwan with *Age Group* as well as Chile with *Gender*. In this example, the *degree of independence* measurement with a threshold of 15% has been used.

6.5 Identification

After discovering some interesting patterns concerning national pride, the next task was to analyze another main topic of the survey: identification (interviewees were asked about the most important group they identify with).

Figure 6.10 shows that *family or martial status*, *occupation* and *gender* are the three largest groups. Brushing by using the super focus has revealed that family was most often mentioned in Australia, North America and Western Europe, but is less important in Asia and South America. The association lines also show the relation of Australia and North America to the family category, but also reveal that, contrary to the continent Asia, South Korea and the Philippines are correlated with this category as well. Other interesting associations include the one between Spain, Taiwan and age group, and Chile with gender. A closer look into the gender group discloses that there is no evidence of gender difference. The number of women stating gender as the most important group they identify with is only diminutively larger than the number of men.

6.5.1 Influence of Age and Education

Using the Hierarchical Scatterplot, the relation of identification groups to age and education (Figure 6.11) can be visualized. Unlike before, the numerical dimension *education: years of schooling* is used, instead of the highest educational level. There is a direct correlation between the two dimensions dealing with education, i.e., the more years somebody spent in school, the higher the educational level is.

Several interesting observations can be made using the average aggregate for both numerical dimensions. First, the average age is, not surprisingly, lowest for *age group* and, quite surprisingly, *social class*. *Religion* on the other hand has the highest average with 48 years. Compared to most others, people identifying themselves with their age group also spent less years in schools, only *ethnic background* and *nationality* have an even lower average. On the other hand, people identifying themselves with their occupation have by far the largest average.

Interestingly, the difference in education is not as high as one may assume. The smallest and largest averages are only (about) a year apart. Switching from average to median aggregates reveals that all categories, except *ethnic background*, have an median of 12 years of schooling. The median of the category *ethnic background* is 11 years. Therefore, one must be careful to draw too quick conclusions from results of a single aggregate and it shows the importance of giving the user the possibility to switch between different aggregates.

In addition to these results, which could have been calculated using a traditional pivot table (Section 2.1.1, the Hierarchical Scatterplot facilitates the analysis by using several other visual clues. First, the size of each circle represents the number of people identifying with each group, allowing for a quick comparison of each group. It also helps to explain the influence of single identification groups on the overall results (the larger the category, the larger its influence on the overall result). Furthermore, the lines connecting each category glyph with the entire hierarchy's result, enhance the visual analysis of how each group deviates from the overall result. For example, long vertical lines imply, that the category is close to the overall value of the horizontal axis, but it departs clearly on the other one (e.g., *relgion* or *social class*). Together with the support for hierarchies, these *deviation vectors* set the Hierarchical Scatterplot apart from comparable techniques, like Gapminder's Trendanalyzer [21] and allow for an even more efficient and comprehensible visual analysis.

One question asked in the survey is "Do you agree that one benefit of the Internet is, that more information is available?". Using brushing and the Hierarchical Scatterplot, we can analyze how this dimension relates to the results found before. Two subsets of the data, all respondents that *agreed strongly* (36%) and the ones that did not agree (*neither agree or disagree, disagree, disagree strongly*; 12%), are brushed in a Parallel Hierarchies view. The influence of this question, as visualized in Figure 6.12, is substantial. As before, the lines connecting each subset with its category help trained users to identify general trends as well as single deviations.

People which agree strongly with the benefit of the Internet tend to be younger and better educated, independent of the group they identify with the most. The largest difference between the two subsets can be found in the category "age group". Overall, the average age difference between the two subsets is about 6 years. The average years of schooling deviation is about 2 years.

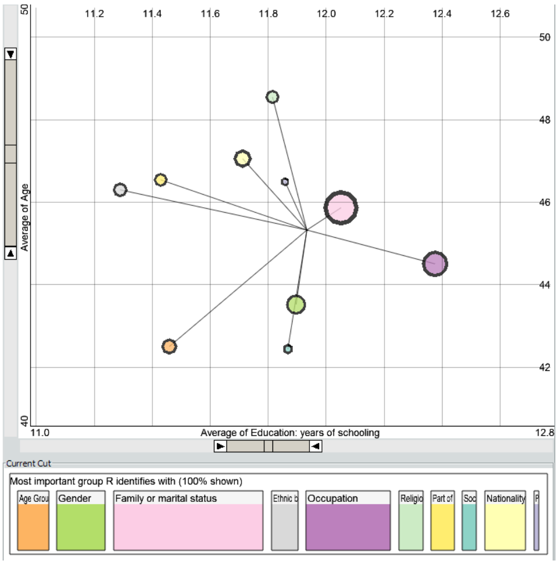


Figure 6.11: Using the Hierarchical Scatterplot different identification groups are visualized with respect to their average age and education. Additionally, the overall average can be compared to each category.

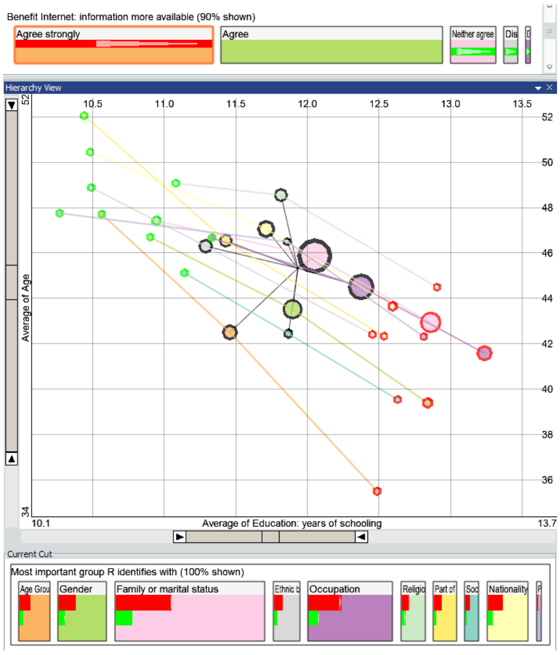


Figure 6.12: The deviation of two brushed subsets (*agree strongly to benefits of the Internet* and *do not agree*) to each other is visualized, revealing that people who *agree strongly* are younger and better educated.

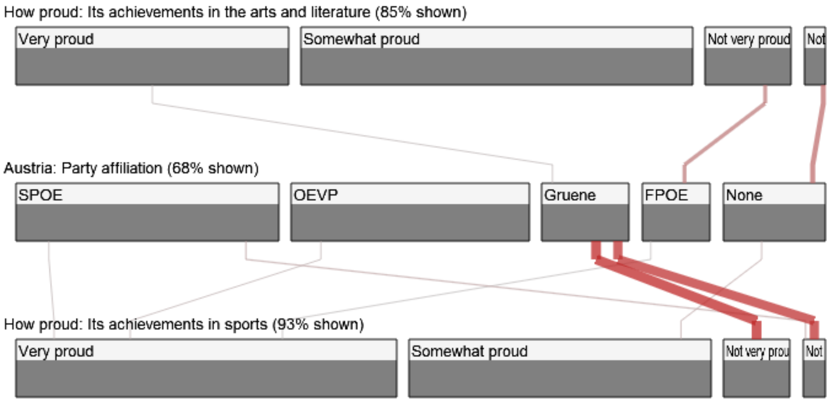


Figure 6.13: Using association lines and the *lift* measurement, reveals that people affiliated with the Green party are less proud of achievements in sports. FPÖ supporters, on the other hand tend to be less proud of achievements in arts and literature.

6.6 Closer Look at Austrian results

The next section focuses on a smaller subset of the entire survey, namely Austria specific variables (e.g., *party affiliation* and *regions*).

6.6.1 Pride and immigrants

Similar to the analysis of national pride in different countries (Section 6.4.1), relations and patterns in respect to the party affiliation have been searched. The *lift* measurement was chosen to discover associations in the Parallel Hierarchies visualization. Afterwards, *confidence* and *degree of independence* have been used to confirm any patterns found before.

Several interesting observations can be made during that process (see Figure 6.13). While respondents affiliated with the Green party are prouder of *achievements in arts and literature* than people affiliated with other parties (or people with no party affiliation), they are considerably less proud in the areas of *sports*, *history* and *armed forces* (the last two not shown in picture). FPÖ supporters on the other hand and people not affiliated with any party tend to be less proud of *achievements in arts and literature*.

As discovered before, Austria had the greatest pride in the social security system of all nations. Interestingly, the visualization shows that SPÖ supporters are less proud in this domain than others. A possible reason for this surprising result may be the fact, that, at the time the survey was conducted (2004), Austria was ruled by a ÖVP/FPÖ government and SPÖ supporters expressed their unhappiness with the political situation.

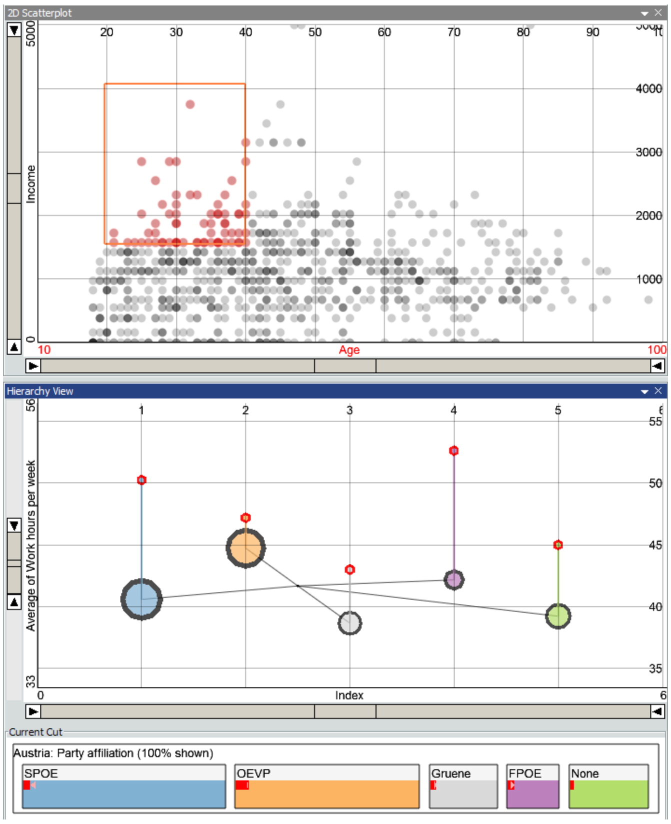


Figure 6.14: A 2D scatterplot has been used to select a subset of people (age: 20 - 40, income: above 1500 Euros). The Hierarchical Scatterplot illustrates the average working hours per week for each party as well as the deviation of the selected subset. Please note that the coloring of the parties is based on a general color scheme, although a more conventional color scheme (e.g., SPÖ = red, ÖVP = black, etc.) would be more appropriate here.

6.6.2 Income, Working Hours and Education

In Figure 6.14, the Hierarchical Scatterplot illustrates a significant deviation of working hours per week for people affiliated with different parties and shows how the *index* ”aggregate” can be used to equally distribute glyphs along one axis. People supporting the Green party have the lowest average working hours per week with about 38.5, while ÖVP supporters work almost 45 hours a week. In between are SPÖ supporters and FPÖ with 40.5 and 42 hours respectively.

People between the age of 20 and 40 who earn more than 1500 Euros a month have been selected in a 2D scatterplot view. This reveals additional information in the Hierarchical Scatterplot. First, the expected frequencies arrows in the cut display show, that there are less SPÖ supporters, but more FPÖ supporters in the selection than expected. Furthermore, the additional glyphs for the focus layer illustrate, that the average working hours per week of the selected people is considerably higher than the overall results. Whereas the deviation is only small for the category ÖVP, it is especially noticeable for SPÖ and FPÖ.

This also demonstrates, how beneficial the integration into an existing visual analysis system is. Using other views, like the 2D scatterplot, and the possibility to combine brushes by logical operations, the user can visually select complex subsets of the data. Comparable techniques (e.g., Parallel Sets or Gapminder) are often implemented as single-view visualizations and therefore, usually lack the flexibility of a complete InfoVis system.

The visual analysis illustrated in Figure 6.15 shows how a whisker plot can be used to augment the standard circles. In the shown example, Green party supporters seem to be considerably higher educated than affiliates of other parties. In fact, the lower end of the whisker plot of the Greens category is at the median position of FPÖ. ÖVP supporters have the second highest education (using the median as the main measurement for comparisons), followed by the categories of FPÖ and SPÖ. The example also depicts, how more complex glyphs, like the whisker plot, can be used to enhance the Hierarchical Scatterplot by providing providing additional information about the data distribution. Whereas novices probably prefer simple representations, the extra information can be very helpful especially for proficient users. The open design of the implementation allows for a simple integration of additional glyph types (as well as additional aggregates).

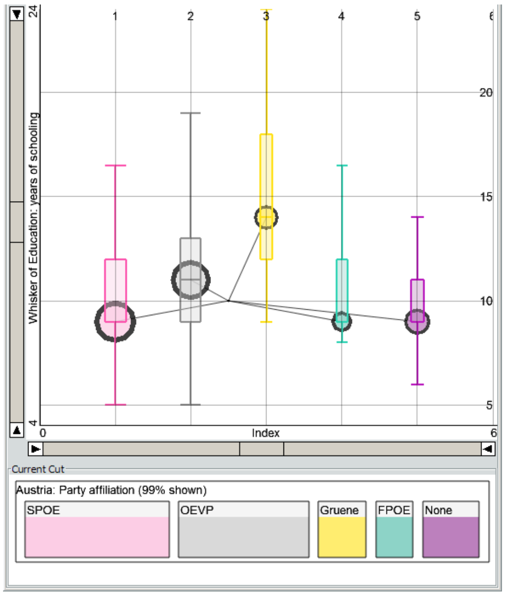

▼
24
1
2
3
4
5
6
20
15
10
5
Whisker of Education: years of schooling
▲
4
0
Index
6
Current Cut
Austria: Party affiliation (99% shown)
SPOE
OEVP
Gruene
FPOE
None

Figure 6.15: The whisker visualization shows that respondents affiliated with the Green party were higher educated than the rest.

Chapter 7

Implementation

Both visualization techniques, the Parallel Hierarchies and the Hierarchical Scatterplot, were designed as extensions for *Bulk Analyzer*, a visual analytics software currently in development at the VRVis Research Center [2] (see Section 2.4). The system is built on the basis of the *InfoVis Library* [51] and allows for the integration of various visualization techniques using a plug-in system. The InfoVis Library, the Bulk Analyzer, and its plug-ins are written in C++ and use the open source framework GTK+ [1] for the user interface and all operating system specific issues. Additionally, the views use OpenGL to render the visualizations at interactive frame rates. The Bulk Analyzer runs on Windows and Linux operating systems.

The following sections will provide some information about technical aspects of the Bulk Analyzer environment used to implement the previously described Parallel Hierarchies and Hierarchical Scatterplot visualizations.

7.1 InfoVis Library and Bulk Analyzer

As mentioned before, the InfoVis Library is an important part of Bulk Analyzer and all visualizations built upon it. The library was designed to provide a common framework for storing structured data for InfoVis applications and provides a set of means to work with this data. The InfoVis Library uses tables to structure the data similar to relational databases. A table consits of an arbitrary number of channels which represent the data attributes (*columns* in relational databases) and entries (*rows*), which represent the items of the table.

Apart from the InfoVis Library, the Bulk Analyzer provides a framework that allows the development of different types of visualizations. New kinds of views, application specific data importers and algorithms for data derivation can be added to the system by utilizing the plug-in mechanism offered by the system. The Bulk Analyzer core checks for available plug-ins at start-up and loads the extensions appropriately. Together with the InfoVis Library, the Bulk Analyzer core implements the entire *Data* part of the Information Visualization Pipeline. While the views are responsible for the mapping of data to visual structures, the Bulk Analyzer core also provides functionality to interactively control this transformation (e.g., by a data mapping widget).

7.2 View Structure

The Parallel Hierarchies and Hierarchical Scatterplot plug-ins, respectively, consist of three main parts: the *View*, the *Frontend*, and the *Backend*. The view class, which implements the `BAIView` interface, functions as the backbone, connecting the other parts and also receiving notifications from the Bulk Analyzer core. Moreover, the view holds a set of currently assigned hierarchies. The frontend provides the user interface including the controls. The backend on the other hand, is responsible for the actual view-logic, the drawing of the visual structures in the main window, and for processing user-interactions interactively. Figure 7.1 shows an overview of the structure, highlighting the main parts of the view.

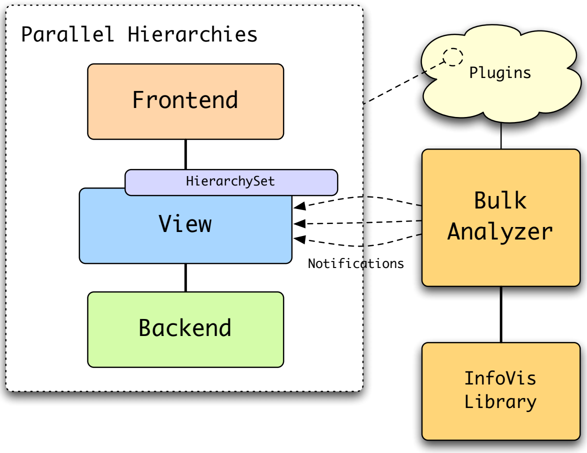


Figure 7.1: The main structure of the Parallel Hierarchies plug-in. The view connects the frontend and the backend. Furthermore, it receives notifications from the Bulk Analyzer core.

The view class can be considered the main class of the Parallel Hierarchies plug-in. Its main purpose is to receive notifications from the Bulk Analyzer core and forward them to the appropriate sub-parts (frontend and backend). Furthermore, the view handles almost all possible user actions.

The frontend is responsible for most things related to the user interface. GTK+ is used to layout the view, provide widgets such as scrollbars and create the view's controls, which are shown whenever a view is active (only one view can be active at any time, clicking inside the view's window activates it).

The backend implements the actual view-logic and handles the drawing for the Parallel Hierarchies visualization. Notifications, regarding user input or data changes, are forwarded by the view class and processed within the backend. In the Parallel Hierarchies visualization, the backend is responsible for computing the dimensions of each visual structure (i.e., the boxes for the hierarchies, categories and layers), conducting the calculations for the association measurements, drawing and updating the visual representations, as well as managing selections and brushing, to name a few tasks.

In order for the application to stay responsive at any time, which was the most challenging task when implementing the views, calculations are carried out in a separate thread, the background worker thread. That means, time- and computationally-intensive updates are not dealt with immediately when the notification arrives, but the notification thread invalidates the parts which have to be recomputed or updated. To further enhance the responsiveness of the views, the concept of *preview rendering*, which is considered in most Bulk Analyzer views, was implemented. As soon as the most important parts have been updated (e.g., the visual representations of the categories), a quick preview of the visualization is rendered. Later, the preview is replaced with the final image.

7.3 Performance Analysis

A performance analysis was carried out using *Intel VTune Performance Analyzer* [29] to analyze the time spent in specific areas of the source code. Additionally, the time spent to update different parts of the visualized data was measured internally. The performance was measured using three generic data sets ranging from 44.170 to 441.700 items. For each single analysis, five hierarchies, each with 26 dimensions, were created and assigned to the Parallel Hierarchies view. One of these hierarchies together with two numerical dimensions was also assigned to the Hierarchical Scatterplot. Afterward, all available association measurements and aggregates were calculated. A Intel Pentium 4 with 2.8 GHz, 1024 MB RAM and an ATI Radeon 9500 graphics card was used in the process.

As mentioned above, Bulk Analyzer uses several threads to carry out updates and calculations across the views. Table 7.1 shows three sample threads and compares the time spent in selected parts of the application. "Thread 1" is mainly responsible for creating the hierarchies and other general Bulk Analyzer functions. "Thread 2" carries out all the calculations needed for the Parallel Hierarchies view, while "Thread 3" does the same for the "Hierarchical Scatterplot". Other threads such as the notification thread are not listed in the table, because the amount of time used for them is negligible. Not surprisingly, "Thread 1" is remains idle most of the time (or carries out other jobs) for medium-sized data sets, but the time spent in the InfoVis Library (IVL) and the Bulk Analyzer Core functions increases with larger data sets. "Thread 2" shows a significant amount of time spent in the IVL as well. Whereas the calculations in the Parallel Hierarchies plug-in are finished quite quickly, most time is spent accessing the

			Dataset Size		
			44.170	176.680	441.700
Threads	Thread 1	InfoVis Library	16%	23%	23%
		BACore	8%	10%	13%
	Thread 2	InfoVis Library	58%	70%	80%
		ParallelHierarchies	24%	16%	8%
	Thread 3	InfoVis Library	49%	48%	45%
		StatLib	22%	32%	35%
		HierScatterplot	7%	4%	3%
Performance	Time to update connections		~ 1 sec	~ 1.5 sec	~ 3 sec
	Time to update aggregates		< 0.1 sec	< 0.3 sec	< 0.7 sec
	Time to update super focus		< 0.1 sec	< 0.1 sec	< 0.3 sec

Table 7.1: Analyzing the views using three different data set sizes shows a majority of time spent in the InfoVis Library and StatLib functions.

data. The analysis of "Thread 3" illustrates similar results. Even less time is spent in the actual Hierarchical Scatterplot source code, but most work is done accessing the data in the IVL and aggregating the items in the Statistics Library (StatLib).

Additionally, the time used to update different parts of the visualization was measured. For all data set sizes, the super focus was updated almost immediately with no visible lag. With the exception of more complex aggregates, the same can be said about the aggregate calculations. Median and Whisker aggregates take slightly longer to calculate (about 0.7 seconds for the largest data set). In the Parallel Hierarchies view, most time is spent when the user switches the active association measurement. For five hierarchies with 26 categories a total of 2704 connections have to be calculated (676 for each adjacent pair).

This analysis shows, that most time is spent accessing the data, while other calculations take only little time. In general, both visualizations scaled very well with the larger data sets and stayed responsive at any time.

Chapter 8

Summary

In the recent years, the progress of computer performance has led to vast amounts of data being generated and stored each day. However, extracting valuable information from large amounts of raw data is challenging. Visual analysis techniques present data in a visual form, consequently taking advantage of the human visual system and appropriate interaction facilities enable the user to explore and analyze the data. The techniques introduced in this work are specifically aimed toward the visualization of categorical data. Traditional item-based visualization techniques (e.g., scatterplots) do not work well for categorical data as they do not use the space efficiently or may even be not reasonably applicable at all, if the categories lack an inherent ordering. Since categorical data is also often structured in a hierarchical way, respective visualization techniques should consider this, too.

This work presents two approaches to visualizing hierarchically structured, categorical data.

8.1 Introduction

In general, visualization is the use of graphical representations to communicate a message [10]. In computer science, visualization is the process of transforming data into an image on the screen. It makes use of the human visual system and brain capabilities, greatly enhancing the detection of noteworthy subsets or patterns within the the data [10]. Even though they share to some extent common goals and techniques, the field of visualization is traditionally divided into two parts: Scientific Visualization (SciVis) and Information Visualization (InfoVis).

Information Visualization is usually used to visualize abstract, heterogeneous data. Whereas Scientific Visualization mostly deal with inherently spacial data (e.g., medical data, flow simulation data, etc.), abstract data does not have (or at least not entirely) an inherent mapping to space, which requires additional steps to map the data to the computer screen [24]. The focus of InfoVis often lies in the exploration of data, making a high degree of user interaction essential.

The approaches presented in this work focus on the visualization of hierarchically structured, categorical data.

Categorical Data is very common in real-world data sets. Basically, two kinds of data attributes can be distinguished: numerical (quantitative, continuous) or categorical (qualitative) attributes. Numerical data always has an inherent ordering and meaningful distances can be computed between any two values. Categorical data usually only has a limited number of distinct values and does not need to have inherent ordering, nor is the calculation of numerical differences possible in general. However, it is sometimes reasonable for the purpose of visualization to treat discrete data with only a few possible values as categorical data as well, although this data provides both ordering and distance measurements. The reason is that visualization techniques for categorical data (which are mostly frequency based) tend to utilize available screen space better for a small amount of distinct values than techniques for numerical data, which typically assume a continuous distribution of values.

The fact that the data may be lacking ordering and distance measures makes its visualization very challenging, because graphical representations may imply incorrect relationships between data entities.

Hierarchical Structures are often used to augment categorical data. The structure may be inherent in the data or generated (by the user or classification algorithms). One approach to create a hierarchy is to aggregate elements of lower levels, commonly used by OLAP applications [69]. Using hierarchies, the data can be viewed and analyzed from varying granularity levels, often greatly enhancing the user's understanding of the data set.

8.2 Related Work

Categorical data lends itself toward a discrete user model, a classification presented in a taxonomy proposed by Tory and Möller [72]. Most common visualization techniques, such as Parallel Coordinates [28] or scatterplot views, on the other hand, are based on a continuous design model. The discrepancy between the user expectation (a discrete model) and the presented image, is one of the reasons why traditional approaches are often not suitable for categorical data.

For categorical data it is more natural to use frequency-based techniques, which implement a discrete model [15]. Examples for frequency-based approaches include histograms [41], parallel bargrams [79], the Mosaic Display [15, 27], and TreeMaps [60, 77]. The techniques that were most influencing for the design of the Parallel Hierarchies visualization are Parallel Sets [7] and Parallel Trees [62].

Parallel Sets are based on the idea of Parallel Coordinates, and similarly represent data dimensions on a parallel axes layout. Each dimension is described by its categories, replacing the numeric axes of Parallel Coordinates by proportionally scaled boxes. To scale these boxes, the relative frequencies of the corresponding categories are calculated. Categories of adjacent dimensions are connected by parallelograms, which represent the relations between different categorical attributes of the data. Comparable to the boxes, the parallelograms are scaled by the frequency of the described data items, allowing the user to compare the width of the parallelograms to analyze the data relations.

Parallel Sets display all information provided by standard crosstabulations, but also allow the visualization of multiple dimensions simultaneously, making the examination of complex relationships and patterns possible. Additionally, the visualization can be extended to display statistical measurements such as the conditional probabilities or the *degree of independence* [7]. These statistics are displayed by histograms, drawn inside each category box. Furthermore, numerical data may be visualized as well, using binning or clustering to categorize the continuous data.

In general, Parallel Sets are highly interactive, allowing the user to rearrange dimensions and categories, highlight the relations between different categorical attributes of the data, or use dimension composition to reduce the number of dimensions.

Parallel Trees use a similar approach to Parallel Sets, but lack of any visualized connections between adjacent dimensions. As an alternative, Parallel Trees use one active dimension, set by the user, to implicitly link it with all others by coloring parts of the boxes. Therefore and contrary to Parallel Sets, which show relationships between all adjacent dimensions, Parallel Trees only show the relationship between the active dimension and all others.

The main feature of Parallel Trees is their support for hierarchical structures. Each hierarchy consists of at least three levels (a top level aggregating all items, at least one intermediate level and a bottom level displaying the highest possible detail) and all are shown simultaneously. The approach allows the selection and filtering of categories at any level, updating the other categories and hierarchies accordingly. Using these tools, the user can drill-down on data subsets or single categories.

Bulk Analyzer The visualizations presented in this thesis have been integrated into the Bulk Analyzer visual analytics system. Bulk Analyzer employs a multi view approach, allowing the display of various visualizations simultaneously. The views are bi-directionally linked and every user interaction (e.g., brushing) in one view updates all other views, a crucial feature for interactive data analysis [14]. Another important visualization concept included in Bulk Analyzer, is Focus+Context visualization, which allows the concurrent display of overview (context) and details (focus) [24].

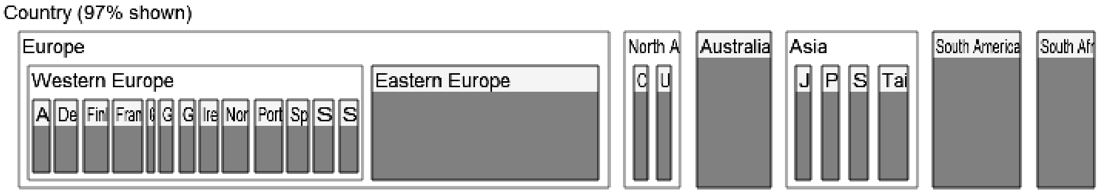


Figure 8.1: Hierarchical structures are displayed using nested boxes. The currently selected cut is visualized, while parent nodes are displayed by borders and labels.

8.3 Parallel Hierarchies

Data sets commonly used in real world applications often contain a large number of categorical dimensions. As pointed out above, traditional visualization techniques do not usually support categorical data well. Solutions geared toward the visualization of categorical data such as Parallel Sets or Parallel Trees, on the other hand, do not visualize continuous data as well. In general, a single visualization technique is unlikely to be suitable for all types of data. Therefore, one of the main motivations for the development of the presented approach was to integrate a categorical data visualization into an existing visual data analysis system (Bulk Analyzer) and provide the user with a wide range of linked visualization techniques.

Parallel Hierarchies builds on the idea of Parallel Sets, but emphasizes relationships and patterns between categories more. Furthermore, the visualization supports hierarchically structured data and all Bulk Analyzer features, such as interactive brushing across various views.

8.3.1 Visualization

Similar to Parallel Sets, a horizontal alignment is chosen to layout an arbitrary number of dimensions, which may be arranged by the user in any order. Adjacent boxes represent the categories of each dimension. The boxes are scaled according to the relative frequency of the respective category, where the full width of the view means 100% of the data. This allows for an easy comparison of categories since categories with a higher number of items are wider than categories with a smaller item count.

Parallel Hierarchies allows the user to examine the data at different levels of abstraction. Two operations, drill-down (moving down the hierarchy) and roll-up (moving up the hierarchy), are used to select a disjunctive and complete subset of nodes of the hierarchy, the cut. The cut represents all the data contained within the hierarchy and can be seen as a state of navigation. Figure 8.1 shows how the hierarchical structure of a dimension is represented in the visualization. Whereas the categories within the cut are displayed as described before, they are nested within their parent categories, which are displayed as scaled boxes as well.

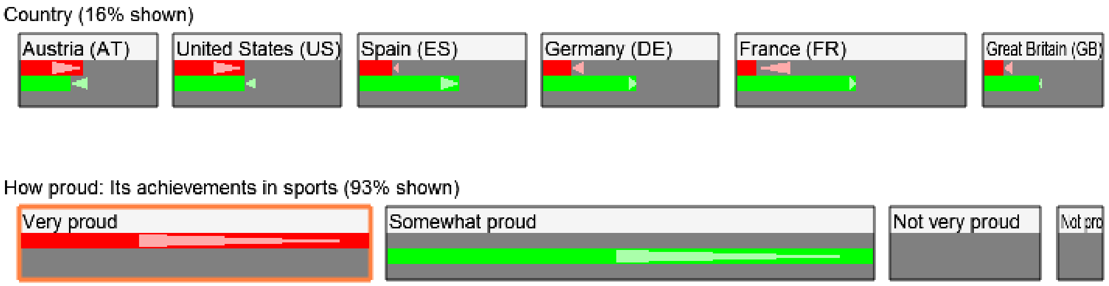


Figure 8.2: Arrows are used to visualize the discrepancy between actual and expected frequency, which can enhance the pattern finding process significantly. In the example the categories ”very proud” (red) and ”somewhat proud” (green) were brushed, revealing their distribution in the above ”Country” dimension.

The Bulk Analyzer system allows for the brushing of multiple individual selections, which are linked with all other views. Therefore, the integration of the layers in the visualization is not only crucial to share information across different views, but also a very powerful tool to explore multiple dimensions at once. To passively integrate the layers in the visualization, individual bars are displayed within each category box of the cut. These bars are scaled similar to the box itself, i.e., if half of the entries of the category are selected, the associated bar will fill half of the category.

To simplify the detection of unexpected relationships, Parallel Hierarchies optionally displays the expected frequencies and their deviation from the actual values for each layer visually (see Figure 8.2. In general, the expected frequency reveals information about data distribution without any prior knowledge about the data set. For example, given a data set of 1000 items, a subset X with 100 items and a subset Y with 250 items, the expected frequency of $X \cap Y$ is 25. For every category and every layer one expected frequency and its deviation to the actual frequency is calculated. The deviation has proved to be of very high interest, since it expresses to which degree a category is over- or under- proportionally related to the selection criterion underlying each layer and it facilitates the perception of relationships between layers and categories significantly. Both results are visualized by a half-transparent arrow, which has its base at the expected value and its top at the actual value.

8.3.2 Displaying Trends and Relations

To further enhance the detection of patterns, the Parallel Hierarchies approach displays trends and relations between categories by visualizing interest measurements, a commonly used technique in the field of association rule mining [3]. The results of these interest measurements are visually represented by lines connecting the categories of adjacent hierarchies, displaying positive (red lines) as well as negative (blue lines) correlations and conveying an approximate impression of how strongly categories or hierarchies are related (thickness of the lines and saturation of the color; see Figure 8.3).

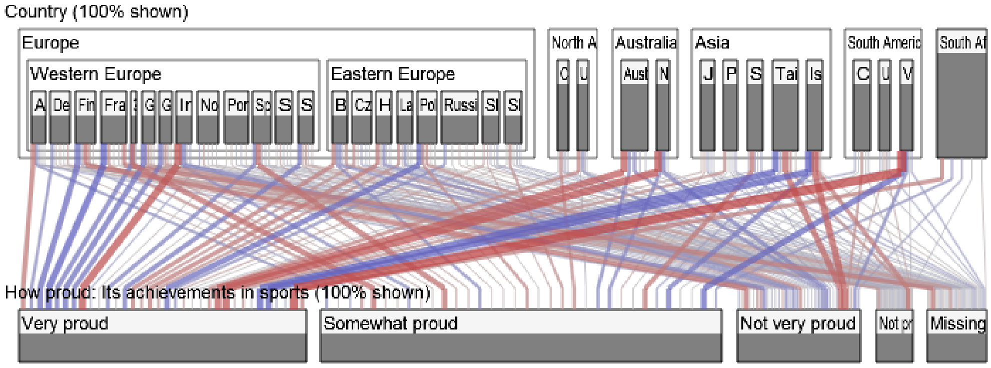


Figure 8.3: The "degree of interest" measurement reveals some interesting relationships between the adjacent hierarchies "Country" and "How proud: Its achievements in sports".

A number of commonly used interest measurements are available in the Parallel Hierarchies view:

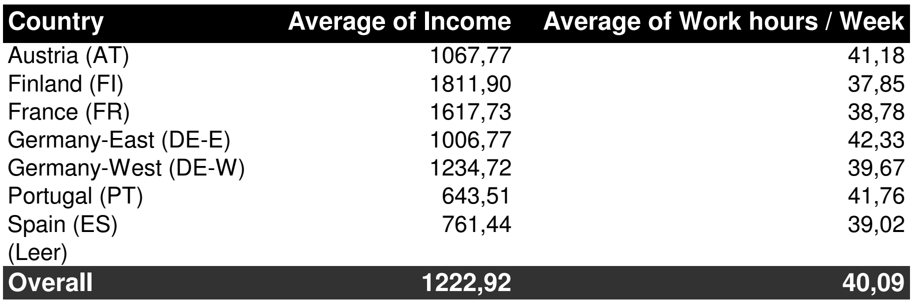
- Support: the relative frequency of the conjunction of two categories X and Y.
- Confidence: the ratio of the relative frequency of $X \cap Y$ to the relative frequency of X.
- Lift: the ratio of confidence to expected confidence.
- Degree of Independence: the deviation of confidence from expected confidence.

The user may change the active measurement at any time. Furthermore, a threshold can be set to hide uninteresting lines, strongly improving the perception of the more distinct relationships.

8.4 Aggregate-based Hierarchical Scatterplot

As already mentioned above, most real-world data sets are not exclusively numerical nor are they strictly categorical. Numerical data may be categorized using intervals and displayed using the Parallel Hierarchies visualization, but even slight changes in the interval borders may affect the results significantly, resulting in the lose of numerical data characteristics (e.g., the possibility to calculate the distance between two values) in the process.

The technique proposed in this section is a novel approach to visualize a single hierarchy with respect to two numerical dimensions, supporting both data types directly. Categories are represented by glyphs and placed on a scatterplot-like view with two numerical axes using numerical aggregates. Numerical aggregates are a simple yet powerful



Country	Average of Income	Average of Work hours / Week
Austria (AT)	1067,77	41,18
Finland (FI)	1811,90	37,85
France (FR)	1617,73	38,78
Germany-East (DE-E)	1006,77	42,33
Germany-West (DE-W)	1234,72	39,67
Portugal (PT)	643,51	41,76
Spain (ES)	761,44	39,02
(Leer)		
Overall	**1222,92**	**40,09**

Figure 8.4: Using a pivot table to display the averages of two continuous (income, work hours per week) dimensions in regard to one categorical dimension (countries).

way to use numerical data in conjunction with a categorical dimension. Figure 8.4 illustrates how aggregates are used in a pivot table to calculate the average of two numerical dimensions using the categories defined by the dimension ”Country”.

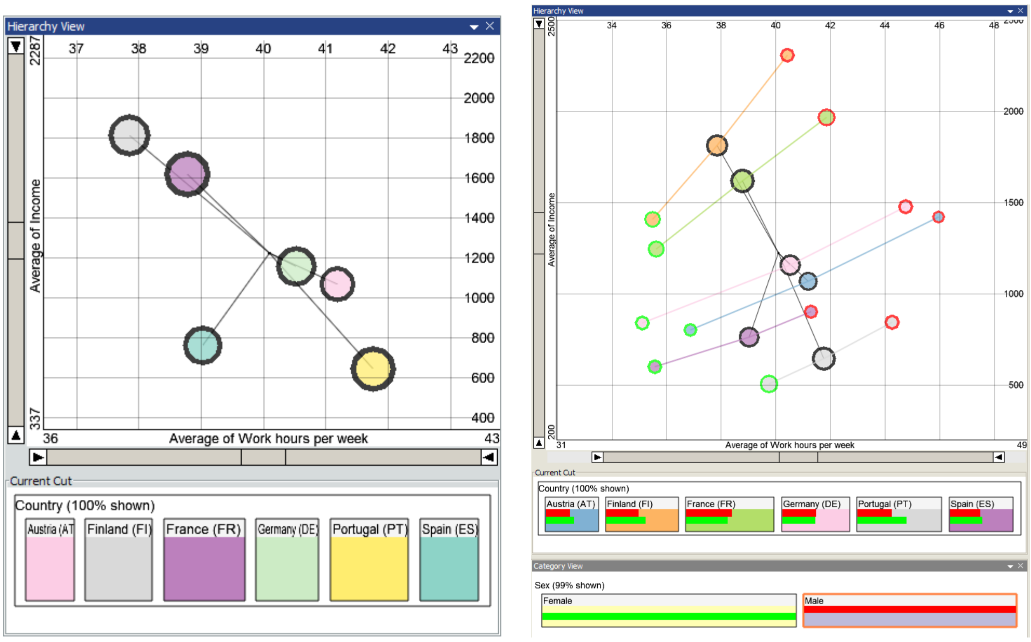
Similar to the Parallel Hierarchies visualization, the Hierarchical Scatterplot supports hierarchically structured data and fits within the Bulk Analyzer framework.

8.4.1 Visualization

Figure 8.5a shows how the information from the pivot table is displayed in the Hierarchical Scatterplot. Each category of the assigned hierarchy's cut (see the previous section) is represented by a glyph (a filled circle). By default, glyphs are scaled by the frequency of their associated category, i.e., the more entries a category has, the larger the size of the respective glyph of the category.

Additionally, the hierarchy is displayed in a separate part of the view below the scatterplot, similar to the visualization in the Parallel Hierarchies view. This allows the user to navigate the hierarchy using drill-down and roll-up operations, but also provides additional information, such as the actual and expected frequencies of brushed subsets. Categories use the same colors in both parts of the view, which is crucial for the user to match categories efficiently.

By default, only the categories of the hierarchy's cut are displayed in the scatterplot as glyphs and the hierarchical structure is only visible in the lower part of the view. Optionally, the structure can be visualized in the scatterplot as well. Parent categories are visualized by black dots, which are placed according to their respective aggregates and connected by lines to their children. Different line widths are used to indicate various levels of the hierarchy. In the example (Figure 8.5a), the lines connect each glyph with the dot representing the entire hierarchy, making it possible to compare the individual aggregates to the overall results and often provide interesting information about patterns in the data set.



(a) (b)

Figure 8.5: The left picture (a) how the data from the pivot table is visualized using the Hierarchical Scatterplot approach. Using a Parallel Hierarchies view, female (green) and male (red) subsets were brushed using Bulk Analyzer's focus and context system in picture (b). The Hierarchical Scatterplot then shows how these subsets affect each category by positioning additional glyphs.

8.4.2 Aggregates

The Hierarchical Scatterplot provides various aggregates that can be chosen independently for each axis: average, median, sum, maximum, minimum, whisker, index and count. Apart from information for positioning of the glyphs, the "whisker" aggregate calculates all values needed to draw a box plot, which replaces the filled circle used for all other aggregates. On the other hand, the "index" and "count" aggregates do not aggregate any data and are therefore independent of any numerical dimension. The "index" attribute is used to equally distribute glyphs on one axis, while the "count" aggregate places the glyphs according to the item count of the respective category.

8.4.3 Layer Integration

The lower part of the view integrates the layers of the Bulk Analyzer system like the Parallel Hierarchies view and also highlights the difference between actual and expected frequencies. The scatterplot section displays the subsets defined by the layers as glyphs, thereby stressing the deviation of the brushed subsets from the entire categories with respect to the chosen aggregates. Placement and visual presentation of these additional glyphs is similar to the original glyphs, which represent the categories. To distinguish the different layers, the border of each additional glyph is colored using the layer's globally defined color. Figure 8.5b demonstrates how, using this approach, brushing can gain additional information. In another view the categories ”Male” and ”Female” were selected, allowing the user to analyze how male and female results vary in regard to the numerical dimensions and the active aggregates.

Apart from this passive integration of the layers, the user may also brush categories directly in the Hierarchical Scatterplot view. By clicking on glyphs or categories, the category will be selected in the focus layer. Additionally, multiple categories can be selected at once by drawing a selection rectangle. To quickly highlight one category, the user can point the mouse cursor over any glyph.

8.5 Conclusion

Although categorical data is very common in the field of visual data analysis, the number of visualization techniques geared toward this kind of data is still quite limited. In this work, two techniques to visualize hierarchically structured, categorical data were presented. The first one, Parallel Hierarchies, builds upon the intuitive approach of Parallel Sets, visualizing multiple hierarchies at once by using the item count of each category to scale its visual representation. Common tasks in InfoVis are to find interesting patterns and unexpected relationships within data sets, and Parallel Hierarchies supports the user by highlighting trends between categories and hierarchies. The second proposed technique, the Hierarchical Scatterplot, presents a novel approach to visualize the categories of a single hierarchy in regard to two numerical dimensions. Both approaches supplement each other very well. Furthermore, the integration into an existing visual analytics system allows the techniques to be used in a wide variety of tasks.

Chapter 9

Conclusions and Future Work

Considering some of the most common visualization techniques, it is apparent that most are not well adapted to categorical data. Surprisingly, only limited research on the subject of categorical data visualization has been carried out in recent years and techniques often tend to be more for special purposes. In this thesis, two novel visualization techniques, designed specifically for categorical data, were described. The presented techniques also greatly benefit from the hierarchical structuring of categorical dimensions. By navigating through the hierarchy, the user can view the data at different levels of abstraction, which has proven to be especially useful for data dimensions with a large number of categories. Both visualizations complement each other well. While the Parallel Hierarchies view is used to display associations and patterns between hierarchies, the Hierarchical Scatterplot provides the user with a detailed view of a single hierarchy and its structure. Furthermore, the integration of the techniques into the Bulk Analyzer framework allows them to be used in a wide variety of problems, as other views may be used to support the analysis as well.

Although the case study has shown that the presented visualizations work very well, there are a few outstanding features which could further enhance their usability. Currently, both techniques are very useful to explore data sets and find patterns, but lack a display of the exact measurements calculated internally. The pop-ups provide some additional information, but may not be sufficient for cases where detailed results are needed. Additionally, the Parallel Hierarchies visualization does not provide the option to brush or select lines which connect adjacent categories. This feature could be used to extract interesting associations and use them outside the visualization, for example, to make predictions based upon the results. In general, untrained users may not be aware of the fact, that some association measurements (confidence and degree of independence) are asymmetric measurements and the visualization must be read top-to-bottom. Therefore, future implementations could highlight the direction, in which the lines are interpreted and allow the user switch the orientation interactively. It would also be worthwhile to see if an automated process to extract association rules, similar to approaches from data mining, would further enhance the user's experience. Such an approach may calculate a number of association measurements, combine the results in a meaningful way and highlight the most promising associations or patterns in the visualization. Currently, the Hierarchical Scatterplot scales the size of the displayed glyphs according to the item count of the respective category or, optionally, its depth in the hierarchy. Future versions, however, could give the user the possibility to assign any numerical dimension to represent the glyph sizes.

In my opinion, this work has shown that the collaboration of statistics and Information Visualization can be very beneficial. Statistics help to decrease the work for the user by allowing him or her to focus on the most noteworthy subsets of the data. Visualization makes use of the human visual system and the domain knowledge of the user. Combining both fields may be challenging, but it is worth the effort for an improved experience. I think that interactively combining the statistical capabilities of the computer with the pattern recognition capabilities of the human is and will continue to be one of the most promising fields of visual data analysis.

While working on this thesis, I have learned how important external feedback is. For example, when showing the Parallel Hierarchies view to users, it turned out, that the default coloring of the graphical representations confused most users. Similarly, one of the most useful feature of the Hierarchical Scatterplot, the display of vectors connecting each category with its father node in the hierarchy, was not included in the initial design, but was integrated after first experiments with the view.

During the course of my studies in this field, I have recognized the interdisciplinary benefits of visualization. For example, easy-to-use software motivates users to experiment with the tools, making visualization an entertaining, yet very valuable educational tool. Here it is even more important to guide users with appropriate visual clues, as complex and large data can easily overwhelm novices, as well as learned users.

In general, I think that a good user interface design will play an increasingly important role in Information Visualization. Whereas today, most research in this field, including this work, focuses on the graphical representation of data, in the future, usability will be even more important to efficiently work with ever increasing amounts of data. Therefore, I believe that efficient ways to interact with the data, together with the integration of statistics, will be fundamental for future approaches.

Chapter 10

Acknowledgments

This work has been done at the VRVis Research Center in Vienna, Austria, and was funded by an Austrian research program called Kplus.

Special thanks are given to Harald Piringer, who gave me the opportunity to work on this interesting project and supported me at all times. His ideas and his vision for the visualization of categorical and hierarchical data were crucial contributions to this work.

I also want to thank Helwig Hauser for his steady support and assistance. He also introduced me to the challenging field of Information Visualization at his excellent classes at the Vienna University of Technology.

Additional thanks have to go to Wolfgang Berger, former colleague in school and at the university, now colleague at the VRVis and mainly long-time friend for all his support in the last years.

I would also like to thank Denise Deutschlander for proof-reading this work and everything else in the last three years.

Bibliography

[1] GTK+ The GIMP Toolkit. `http://www.gtk.org/`, 2007.

[2] VRVis Research Center. `http://www.vrvis.at/`, 2007.

[3] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Mining association rules between sets of items in large databases. In *SIGMOD '93: Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216, New York, NY, USA, 1993. ACM Press.

[4] Todd Barlow and Padraic Neville. A Comparison of 2-D Visualizations of Hierarchies. In *INFOVIS '01: Proceedings of the 2001 IEEE Symposium on Information Visualization*, page 131, Washington, DC, USA, 2001. IEEE Computer Society.

[5] Jeff Beddow. Shape coding of multidimensional data on a microcomputer display. In *VIS '90: Proceedings of the 1st IEEE conference on Visualization '90*, pages 238–246, Los Alamitos, CA, USA, 1990. IEEE Computer Society Press.

[6] Fabian Bendix. Visual Analysis of Categorical Data - Parallel Sets. Master's thesis, University of Technology Vienna, 2004.

[7] Fabian Bendix, Robert Kosara, and Helwig Hauser. Parallel Sets: Visual Analysis of Categorical Data. In *INFOVIS '05: Proceedings of the 2005 IEEE Symposium on Information Visualization*, pages 133–140, 23-25 Oct. 2005.

[8] Cindy Brewer. ColorBrewer. `http://www.colorbrewer.org/`, 2002.

[9] Sergey Brin, Rajeev Motwani, Jeffrey D. Ullman, and Shalom Tsur. Dynamic itemset counting and implication rules for market basket data. In *SIGMOD '97: Proceedings of the 1997 ACM SIGMOD international conference on Management of data*, pages 255–264, New York, NY, USA, 1997. ACM Press.

[10] Stuart Card, Jock Mackinlay, and Ben Shneiderman. *Readings in information visualization: using vision to think.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

[11] Herman Chernoff. The Use of Faces to Represent Points in k-Dimensional Space Graphically. *Journal of the American Statistical Association*, 68(342):361-368, 1973.

[12] William S. Cleveland. *Visualizing Data.* Hobart Press, 1993.

[13] Edgar F. Codd. Providing OLAP (On-line Analytical Processing) to User-Analysts. `http://dev.hyperion.com/resource_library/white_papers/providing_olap_to_user_analysts.pdf/`, 1993.

[14] Helmut Doleisch, Martin Gasser, and Helwig Hauser. Interactive feature specification for focus+context visualization of complex simulation data. In *VISSYM '03: Proceedings of the Symposium on Visualization*, pages 239–248, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[15] Michael Friendly. Extending mosaic displays: Marginal, partial, and conditional views of categorical data. *Journal of Computational and Graphical Statistics*, 8:373–385, 1998.

[16] Michael Friendly. *Visualizing Categorical Data.* SAS Publishing, 2000.

[17] Michael Friendly. Visualizing categorical data: Data, stories and pictures. In *SAS User Group International Conference Proceedings*, pages 190–200, 2000.

[18] Ying-Huey Fua, Matthew O. Ward, and Elke A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *VIS '99: Proceedings of the 1999 IEEE Conference on Visualization*, pages 43–508, 24-29 Oct. 1999.

[19] Ying-Huey Fua, Matthew O. Ward, and Elke A. Rundensteiner. Navigating hierarchies with structure-based brushes. In *INFOVIS '99: Proceedings of the 1999 IEEE Symposium on Information Visualization*, pages 58–64,146, 24-29 Oct. 1999.

[20] George W. Furnas. Generalized fisheye views. pages 16–23, 1986.

[21] Gapminder Foundation. Gapminder. `http://www.gapminder.org/`, 2007.

[22] Michael Hahsler. A Comparison of Commonly Used Interest Measures for Association Rules. `http://wwwai.wu-wien.ac.at/~hahsler/research/association_rules/measures.html`, 2006.

[23] Marc D. Hansen. A Survey of Systems in the Diagrammatic Visual Data Querying Domain. Technical report, UCSC, 2005.

[24] Helwig Hauser. Generalizing Focus+Context Visualization. In *Dagstuhl Seminar 03231: Scientific Visualization: Extracting Information and Knowledge from Scientific Data Sets, also available as VRVis Technical Report TR-VRVis-2003-037*, 2003.

[25] Helwig Hauser and Robert Kosara. Interactive Analysis of High-Dimensional Data Using Visualization. In *Workshop on Robustness for High-dimensional Data (RobHD 2004)*, Vorau, Austria, 2004.

[26] Helwig Hauser, Florian Ledermann, and Helmut Doleisch. Angular brushing of extended parallel coordinates. In *INFOVIS '02: Proceedings of the 2002 IEEE Symposium on Information Visualization*, pages 127–130, 28-29 Oct. 2002.

[27] Heike Hofmann. Exploring categorical data: interactive mosaic plots. *Metrika*, 51(1):11–26, 2000.

[28] Alfred Inselberg and Bernhard Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *VIS '90: Proceedings of the 1st IEEE conference on Visualization '90*, pages 361–378, 23-26 Oct. 1990.

[29] Intel Corporation. Intel VTune. `http://www.intel.com/cd/software/products/asmo-na/eng/vtune/239144.htm`, 2007.

[30] Inxight Software, Inc. Inxight TableLens. `http://www.inxight.com/products/sdks/tl/`, 2007.

[31] Jimmy Johansson, Patric Ljung, Mikael Jern, and Matthew Cooper. Revealing Structure within Clustered Parallel Coordinates Displays. In *INFOVIS '05: Proceedings of the 2005 IEEE Symposium on Information Visualization*, page 17, Washington, DC, USA, 2005. IEEE Computer Society.

[32] Daniel A. Keim. Visual Techniques for Exploring Databases. In *KDD '97: International Conference on Knowledge Discovery in Databases*, pages 1–121, 1997.

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

[34] Daniel A. Keim. Scaling Visual Analytics to Very Large Data Sets. In *Workshop on Visual Analytics*, 2005.

[35] Daniel A. Keim, Ming C. Hao, and Umeshwar Dayal. Hierarchical Pixel Bar Charts. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):255–269, 2002.

[36] Daniel A. Keim, Florian Mansmann, Jorn Schneidewind, and Hartmut Ziegler. Challenges in Visual Data Analysis. In *INFOVIS '06: Proceedings of the 2006 IEEE Symposium on Information Visualization*, pages 9–16, Washington, DC, USA, 2006. IEEE Computer Society.

[37] Daniel A. Keim, Wolfgang Müller, and Heidrun Schumann. Visual Data Mining. In *EUROGRAPHICS 2002*, Saarbruecken, Germany, 2002.

[38] Alfred Kobsa. An empirical comparison of three commercial information visualization systems. In *INFOVIS '01: Proceedings of the 2001 IEEE Symposium on Information Visualization*, pages 123–130, 22-23 October 2001.

[39] Alfred Kobsa. User Experiments with Tree Visualization Systems. In *INFOVIS '04: Proceedings of the 2004 IEEE Symposium on Information Visualization*, pages 9–16, Washington, DC, USA, 2004. IEEE Computer Society.

[40] Erica Kolatch and Beth Weinstein. Cattrees: Dynamic visualization of categorical data using treemaps. `http://www.cs.umd.edu/class/spring2001/cmsc838b/Project/Kolatch_Weinstein/`, 2001.

[41] Robert Kosara, Fabian Bendix, and Helwig Hauser. TimeHistograms for Large, Time-Dependent Data. In *VISSYM '04: Proceedings of the Symposium on Visualization*, pages 45–54, 2004.

[42] Michael D. Lee, Rachel E. Reilly, and Marcus E. Butavicius. An empirical evaluation of chernoff faces, star glyphs, and spatial visualizations for binary data. In *APVis '03: Proceedings of the Asia-Pacific symposium on Information Visualisation*, pages 1–10, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc.

[43] Sheng Ma and Joseph L. Hellerstein. Ordering categorical data to improve visualization. In *INFOVIS '99: Proceedings of the 1999 IEEE Symposium on Information Visualization*, pages 15–18, 1999.

[44] Hari Mailvaganam. Introduction to OLAP. `http://www.dwreview.com/OLAP/Introduction_OLAP.html`, 2007.

[45] Jeremy Manson. Occlusion in Two-Dimensional Displays: Visualization of Meta-Data. Technical report, University of Maryland, College Park, `http://www.cs.umd.edu/hcil/academics/courses/fall1999/cmsc838s/Project/jmanson/`, 1999.

[46] D.L. Massart, Johanna Smeyers-Verbeke, Xavier Capron, and Karin Schlesier. Visual Presentation of Data by Means of Box Plots. *Practical Data Handling, LCGC Europe*, 18(4):215–218, 2005.

[47] Riadh Ben Messaoud, Omar Boussaid, and Sabine Loudcher Rabaseda. Mining Association Rules in OLAP Cubes. In *Innovations in Information Technology*, pages 1–5, Nov. 2006.

[48] Gordon E. Moore. Cramming More Components Onto Integrated Circuits. *Proceedings of the IEEE*, 86(1):82–85, 1998.

[49] OLAP Council. OLAP Council Whitepaper. `http://www.olapcouncil.org/research/whtpaply.htm/`, 2007.

[50] Ronald M. Pickett and Georges G. Grinstein. Iconographic Displays For Visualizing Multidimensional Data. In *SMC '88: Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics*, volume 1, pages 514–519, August 8-12, 1988.

[51] Harald Piringer. Design Guidelines and Concepts of the InfoVis Library. Technical Report TR-VRVis-2005-034, VRVis, 2005.

[52] D. J. Power. A Brief History of Spreadsheets. `http://dssresources.com/history/sshistory.html`, 2004.

[53] International Social Survey Programme. National Identity II. `http://zacat.gesis.org/webview/index.jsp?object=http://zacat.gesis.org/obj/fStudy/ZA3910`, 2003.

[54] Ramana Rao and Stuart K. Card. The Table Lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 318–322, New York, NY, USA, 1994. ACM Press.

[55] Theresa-Marie Rhyne, Melanie Tory, Tamara Munzner, Matt Ward, Chris Johnson, and David H. Laidlaw. Information and Scientific Visualization: Separate but Equal or Happy Together at Last. In *VIS '03: Proceedings of the 2003 IEEE Conference on Visualization*, page 115, Washington, DC, USA, 2003. IEEE Computer Society.

[56] Patrick Riehmann, Manfred Hanfler, and Bernd Froehlich. Interactive Sankey Diagrams. In *INFOVIS '05: Proceedings of the 2005 IEEE Symposium on Information Visualization*, pages 233–240, Washington, DC, USA, 2005. IEEE Computer Society.

[57] Thomas Rongitsch. Information Visualization and Data Mining - A Comparison and Integration. Master's thesis, University of Technology Vienna, 2005.

[58] Geraldine E. Rosario, Elke A. Rundensteiner, David C. Brown, and Matthew O. Ward. Mapping nominal values to numbers for effective visualization. In *INFOVIS '03: Proceedings of the 2003 IEEE Symposium on Information Visualization*, pages 113–120, 19-21 Oct. 2003.

[59] Tobias Schreck, Daniel A. Keim, and Florian Mansmann. Regular TreeMap Layouts for Visual Analysis of Hierarchical Data. In *SCCG '06: Spring Conference on Computer Graphics*. Casta Papiernicka, Slovak Republic, ACM Siggraph, 2006.

[60] Ben Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Trans. Graph.*, 11(1):92–99, 1992.

[61] 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, 3-6 Sept. 1996.

[62] Mark Sifer. User interfaces for the exploration of hierarchical multi-dimensional data. In *VAST '06: Proceedings of the 2006 IEEE Symposium On Visual Analytics Science And Technology*, pages 175–182, 2006.

[63] Panopticon Software. Panopticon - What is a Tree Map? `http://www.panopticon.com/panopticon/Content?id=268`, 2007.

[64] Michael Spenke and Christian Beilken. Visualization of trees as highly compressed tables with InfoZoom. In *INFOVIS '03: Proceedings of the 2003 IEEE Symposium on Information Visualization*, pages 122–123, 2003.

[65] Spotfire Inc. Spotfire. `http://spotfire.com/`, 2007.

[66] SPSS Inc. SPSS. `http://www.spss.com`, 2007.

[67] StatSoft Inc. Electronic Statistics Textbook. `http://www.statsoft.com/textbook/stathome.html`, 2007.

[68] Chris Stolte, Diane Tang, and Pat Hanrahan. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):52–65, 2002.

[69] Chris Stolte, Diane Tang, and Pat Hanrahan. Query, analysis, and visualization of hierarchically structured data using polaris. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 112–122, New York, NY, USA, 2002. ACM Press.

[70] Tableau Software Inc. Tableau Software. `http://www.tableausoftware.com/`, 2007.

[71] Kurt Thearling. Whitepaper, an introduction to data mining: Discovering hidden value in your data warehouse. `http://www.thearling.com/text/dmwhite/dmwhite.htm`, 2007.

[72] Melanie Tory and Torsten Möller. Rethinking Visualization: A High-Level Taxonomy. In *INFOVIS '04: Proceedings of the 2004 IEEE Symposium on Information Visualization*, pages 151–158, 10-12 Oct. 2004.

[73] Edward R. Tufte. *The visual display of quantitative information.* Graphics Press, Cheshire, CT, USA, 1986.

[74] Svetlana Vinnik and Florian Mansmann. From Analysis to Interactive Exploration: Building Visual Hierarchies from OLAP Cubes. In *EDBT 2006: Proceedings of 10th International Conference on Extending Database Technology*, pages 496–514, 2006.

[75] Matthew O. Ward. XmdvTool: integrating multiple methods for visualizing multivariate data. In *VIS '94: Proceedings of the 1994 IEEE Conference on Visualization*, pages 326–333, 17-21 Oct. 1994.

[76] Matthew O. Ward. A taxonomy of glyph placement strategies for multidimensional data visualization. *Information Visualization*, 1(3/4):194–210, 2002.

[77] Martin Wattenberg and Ben Bederson. Dynamic treemap layout comparison. http://www.cs.umd.edu/hcil/treemap-history/java_algorithms/LayoutApplet.html, 2007.

[78] Wikipedia. Visualization (graphic). http://en.wikipedia.org/wiki/Visualization_%28graphic%29, 2007.

[79] Kent Wittenburg, Tom Lanning, Michael Heinrichs, and Michael Stanton. Parallel bargrams for consumer-based information exploration and choice. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 51–60, New York, NY, USA, 2001. ACM Press.

[80] Han-Ming Wu and Chun houh Chen. Lecture Notes: Statistical Graphics and Visualization. Web: http://www.sinica.edu.tw/~hmwu/, 2006.