Hierarchical Difference Scatterplots: Interactive Visual Analysis of Data Cubes

Harald Piringer
VRVis Research Center
Vienna, Austria
hp@vrvis.at

Matthias Buchetics
VRVis Research Center
Vienna, Austria
buchetics@vrvis.at

Helwig Hauser
Department of Informatics
University of Bergen, Norway
Helwig.Hauser@uib.no

Eduard Gröller
Institute of Computer Graphics and Algorithms
Vienna University of Technology, Austria
groeller@cg.tuwien.ac.at

ABSTRACT

Data cubes as employed by On-Line Analytical Processing (OLAP) play a key role in many application domains. The analysis typically involves to compare categories of different hierarchy levels with respect to size and pivoted values. Most existing visualization methods for pivoted values, however, are limited to single hierarchy levels. The main contribution of this paper is an approach called Hierarchical Difference Scatterplot (HDS). A HDS allows for relating multiple hierarchy levels and explicitly visualizes differences between them in the context of the absolute position of pivoted values. We discuss concepts of tightly coupling HDS to other types of tree visualizations and propose the integration in a setup of multiple views, which are linked by interactive queries on the data. We evaluate our approaches by analyzing social survey data in collaboration with a domain expert.

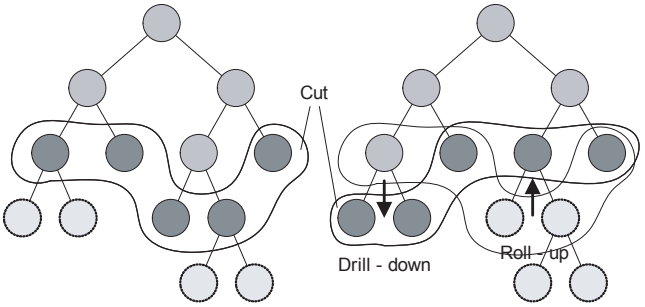

Figure 1: Navigating a hierarchy. Dark nodes represent the current state of navigation (the "cut"); nodes above the cut are contextual information and nodes below the cut are not visualized. Drill-down and roll-up operations transform the left hierarchy to the one on the right-hand side.

1. INTRODUCTION

Data dimensions of multivariate datasets can roughly be distinguished as being either continuous or categorical. While the data of some application fields is predominantly continuous (e.g., physical quantities), many application domains have to deal with mixed data, which has many categorical as well as continuous attributes (e.g., data from Customer Relationship Management). In this case, pivot tables are widely used to summarize the values of continuous attributes with respect to a classification given by categories. On-Line Analytical Processing (OLAP) [4] uses categorical attributes, called *Dimensions*, to split the data before aggregating continuous attributes, called *Numeric Facts*. An important aspect of OLAP systems is to use large-scale overview summaries of the data as starting point for selective drill down into interesting parts of the data.

OLAP is based on the fact that categorical data is closely related to hierarchical data and selective drill down (and roll up) is thus related to navigating a hierarchy. Apart from inherently hierarchical categories (e.g., years can be subdivided into months, days, hours, etc.), dimension composition is the key approach for defining hierarchies as it allows for specializing the categories of one attribute by the categories of another one. For example, two separate attributes "sex" and "age group" can be combined to obtain a category like "female and younger than 30". In the context of information drill down, pivot tables are also hierarchically structured and often referred to as data cubes (or OLAP cubes). Interactive analysis tools for pivot tables should consequently support navigation in a way that it is up to the user to decide where to drill down and where to stay at a summary level. They should reflect this hierarchical aspect in the visualization.

Apart from the navigation within the hierarchy itself, a frequent analysis task is to compare categories within one hierarchy level and also between multiple hierarchy levels. The difference of pivoted values with respect to parent categories may characterize individual categories very well as demonstrated by common statements like "the average income in a particular region is x percent higher as compared to the entire country". A visualization approach for OLAP cubes should therefore also facilitate relating categories along the hierarchy.

Based on these considerations, this paper introduces the Hierarchical Difference Scatterplot (HDS) as a novel approach to the interactive visual analysis of OLAP cubes. The following list of goals and tasks guided the design of HDS:

- Relating categories to siblings and to parent categories with respect to two continuous attributes. Our consideration is that differences between pivoted values of parent and child categories provide an intuitive way of comparison. We therefore represent them explicitly.
- Integrating multiple hierarchy levels into a single visualization in order to analyze hierarchy levels in the context of the other levels.
- Supporting local drill-down and roll-up (see Fig. 1). Unlike other hierarchical visualizations, it is an essential aspect of HDS to provide different levels of detail for various parts of the data instead of representing the entire hierarchy as such. This is in accordance with drill-down tasks in huge OLAP cubes, which also often emphasize depth rather than breadth.
- Supporting a setup of multiple linked views in order to dynamically integrate results of arbitrary queries as defined by the user in linked visualizations (e.g., a certain cluster of customers of a sales dataset as selected in parallel coordinates).

Our clear focus is on supporting specific OLAP tasks by a combination of visualization and interaction. It is explicitly not the goal of HDS to be superior to existing tree visualizations with respect to providing visually pleasing still images of huge hierarchies as a whole. For tasks where this is required, we discuss, how other types of hierarchical visualizations can be tightly coupled to HDS. As one of many potential application scenarios, we evaluate our approach by analyzing a real-world social survey regarding national identity. The analysis has been conducted in collaboration with a social scientist. We also provide a discussion of analysis tasks as supported by HDS, limitations, and a motivation for visualizing differences explicitly.

2. RELATED WORK

Pivot tables have long been used to summarize values of continuous attributes with respect to a classification given by categories. Flat pivot tables can be visualized using common techniques for multivariate, quantitative data. The Gapminder Trendalyzer [5], for example, maps two aggregated indicators of countries to the axes of a time-dependent scatterplot and shows the population, i.e., the size of the category, by the area of according circles.

The concept of pivoting data is also important for databases, where the predominant Structured Query Language (SQL), for example, offers the "GROUP BY" clause of "SELECT" statements for this purpose. However, as Gray et al. [6] point out, SQL statements have limitations with respect to drill-down and roll-up operations. Therefore they propose to treat multidimensional databases as n-dimensional data cubes, which have widely been adopted by On-Line Analytical Processing (OLAP) [4]. OLAP supports drill-down operations by splitting single categories with respect to additional dimensions.

While most OLAP front-ends only offer selected business graphics, Polaris [19] uses a formal algebra as specification of pivot tables and their visual representation. The user can incrementally construct complex queries by intuitive manipulations of this algebra. The layout is based on small-multiple displays of information [22]. Stolte et al. [20] also describe an extension to the algebra for rich hierarchical structures. Polaris is a very intuitive and highly effective approach for analyzing data cubes, as shown by the success of its commercial version Tableau [1]. However, Polaris displays a single level of detail (i.e., hierarchy level) and thus does not support comparisons between different levels of detail. The authors of Polaris also describe design patterns for adapting visualizations of data cubes on multiple scales [21]. This work deals with transitions between level of details while still showing a single level of detail at a time. It has been mentioned as future work to communicate parent-child relationships and to deal with non-uniform branching factors. The current version 5 of Tableau [1], however, does support comparisons between hierarchy levels using sub-totals and grand-totals, which are displayed in additional rows and columns. As the main drawback of this approach, comparisons require the user to look at multiple places on the screen in a successive manner. This makes comparisons difficult as will be discussed in more detail in section 6. This problem is inherent for approaches that rely on showing absolute values in a side-by-side manner. Therefore, visualizing differences explicitly was a main consideration in the design of HDS.

Yang et al. [26] propose a general framework for interactive hierarchical displays of large multivariate datasets, and they apply this framework to extend parallel coordinates, star glyphs, scatterplot matrices, and dimensional stacking. This approach categorizes a dataset by clustering before using this classification for multi-resolution analysis of aggregated values. However, unlike HDS as introduced in this paper, Hierarchical Parallel Coordinates are limited to comparing results along one cut through the hierarchy, while our approach focuses on differences between levels. Sifer [17] proposes parallel trees, which employ a parallel axes layout for aligning multiple drill downs into a data cube. The categories of all hierarchy levels are stacked on top of each other. For analysis, the user may relate one active dimension to all others by coloring parts of the boxes. This implicitly conveys the information for comparing siblings as well as child categories to parent categories. Differences are not represented explicitly which requires remembering one category and shifting the attention to another one for comparison. This becomes even more difficult as categories are scaled in proportion to their relative frequencies and thus their size may differ significantly. Moreover, parallel trees require categorization of continuous dimensions (i.e., facts) and do not support typical aggregations like average or sum. This severely limits their applicability to frequent OLAP tasks.

There has been very much research on the *visualization of hierarchies* and hierarchically structured data. Containment-based approaches like Tree Maps [16] are one of the most popular techniques and show the size of the hierarchy nodes very well, while depth information is occasionally harder to read. In contrast, node-link representations [2; 7] show the structure more explicitly, but most approaches do not clearly convey the size of the nodes. The rooted tree growing from top to bottom is a very common layout, but does not utilize space efficiently for large hierarchies. Centric approaches are superior in this respect as they grow outwards from the representation of the root node and thus allocate more space

to more detailed levels of the hierarchy. Nodes are typically placed corresponding to their position in the hierarchy, e.g., putting nodes with equal depth on concentric circles (radial tree) [2] or enclosing each sub-tree in a bubble (balloon tree) [7]. There are many extensions and variations to these approaches: focus+context techniques to improve scalability [9], combinations of node-link representations and enclosure [27], combinations of centric layout and enclosure [25], and edge bundles for integrating relations between items into the visualization [8]. Only a few approaches derive the node placement from multi-variate properties (i.e., each node is associated with several attributes) rather than edge topology as necessary for typical OLAP tasks.

Wattenberg proposes PivotGraph [24] for analyzing multivariate graphs and he addresses OLAP by supporting drill-down and roll-up. The graph layout corresponds to a grid which is given by two categorical dimensions for the X and the Y axes, respectively, and edge thickness is determined from the number of edges being aggregated. While the basic idea of property-based node placement is similar to HDS, there are several differences. PivotGraph only supports placement based on discrete dimensions while HDS uses a node layout scheme suited for comparison of differences between continuous facts. Moreover, PivotGraph visualizes a single level of detail at a time (similar to Polaris [19]) and thus does not allow for relating nodes to their parents. After all, the intention of PivotGraph is to improve the interpretability of the graph topology for a particular level of detail, while HDS focuses on comparing aggregated facts along and across a categorical hierarchy.

Queries defined through interaction within visual representations (also known as "brushing") are a proven standard approach for the identification of selected data subsets of interest. Successful systems such as Spotfire [18] offer interactive queries as an integral technology to link multiple views. There has been little research on integrating brushed subsets in hierarchical visualization techniques. In particular, no approach explicitly characterizes brushed subsets by displaying the difference between the properties of an entire category and its selected part.

3. HIERARCHICAL DIFFERENCE SCATTERPLOTS

This section introduces the Hierarchical Difference Scatterplot (HDS) as a novel combination of scatterplots and tree visualizations. After describing the approach itself, we provide examples of tightly coupling HDS to other hierarchical visualizations and propose techniques for linking our technique to other multivariate visualizations.

3.1 Visualization

The main idea of HDS is to layout nodes of a tree based on properties similar to a scatterplot (see Fig. 2). For parameterization, HDS require a pre-defined hierarchy, i.e., a data cube, and several properties, which are assigned to the visual attributes X-position, Y-position, size, and color. Properties may be pivoted values of continuous data attributes. An example are aggregated "measures" like the average revenue per node or other aggregates like minimum, maximum, median, sum, etc. Another possibility are inherent features of hierarchy nodes like absolute frequencies or depth. Applying data-driven glyph placement [23], the properties assigned to

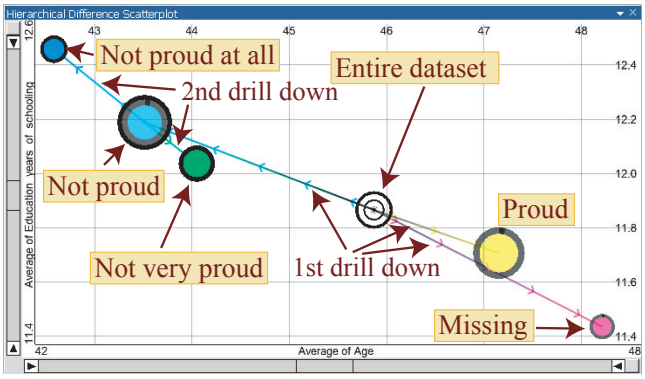

Figure 2: A simple hierarchy as conceptual example: the average age (X-axis) and the average years of schooling (Y-axis) are compared for several degrees of pride on armed forces and the entire data. Drill-down on "not proud" distinguishes "not very proud" and "not proud at all". The size of nodes shows the number of respective interviewees. The visualization reveals that pride is increasing with age and is decreasing with education.

the X- and Y-attributes are directly mapped to the position of the visual representations of categories. In addition to X- and Y-position, the user may independently assign different properties to size and color which is comparable to Polaris [19], or use default settings. For example, size per default represents the number of raw data items for each node. Color is discussed further below.

In accordance with the idea of information drill-down, the user may increase the complexity incrementally and selectively. Initially, the entire data cube is handled as a single category and it is thus shown as one visual item. By clicking on this item, the user may drill down to the next hierarchy level that displays the respective hierarchy nodes as additional visual items. Clicking on any of these items adds its direct children and thus increases the amount of shown information locally for this particular sub-tree (see Fig. 2). As most important aspect of HDS, the visualization is not limited to the categories within the current state of navigation in the hierarchy (referred to as "cut", see Fig. 1), but also includes all nodes above the cut up to the root of the hierarchy. This allows for direct comparison of properties between child nodes and parent nodes as both are displayed in the same visualization and thus share the same visual context with respect to node placement. However, this necessitates concepts for discriminating levels of the hierarchy and recognizing structural relationships, which we address in multiple ways.

First and foremost, lines connect each parent to all visualized children, thus representing the topology of the hierarchy. In order to improve the distinction of lines in densely populated areas, connection lines smoothly blend the color of the parent to the color of the child. As interesting aspect, these directed lines could be seen as "skeleton" of the visualization, which sketches the structure of the scatterplot of non-aggregated raw data entries. Even more important in the context of OLAP, the lines explicitly visualize the difference between the properties of each category with respect to its direct parent category (or the root of the hierarchy). Both the length and the angle have semantics, namely the overall amount of difference and the ratio. Due to the 2D

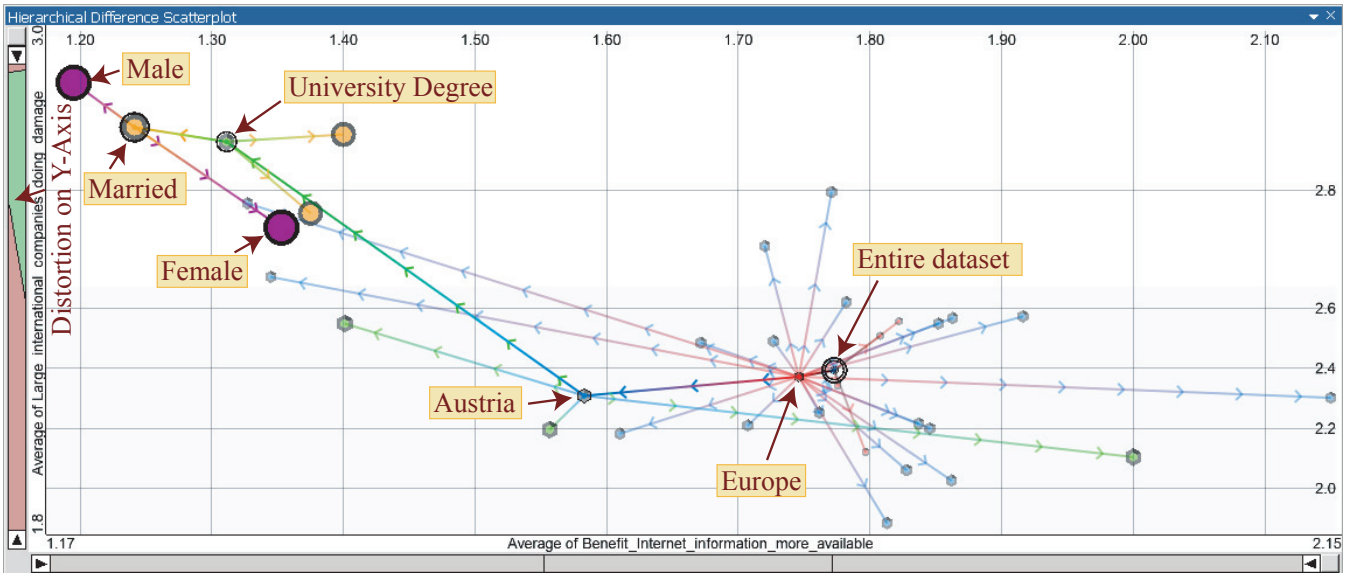

Figure 3: Example of a deep drill-down: the focus is on comparing men and women of the category path Europe – Austria – University degree – Married (i.e., five levels of the hierarchy plus the root) with respect to their attitude towards the Internet and international companies. Size, color and opacity are used to visually discriminate hierarchy levels. All siblings along the path are shown as valuable context information. Distortion is used on the Y-axis.

layout, the lines support the perception of relationships between differences on the X and the Y axis. This allows for fast identification of sub-categories deviating in the same way from their parents for multiple sub-trees. In our implementation, optional small arrows pointing towards the respective child indicate the direction and facilitate tracing the structure of the hierarchy in some situations at the cost of increased clutter.

As mentioned above, each visual attribute can be used in different ways. In particular, each attribute can be used to enhance the discrimination of hierarchy levels, where transparency can be modulated independently from color. Transparency and size-based discrimination amount to a focus + context approach. One hierarchy level C is considered to be the current one, which is drawn opaque and in full size. Opacity and size decrease for lower and higher levels N with a factor of $1/2^{|C-N|}$. The current hierarchy level is a global property of the visualization, i.e., the same depth is highlighted through all sub-trees. Drill-down and roll-up operations automatically update the current level, or the user may manually set any level as current. Expanded nodes, i.e., nodes above the cut, are highlighted by an additional opaque circle. Directed lines leading towards expanded nodes are always drawn in full opacity (see Fig. 3), which facilitates tracing individual sub-trees as generated by local drill-down.

HDS offer various modes for coloring hierarchy nodes. In addition to representing common categorical properties like size or pivoted values of an arbitrary measure as mentioned above, users may optionally also emphasize the structure of the hierarchy. Hierarchy-based coloring recursively subdivides the hue circle in a similar way as described for the Interring [25]. The segment of the hue circle assigned to each node is proportional to the number of leaf-nodes in the sub-tree and the hue in the middle of the segment is applied to the node itself. Color is a particularly important issue when coupling different tree-visualizations, as it supports the visual matching of hierarchy nodes (see Section 3.2).

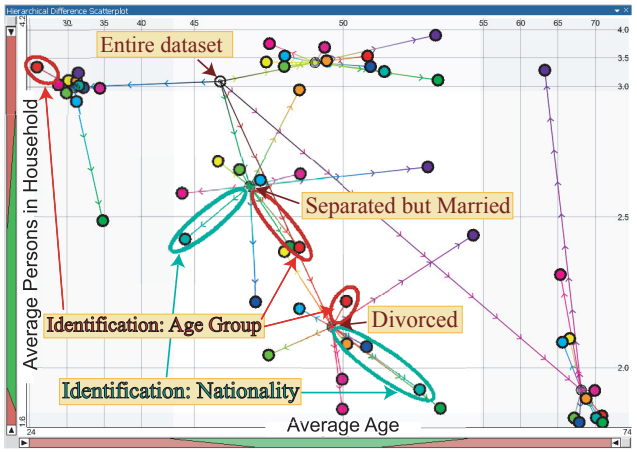

Figure 4: Comparing multiple sub-trees: interviewees are distinguished by their marital status and most important identification (in this order). Each class is characterized by its average age and the average number of persons in the household. While most identification nodes deviate roughly in the same direction for all marital status nodes, some interesting exceptions, like "Nationality", show contrary behaviour for different nodes. Color is derived from the category name. Spatial distortion is applied on both axes to focus on divorced and separated but married interviewees.

With an increasing number of displayed nodes, the extents and the density of the visualization may vary significantly. Restricting the displayed value range in a similar manner as in Spotfire [18] is supported by our approach, but it has the disadvantage that users may lose the overview because the entire hierarchy is not visible any more. As an alternative, we offer spatial distortion in a similar way as Table Lens [14]. This has proven useful to provide focus plus context for areas where nodes with similar properties lie close together. Applying a piecewise linear visual transfer function [3], the user may smoothly magnify any contiguous sub-interval of the displayed value range. The factor is chosen separately for the X- and Y-axis (see Fig. 3 and 4). The reason for using a piecewise linear function instead of using non-linear distortion (e.g., fish-eye distortion [9]) is that differences between nodes remain comparable as long as all involved nodes are inside the focus, which can easily be ensured by the user.

3.2 Coupling Tree Visualizations

Arguably, no single visualization approach perfectly covers all aspects of hierarchical data. The clear focus of HDS is on supporting the interactive analysis of data cubes in the context of OLAP. By displaying multiple pivoted values (or other properties) and the differences to parent levels at the same time, HDS visualize comparatively much information per node. Due to the data-centric layout, however, HDS do not perfectly scale to the visualization of both depth and breadth of large hierarchies at the same time (i.e., the hierarchy as a whole). This is due to well-known graph-drawing problems like a potentially high number of crossing edges. However, as discussed in section 6, this is not a limitation with respect to analyzing large real-world data cubes, because the user may increase the complexity incrementally and selectively by drilling down to interesting details while staying at a coarse level for less interesting sub-trees (or even hiding them).

Still, aspects conveyed not so well by HDS might be interesting. We therefore briefly discuss concepts of tightly coupling HDS to other approaches for visualizing hierarchies in order to combine their benefits when analyzing the same hierarchy. As an example, we have implemented a layout similar to parallel trees [17] as used by Sifer to analyze OLAP data. This layout is related to ArcTrees [10], which we refer to as hierarchical bargrams since we do not show any arcs. In hierarchical bargrams, a horizontal bar representing 100% of the displayed data is subdivided in proportion to the relative frequencies of the categories in the first level of the hierarchy. The obtained boxes are recursively split in proportion to the relative frequencies of their sub-categories. This generates bars nested inside the representation of their parent-category. Each bar displays the name of the respective node (see Fig. 5).

We have identified the following attributes for tightly coupling HDS to other kinds of tree visualizations.

- **State of navigation** The user may perform drill-down and roll-up operations in any visualization, which consistently updates all views. In the hierarchical bargrams, the recursion stops at the current cut, which is also conceivable for most other types of tree visualizations (like treemaps).
- **Color** As discussed above, HDS offer multiple ways for using color. Applying consistent coloring of nodes to all visualizations greatly facilitates the visual matching between them. In our case, the bars in the bargrams are drawn in the same color as the nodes in the HDS. Deriving the color from the position of nodes in the HDS (e.g., by mapping the position on the X-axis or the difference from the root to color) enhances the matching even more. Coupling by color is possible for almost all types of tree visualizations.
- **Order** Many tree visualizations have a degree of freedom in which order siblings are represented. This freedom can be used to roughly maintain proximities between nodes throughout all visualizations. The hierarchical bargrams, for example, optionally order sibling nodes with respect to their position on the X- or Y-axis in the HDS.
- **Selection** Interaction is generally very powerful for linking visualizations. We provide different types of selection: (1) based on dedicated mark up interactions (e.g., by drawing a rubber band or actively clicking on an item) (2) temporarily hovering over visual items, which highlights the node or sub-tree beneath the mouse cursor throughout all visualizations. This has turned out to be very intuitive and fast for matching nodes as no mouse clicks are needed.

3.3 Integrating Selected Subsets

The previous section discussed tightly coupling HDS to other tree visualizations. This section describes the integration of subsets as defined by brushing arbitrary multivariate visualizations like parallel coordinates. It also applies to linking multiple instances of HDS visualizing different hierarchies. Linking views by interactive queries has established itself as important concept, because different sub-tasks of a complex analysis typically require different types of visualization. For example, the user may want to identify multidimensional clusters in parallel coordinates, and immediately relate each cluster to a hierarchy as visualized by HDS.

In a linked setup, each type of visualization typically highlights the subset of selected entries in an appropriate way. In HDS, the integration is based on the fact that the selection state is categorical too. Each row in the underlying non-aggregated main data table is either selected or not at any point in time with respect to a particular query. Employing the concept of dimension composition, a selection thus refines any hierarchy node X into "X and selected" and "X and not selected". This allows for visualizing selections similar to normal child nodes.

For each node X of the cut, the aggregations of the selected part of X (unless empty) are computed and visualized at the respective position in the plot (see Fig. 6). As for actual sub-categories, a line connecting the representations of the entire category X and its selected part explicitly represents the difference between both with respect to pivoted values. In order to discriminate multiple selections, the border of selection nodes is drawn in the color of the respective query, while this part is black for actual nodes of the hierarchy. Immediately updating the visualization at each modification of the selection implicitly generates an animation of change similar to moving the time slider of the Gapminder Trendalyzer [5]. In our case it concerns general variation on arbitrary data dimensions. The modification speed of each node representation reflects the gradient of change with re-

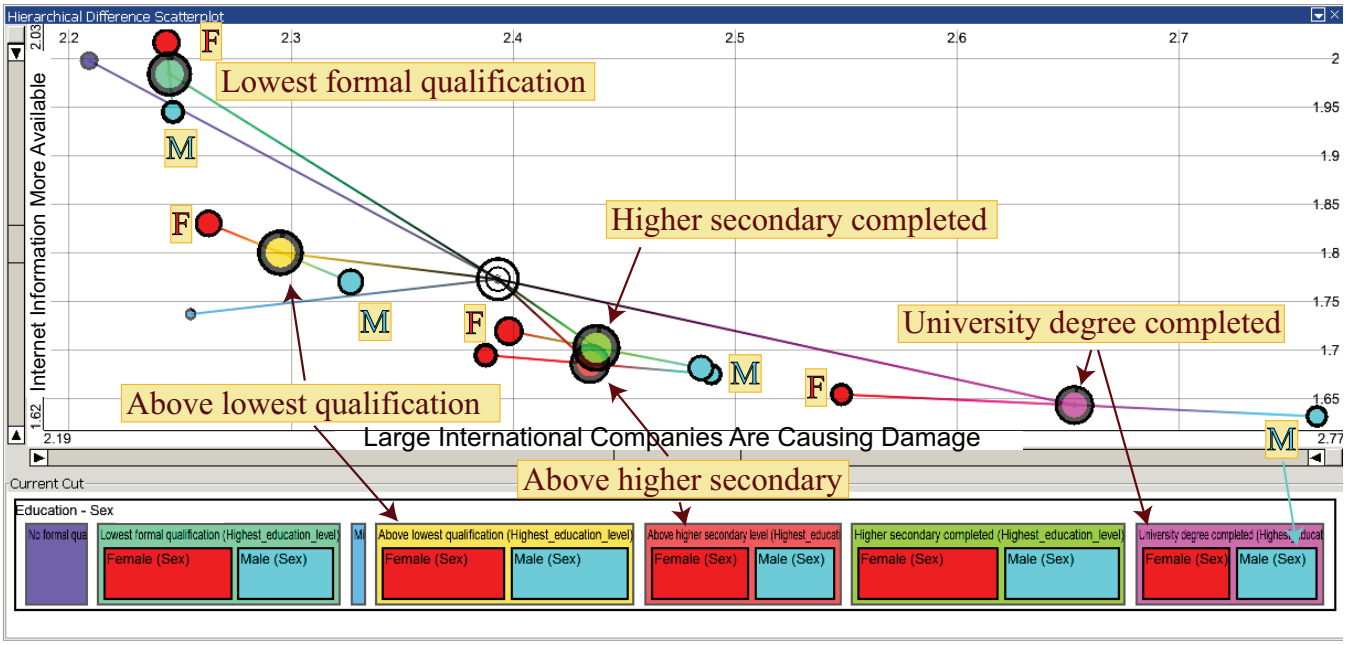

Figure 5: Tightly coupling HDS to hierarchical bargrams for displaying frequencies and names of hierarchy nodes. Several education levels, partly split into male (blue and letter "M") and female (red and letter "F"), are compared with respect to the average attitude towards international companies (X-axis, hue) and benefits of the Internet (Y-axis, saturation). The nonlinear relationship between the questions and the influence of education and sex are clearly visible.

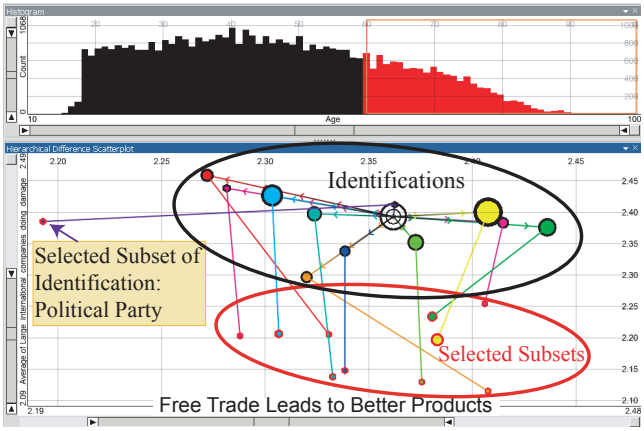

Figure 6: Integrating queries: interviewees older than 60 years, as brushed in a histogram, are highlighted for each category regarding most important identification with the average attitude towards free trade (X) and damage done by international companies (Y) assigned to the axes of the HDS. The visualization shows that elderly people tend to have an over-proportionally negative attitude towards international companies, while the attitude towards free trade is in most cases independent of age. The category "political party" is an exception, though, as the acceptance of free trade is higher for elderly people and – unlike for the other categories – more significant than the difference regarding international companies.

spect to the selection criterion. As recently discussed by Robertson et al. [15], it also reveals overall trends, e.g., all selection nodes move from left to right, and makes outliers discernable, which move in a contrary direction.

4. IMPLEMENTATION AND USER INTERFACE

HDS have been implemented in the context of VISPLORE, an application framework for visually supported knowledge discovery in large and high-dimensional datasets. VISPLORE supports the analysis of datasets with millions of entries and hundreds of dimensions at interactive rates on consumer hardware. This has a major impact on the design of all views (including HDS) and necessitates advanced software techniques like multithreading [12]. VISPLORE also supports missing values and requires all views to do so.

VISPLORE currently provides more than 10 different visualizations, which are partly standard (e.g., 2D and 3D scatter plots, parallel coordinates, histograms, etc.) and partly specific to certain application tasks [11]. A key aspect of VISPLORE is to discriminate multiple queries, which are defined by composite brushing and are highlighted by all views in a linked way. All components also offer convenience functionality like undo/redo and a consistent way to arrange controls like data dimensions of the current dataset. In particular, the user may at any time specify new hierarchies of arbitrary complexity by dimension composition or by combining categories. Data dimensions and hierarchies can easily be assigned to views, which is the way how the axes and the displayed hierarchy of HDS are parameterized.

Making the user interface easy-to-use was also an essential design aspect of HDS. The user may perform drill-down and roll-up operations by just clicking on a visual representation, or may hide entire sub-trees. Tool tips provide details-on-

demand showing the name, the size, and the aggregated values for the node beneath the mouse cursor. In order to highlight subsets of the data in linked views, the user can brush nodes by either clicking on them or dragging a rubber band. Dedicated widgets next to the X- and Y-axis offer all functionality related to adapting the displayed value range and the spatial distortion.

5. CASE STUDY AND EVALUATION

We now discuss the evaluation of our approach by the interactive visual analysis of a large survey, which we did together with a sociologist. The analysis of opinion polls is an important topic, where too little attention has been devoted to. HDS are designed to be generally applicable to data cubes of any kind, e.g., business data as a typical application of OLAP, and are not limited to opinion poll data. The sociologist had rich experience with the analysis of surveys, but had used static statistical software and had never used interactive visualizations before.

The survey was conducted by the International Social Survey Programme (ISSP) [13] in 33 countries between February 2003 and January 2005 with 44.170 respondents in total. Disregarding country-specific and thus incomparable questions, the dataset consists of 104 predominantly categorical attributes. The attributes are partly demographic questions and partly concern the attitude towards national consciousness, identity, and pride. The answers to most questions comprise 4 or 5 levels, e.g., very proud, somewhat proud, not very proud, not proud at all. This allows for both treating them as categories as well as computing meaningful aggregations, e.g., the average accordance to a statement. The dataset contains missing values, which represent an own category for categorical attributes. Missing values are disregarded when aggregating a continuous attribute.

Before analyzing the questions regarding attitude and pride, the sociologist first wanted to gain an overview about characteristics of various demographic categories, figures of the survey, and potential relationships between them. HDS facilitate this task, as it is fast to visualize simple pivot tables like the average number of persons in a household per country and they also quickly provide the size of each category. Within a few minutes, the expert could look at dozens of combinations, partly confirming expected facts, e.g., the average age of widowed people is 22 years higher than the average of the dataset. Partly, this basic analysis already revealed unexpected features like a significant variance in the average age of interviewees throughout the countries (which must be taken into account for subsequent conclusions).

Already for such flat pivot tables, the sociologist appreciated being shown the average of the entire dataset as visual reference. The reason is that this reference is not affected by categories of different size - a common problem when trying to determine the centre in a purely visual manner (e.g., by assuming the centre of the image as centre of the data, which is typically misleading). As criticism regarding our implementation, the expert said that he lacked labels next to the nodes, although he admitted that tooltips partly compensate for that. We suggested using coupled bargrams as legend and deriving the order of the nodes from the X-position in the HDS. He made use of them for cases where only a few nodes are simultaneously shown, while they turned out to be of limited scalability for more complex hierarchies.

After analyzing cross tabulations between categories (a frequent task in sociology) in another visualization of our framework, the expert returned to HDS in order to characterize categories in the context of other categories. For example, he was interested whether different categories concerning identification have a similar distribution of age for different marital status categories (see Fig. 4). Showing multiple hierarchy levels simultaneously and explicitly representing the difference between them turned out to significantly help answering this and other comparatively complex questions. Within a short time, the sociologist identified multiple interesting and unexpected facts in the data. Comparing the difference vectors of the red dots in Figure 4, for example, reveals that for all categories related to marital status, the subset specifying "age group" as most important identification tends to be older than the average of the entire category. However, singles are a remarkable exception, as the "age group" sub-category of singles is the youngest of all. Assigning the same color to related sub-categories (e.g., red to all "age group" sub-categories) greatly facilitates such comparisons between different sub-trees. As the visualizations became more complex, the sociologist used distortion increasingly often and found it a convenient way to clarify relationships for densely populated areas.

As the next step of the analysis, the expert was interested in results concerning attitude and pride. Figure 5, for example, shows that people with a positive attitude towards the Internet turned out to be less sceptical towards large international companies. It further reveals a strong influence of the education level. For drill-down scenarios involving more hierarchy levels, the sociologist liked that he could focus on particular categories but still see the rest as context information, as illustrated by figure 3. While focussing on Austrian interviewees with a university degree, still all other education levels are shown for Austria, all other European countries, and all continents. The centre of the entire dataset is given as well. The expert considered such deep local drill-downs a key advantage of HDS. Analyzing the difference between two levels is of course also possible by visualizing this derived information in simple scatterplots. Relating four or five levels at a time, however, would generate numerous derived data dimensions, which are hard to analyze intuitively without HDS.

The sociologist needed some time to familiarize with the idea of specifying ad-hoc categories by brushing linked visualizations. He eventually embraced this approach and used queries as defined in linked visualizations frequently for two types of tasks:

- **Motion** Due to the immediate update, changing the query in one view generates an animation in HDS. Figure 6 shows an example, where interviewees are selected by age in a histogram. Moving the interval from young towards old makes the selected parts of most identification classes in the HDS wander from top to bottom, indicating more scepticism towards international companies for elderly people. It also reveals interesting contrary trends for "political party" and "ethnic background" regarding the attitude towards free trade in dependence of age.
- **Highlighting** When comparing multiple sub-trees, a convenient way of identifying related categories throughout all shown extracted branches is by brushing this

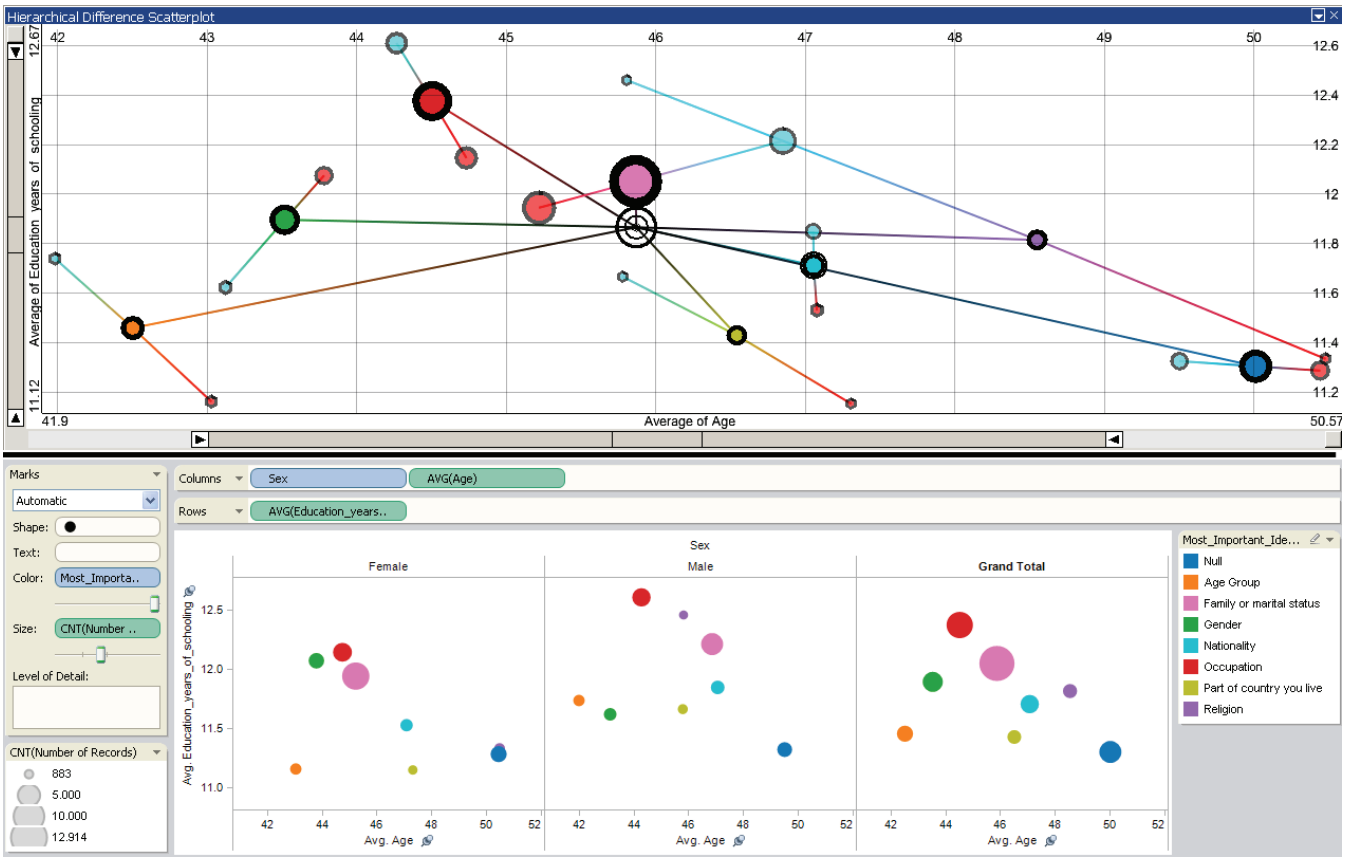

Figure 7: Comparing hierarchy levels using HDS (upper half) and using multiple scatterplots in Tableau (lower half). The average age (X axis) and the average number of education years (Y axis) are shown for groups having different most important identifications (color), which are further subdivided by sex. The same colors are used for corresponding identifications in both halves. In HDS, "male" is drawn blue and "female" red, while multiple panes are used below. Comparing especially the horizontal position of items is difficult across columns, while even minor differences are clearly conveyed by HDS. Please note that a color version of the figure is available in the online pdf-version of the paper.

particular category in a linked view. For example, instead of assigning the same color to related subcategories in figure 4, it would also be possible to highlight all "age group" nodes by selecting the category "age group" in another view, e.g., in a second instance of HDS. Using the "Superfocus", the sociologist could identify many different categories in a short time. This was particularly useful when color was needed otherwise - for example to discriminate hierarchy levels as in figure 3.

Although we can only describe a small part of our analysis here, this application has demonstrated how HDS facilitate and speed up the interactive analysis of data cubes. As result of our evaluation, the sociologist particularly liked being shown the centre of the data as reference and being able to analyze multiple levels of the hierarchy in the context of each other. Despite tooltips and coupled hierarchy visualizations, his most important criticism concerned the lack of labels, which we will address in future work.

6. DISCUSSION AND FUTURE WORK

The main idea of HDS is to support the interactive visual analysis of data cubes. Selective drill down ensures that users can increase the amount of detail incrementally for sub-trees of interest. This is an important aspect regarding the scalability of HDS. As for all approaches relying on pivot tables, the speed for aggregating data is the most significant limitation with respect to the number of underlying data rows. Aggregating data is generally fast even for millions of data rows and may even make use of explicit optimizations for data cubes in data warehouses. Therefore, HDS scale well for data sets consisting of multiple millions (and even billions) of underlying data records, which makes them applicable to real world data cubes.

A relevant question concerns the amount of detail (i.e., how many hierarchy levels and how many nodes), that can be shown before the visualization suffers from cluttering. An answer depends on the purpose. Generally speaking, HDS are suitable for:

- comparisons *along the hierarchy*. The main intention is to relate a few particular nodes to their direct and indirect parent nodes. Such comparisons involve local drill downs of numerous hierarchy levels while typically little information is shown per level (see Fig. 3 for an example). In this case, the most interesting information is the path to the root node (i.e., the properties of the entire data cube). Siblings provide rather context information and it is often even tolerable to hide sib-

lings for certain hierarchy levels. In this case, comparing ten or more hierarchy levels is possible. However, as shown by Fig. 3, the number of visual attributes needed for discriminating hierarchy levels generally increases with the number of hierarchy levels. As a special case, mapping the depth of a node within the hierarchy to its position on one of the two axes yields a common rooted tree layout. This layout is guaranteed to have no crossing edges as long as not more than one node is expanded per hierarchy-level.

- comparisons *across one hierarchy level*. The focus is on the position of siblings relative to each other and to common parent nodes (see Fig. 4 for an example). Much information is shown for a single hierarchy level while little information – if any – is typically shown for other levels. In this case, HDS resemble non-hierarchical scatter plots, but may still convey additional information (e.g., the properties of the entire data cube as one additional item). In this case, comparing a few hundred categories is possible.

As a consequence of displaying much information per node (i.e., two pivoted properties and topology), HDS are limited with respect to showing both depth and breadth of large hierarchies simultaneously. Showing large hierarchies in their entirety was not a design goal of HDS and it is not necessary for many tasks. As discussed in section 2, most tree visualizations convey the topology but disregard multi-variate attributes. Most approaches for OLAP, on the other hand, consider multiple attributes but are limited to displaying a single hierarchy level.

Tableau [1] optionally displays multiple hierarchy levels using sub totals and grand totals which are added as additional rows or columns. However, comparisons require looking at multiple places on the screen in a successive manner. Generally speaking, comparisons become increasingly difficult and less precise with increasing visual distance and number of visualizations involved in the comparison. For example, while detecting even minor differences in the height of two adjacent bars of a bar chart is easily possible, comparing the position of points of multiple non-adjacent scatterplot panes is difficult and coarse. The reason is that the user is forced to "remember" one pane while shifting his focus to another – potentially distant – pane. Fig. 7 illustrates this aspect. Although three panes (as shown in the lower half) is quite a small number, precise comparisons are particularly difficult with respect to the position on the X-axis. The figure also shows that using a single row makes comparisons with respect to height much easier, because the same vertical reference is given for all items. Using multiple rows (e.g., by assigning identification to rows instead of using color) would severely compromise comparability of the Y-position as well. In the worst case, comparing panes might even involve scrolling the entire visualization. This problem is inherent for approaches that do not explicitly visualize the difference between items but rely on showing multiple visualizations in a side-by-side manner as small-multiple visualizations typically do. Drawing items in a single scatterplot does explicitly visualize the difference between them, as this difference is directly proportional to their distance. This was a main consideration in the design of HDS.

There are multiple interesting directions for future work. First, we intend to conduct a large-scale user study in order to evaluate HDS more formally. Second, the issue of labelling nodes as mentioned by the sociologist needs to be addressed. The challenge is to add labels in a scalable way without compromising readability. Third, we plan to examine the effect of varying the shape of node representations on the interpretability of the visualization.

Other interesting questions for future research concern the applicability of HDS within small-multiple displays. A scatterplot matrix, for example, would allow for visualizing more than two measures at a time. Moreover, comparing drill-downs for multiple sub-trees of one node in a side-by-side manner could be an option for analyzing many deep drill-downs simultaneously. However, care will have to be taken as the aforementioned disadvantages of small-multiple visualizations apply in this case.

7. CONCLUSION

The analysis of data cubes is a key issue in many application domains. It involves navigating a potentially large hierarchy as well as comparing nodes within one or between multiple hierarchy levels with respect to properties like size and pivoted values. Particularly the difference between hierarchy levels is important information, which is not adequately represented by existing visualization techniques. Therefore, this paper introduced Hierarchical Difference Scatterplots (HDS) as an interactive approach to analyze multiple hierarchy levels in the context of each other and to emphasize differences between them. Visualizing both the topology and two pivoted values per node, HDS display much information at a time. For many tasks, this means an added value as compared to alternative approaches. For example, analyzing differences between hierarchy levels using non-hierarchical scatterplots requires the user to look at multiple views (i.e., positions of the screen) in a successive manner. HDS display the difference between categories explicitly within one visualization, which makes comparisons more intuitive and more precise.

A key idea of HDS is to allow for incrementally and selectively increasing the amount of detail using local drill-down. This ensures that the proposed concept of HDS is reasonably applicable to data cubes of any size. HDS employ several focus plus context approaches involving transparency, size, and distortion in order to ensure interpretability also for a significant number of displayed nodes. As other tree-visualizations are superior with respect to providing a pleasant layout of the entire topology or showing frequencies, we discussed concepts of tightly coupling HDS to other tree visualizations. Moreover, we discussed linking arbitrary other visualizations by user-defined queries to HDS. This allows for analyzing properties of ad hoc categories, it reveals trends through animations when changing queries, and it may also be used to highlight particular nodes. We described an evaluation of our approach by analyzing a large survey, which revealed numerous interesting and non-trivial aspects within a short time.

8. ACKNOWLEDGMENTS

This work was done at the VRVis Research Center in Vienna, Austria, in the scope of the projects MUMOVI, VISCOMP, and AVISOM (Nr. 818060). Thanks go to Florian Spendlingwimmer for the sociological support with the case study and to Martin Brunnhuber for important parts of the

implementation. Additional thanks go to Wolfgang Berger, Philipp Muigg, and Helmut Doleisch for help in preparing this paper. Finally, we want to thank our company partner AVL List GmbH for co-financing the application framework.

9. REFERENCES

[1] Tableau software. `http://www.tableausoftware.com`.

[2] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall PTR, 1998.

[3] S. Card, J. Mackinlay, and B. Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann Publishers Inc., 1999.

[4] E. F. Codd. Providing OLAP (On-line Analytical Processing) to User-Analysts, 1993.

[5] Gapminder Foundation. Gapminder. `http://www.gapminder.org/`.

[6] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. *Data Min. Knowl. Discov.*, 1(1):29–53, 1997.

[7] I. Herman, G. Melançon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, 2000.

[8] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):741–748, 2006.

[9] J. Lamping, R. Rao, and P. Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 401–408, 1995.

[10] P. Neumann, S. Schlechtweg, and M. Carpendale. Arctrees: Visualizing relations in hierarchical data. In *Proceedings of Eurographics, IEEE VGTC Symposium on Visualization*, pages 53–60, 2005.

[11] H. Piringer, W. Berger, and H. Hauser. Quantifying and comparing features in high-dimensional datasets. *International Conference on Information Visualisation (IV08)*, 0:240–245, 2008.

[12] H. Piringer, C. Tominski, P. Muigg, and W. Berger. A Multi-Threading Architecture to Support Interactive Visual Exploration. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1113–1120, November-December 2009.

[13] I. S. S. Programme. National Identity II. `http://zacat.gesis.org`, 2003.

[14] R. Rao and S. 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.

[15] G. Robertson, R. Fernandez, D. Fisher, B. Lee, and J. Stasko. Effectiveness of animation in trend visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1325–1332, 2008.

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

[17] M. 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.

[18] Spotfire Inc. Spotfire. `http://spotfire.com/`.

[19] C. Stolte, D. Tang, and P. 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.

[20] C. Stolte, D. Tang, and P. 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.

[21] C. Stolte, D. Tang, and P. Hanrahan. Multiscale visualization using data cubes. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 9(2):176–187, 2003.

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

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

[24] M. Wattenberg. Visual exploration of multivariate graphs. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 811–819. ACM, 2006.

[25] J. Yang, M. O. Ward, and E. A. Rundensteiner. Interring: An interactive tool for visually navigating and manipulating hierarchical structures. In *INFOVIS '02: Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02)*, pages 77 – 84, 2002.

[26] J. Yang, M. O. Ward, and E. A. Rundensteiner. Interactive Hierarchical Displays: a General Framework for Visualization and Exploration of Large Multivariate Data Sets. *Computers & Graphics*, 27(2):265–283, 2003.

[27] S. Zhao, M. J. McGuffin, and M. H. Chignell. Elastic hierarchies: Combining treemaps and node-link diagrams. In *INFOVIS '05: Proceedings of the IEEE Symposium on Information Visualization*, pages 57–64. IEEE Computer Society, 2005.