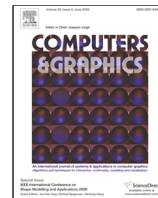




Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Parallel Hierarchies: A Visualization for Cross-tabulating Hierarchical Categories

Zana Vosough^{a,b,*}, Marius Hogräfer^a, Loïc A. Royer^c, Rainer Groh^b, Hans-Jörg Schulz^d

^aSAP SE, Chemnitz Str. 48, 01187 Dresden, Germany

^bTU Dresden, Nöthnitzer Str. 46, 01187 Dresden, Germany

^cChan Zuckerberg Biohub, 499 Illinois St, CA 94158 San Francisco, USA

^dAarhus University, Åbogade 34, 8200 Aarhus N, Denmark

ARTICLE INFO

Article history:

Received 5 April 2018

Keywords: Tree Visualization, Categorical Data Analysis, Interactive Exploration

ABSTRACT

The visualization of categorical datasets is an open field of research. While a number of standard diagramming techniques exist to investigate data distributions across multiple properties, these are rarely geared to take advantage of additional data properties – either given or derived. As a result, the data display is not as expressive as it could be when incorporating these properties, and it misses out on the potential of leveraging these properties for the data's interactive exploration. In this paper, we present the visualization technique Parallel Hierarchies that is specifically tailored to take hierarchical categorizations into account. With Parallel Hierarchies, it is possible to individually adjust the desired level of detail for each categorical data property through drill-down and roll-up operations. This enables the analyst to selectively change levels of detail as the data analysis progresses and new questions arise. We illustrate the utility of Parallel Hierarchies with a demographic and a biological use case, and we report on a qualitative user study evaluating this visualization technique in an industrial scenario.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

Many aspects of our daily lives are hierarchically categorized: the jobs we perform are specified using the Standard Occupational Classification (SOC) hierarchy [1], the books we loan from the library are organized by the Dewey Decimal Classification (DDC) [2], the illnesses we get are catalogued in the International Statistical Classification of Diseases and Related Health Problems (ICD-10) [3], and so on. One of the most interesting aspects of these hierarchical categorizations is when they get applied to the same set of individuals or items, as this enables us to systematically explore dependencies or cross-correlations between them. For example, people in certain occupations may be more likely to get certain health problems, and people with particular health problems may be more likely to read books on specific self-help topics, and vice versa. In

particular when not yet knowing these dependencies, interactive exploration of different hierarchies in various combinations and relations to each other, as well as at different levels of detail can reveal unexpectedly high or low numbers – so called *frequency counts* – between categories from different hierarchies.

Data visualization can enable such an interactive exploration of how numerical quantities distribute across multiple hierarchies. For relating hierarchical data to each other, the most common visualization approach is to draw different hierarchies side by side and to connect them with visual links [4]. In lack of a name for this type of visualization, it has been alluded to “as what Parallel Coordinates would resemble if the axes were hierarchical in nature” [4, p.10]. Commonly, this approach is focused on structural comparisons between similar hierarchies, such as showing the overlap between them, or determining which nodes have been added, removed, or changed with each version of a hierarchy [5, 6]. Yet for quantitative comparisons between entirely different hierarchies, this type of

*Corresponding author: zana.vosough@sap.com

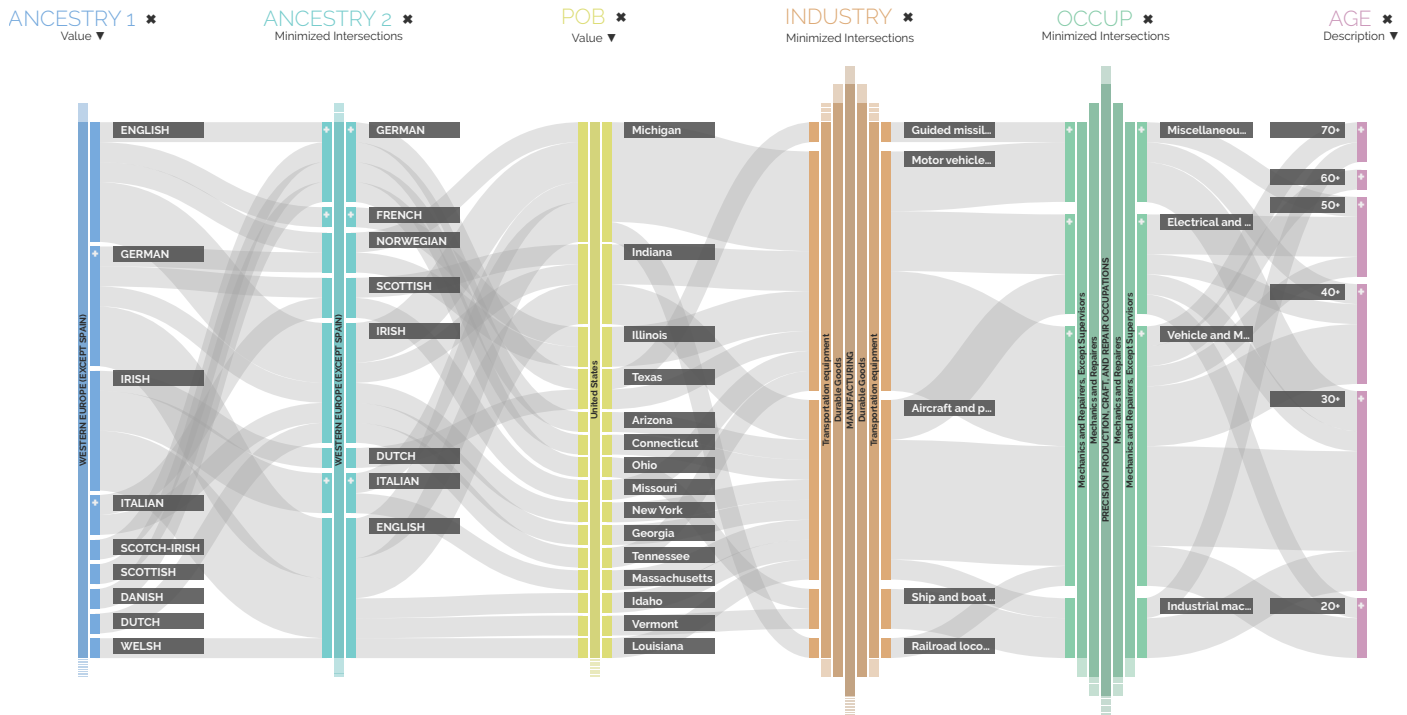


Fig. 1. Parallel Hierarchies showing some data properties of the US Census 1990 dataset. The current view shows the distribution of people with Western European roots who were born in the US and work as mechanics on any kind of transportation equipment. One can see from this figure, that this was a young profession in the 1990's with approx. half of these people being in their 20's and 30's. Also, the state of Michigan stands out, which is no wonder as it is not only famous for its automotive industry, but also for its people of European descent.

visualization has never actually been introduced, its design implications have never been discussed, and the resulting representations have never been evaluated.

This paper sets out to change this by providing a thorough description for this type of visualization, which we call *Parallel Hierarchies* and which is illustrated in Fig. 1. By contributing the necessary details to decide when, in which way, and to what end to use *Parallel Hierarchies*, this paper provides a point of reference for future implementations, derivations, and applications of this visualization technique. This contribution breaks down into three aspects:

- a formulation of the data analysis problem addressed by Parallel Hierarchies in Section 2 and an overview of related work and visually similar techniques pertaining to Parallel Hierarchies in Section 3;
- a description of the visual and interactive design aspects of Parallel Hierarchies in Section 4, as well as a discussion of common layout issues and limitations, and possible remedies for them in Section 7;
- use case examples illustrating how to apply Parallel Hierarchies in practice in Section 5 and a qualitative evaluation highlighting some user responses and observed usage patterns in Section 6.

In addition, we made the JavaScript/D3-based source code of Parallel Hierarchies freely available under an Apache 2.0 license at <https://parallelhierarchies.github.io>

2. The Hierarchical Cross-tabulation Problem

Parallel Hierarchies are a visual-interactive solution to the problem of cross-tabulating numerical aggregates over hierarchical categories. This section unpacks and describes this problem by breaking it down into the properties of the input data (Which data is to be processed and shown by the visualization?) and the necessary affordances of the visual output (Which actions must be possible to perform on the visualization?). We exemplify both using the 1990 US Census data, of which 1% and 5% samples are publicly available at https://www2.census.gov/census_1990/.

2.1. The Data: Hierarchical Categorical Aggregates

The data visualized by Parallel Hierarchies has three important ingredients: *data items* with categorical properties, *hierarchies* defined over these categorical properties, and *numerical aggregates* for each category.

Data Items. The basis of the data is formed by a set of items. In case of the 1%-sample of the US Census data, this set consists of 1.3 million data items – one for each person in the sample. Each data item has various properties. The properties of interest to us are the categorical ones, such as place of birth, ancestral background, occupation, and industry from the census dataset. If necessary, numerical properties can be transformed into categorical ones to also include them in the analysis [7].

Hierarchies. Each categorical data property is further hierarchically structured. This structure is usually explicitly given in the form of a known classification, like the ones mentioned in the introduction. But, it is also possible to define or derive

hierarchical structures for a flat categorization. Defining such a hierarchy involves some background knowledge about the data property – e.g., by grouping people’s ages into age groups (babies, children, teenagers, young adults, etc.). Yet, it can also be automatically derived without any background knowledge on the data property, for example, by using a hierarchical clustering algorithm for categorical data like ROCK [8]. Each hierarchical property forms a set collection with the sets being structured as an *inclusion hierarchy* [9, 10]. This means individual items are grouped into smaller sets, smaller sets are subsequently grouped into larger sets, until a single unifying set is formed at the very top of the hierarchy – for example, places of birth (e.g., municipalities) being grouped first by county, then by state, country, and continent, until everything is unified under the singular set “world”.

As we have multiple hierarchical categorizations given, these can also be understood as special cases of *multitrees* [11] or *polyarchies* [12] defined over the item set. In the US Census dataset, hierarchies are predefined for the aforementioned categorical properties at https://www2.census.gov/census_1990/pums_1990_b/document/.

Aggregates. Finally, we have aggregates (numerical values) given for each category and their hierarchical groupings into set collections. An aggregate can be understood as a function that maps any data subset onto a non-negative real value. The most common aggregate is simply the number of elements in a set. For the census data, this would translate to the number of people in a category – i.e., its frequency count. In addition, it is also possible to involve another data property as a positive weight being associated with each individual data item. Instead of adding up the mere item counts for the categories, one can also add up their weights into weighted frequency counts.

For the census data, we could use each person’s income as a weight, so as not to compare the number of people in different categories, but their combined income. Weighting by income might make more sense to marketers and sales people, trying to target groups of people with purchasing power. Whereas using the mere number of people might make more sense to political candidates, trying to win over the largest number of people, as their votes are all counted equal regardless of their income.

2.2. The Task: Cross-tabulating Hierarchical Categories

The task we aim to perform on this data has two aspects: (1) to find meaningful aggregation levels for the different hierarchies defined over the categorical properties of the dataset, and (2) to investigate the interrelation between the different categories of the dataset at these levels.

For finding suitable aggregation levels, two principal approaches are possible: *bottom-up* by starting from the most detailed level with all categories being shown and then successively folding the detailed categories into fewer high-level groups (roll-up); or *top-down* by starting from the least detailed level with only the top categories being shown and then successively unfolding them into more low-level groups (drill-down). Bottom-up and top-down approaches form dual perspectives on the task [13], each with its own emphasis on the problem. In terms of the census example, the bottom-up approach emphasizes the individual persons and how they are grouped into

increasingly broader categories, while the top-down approach emphasizes the overall population and how it is segmented into increasingly narrower categories.

For the interrelations between the different categories, the data is *cross-tabulated*. This means given two sets from two different hierarchical set collections, we are interested in their intersection – i.e., the data items they share – and its corresponding aggregate value. An example from the census data would be when cross-tabulating the attributes *place of birth* and *ancestry*: If the first set consists of all people born in California and the second set consists of all people with Hispanic roots, their intersection consists of all people fulfilling both of these properties. Doing this for all possible property combinations yields (weighted) *pairwise frequency counts*.

In combination, the described data and task specify the type of problems for which Parallel Hierarchies provide a visual-interactive solution. The following section highlights visualization techniques that address similar problems to this.

3. Related Work

From a data perspective, visualization techniques tailored to categorical datasets with additional properties have been presented in various contexts. The range of these techniques spans from time-oriented categorical data – e.g., to study patient data over time [14, 15], to geospatial categorical data – e.g., to study election results [16, 17].

From a visual perspective, visualizations in the style of *Parallel Coordinates* [18, 19, 20] using interconnected parallel axes have been proposed for a variety of data types. Examples are *Parallel Tag Clouds* for textual data [21], *Temporal Density Parallel Coordinates* for time-varying data [22], and *Parallel Node-Link Bands* for multi-modal social networks [23].

From both of these perspectives, Parallel Hierarchies add onto a rich State-of-the-Art. In doing so, Parallel Hierarchies combine ideas and approaches from various fields. Looking at the data shown by Parallel Hierarchies (cf. Section 2.1), the necessity of its cross-sectional character becomes apparent: sets of data items are usually shown using set visualization [24, 25], hierarchical structures are commonly displayed using tree visualization techniques [26, 27], and quantitative aggregates and their distribution over various categories are mostly dealt with by using categorical data visualization [28, 29].

Consequently, visualization techniques related to Parallel Hierarchies also incorporate aspects of multiple of these areas, as it is exemplified in Fig. 2. In the following, we discuss the shown examples, which stand as representatives for the types of visualization dealing with the different data combinations.

3.1. Visualizing Categorical Aggregates of Item Sets

Visualizations of this type show for an item set the pairwise frequency counts between a number of properties. What they lack is any hierarchy being defined over those properties.

Parallel Sets [30, 31] represent each property as an axis and connect the categories of neighboring axes with ribbons whose width is proportional to the number of items that exhibit both categorical traits. In the example in Fig. 2(a), the item set

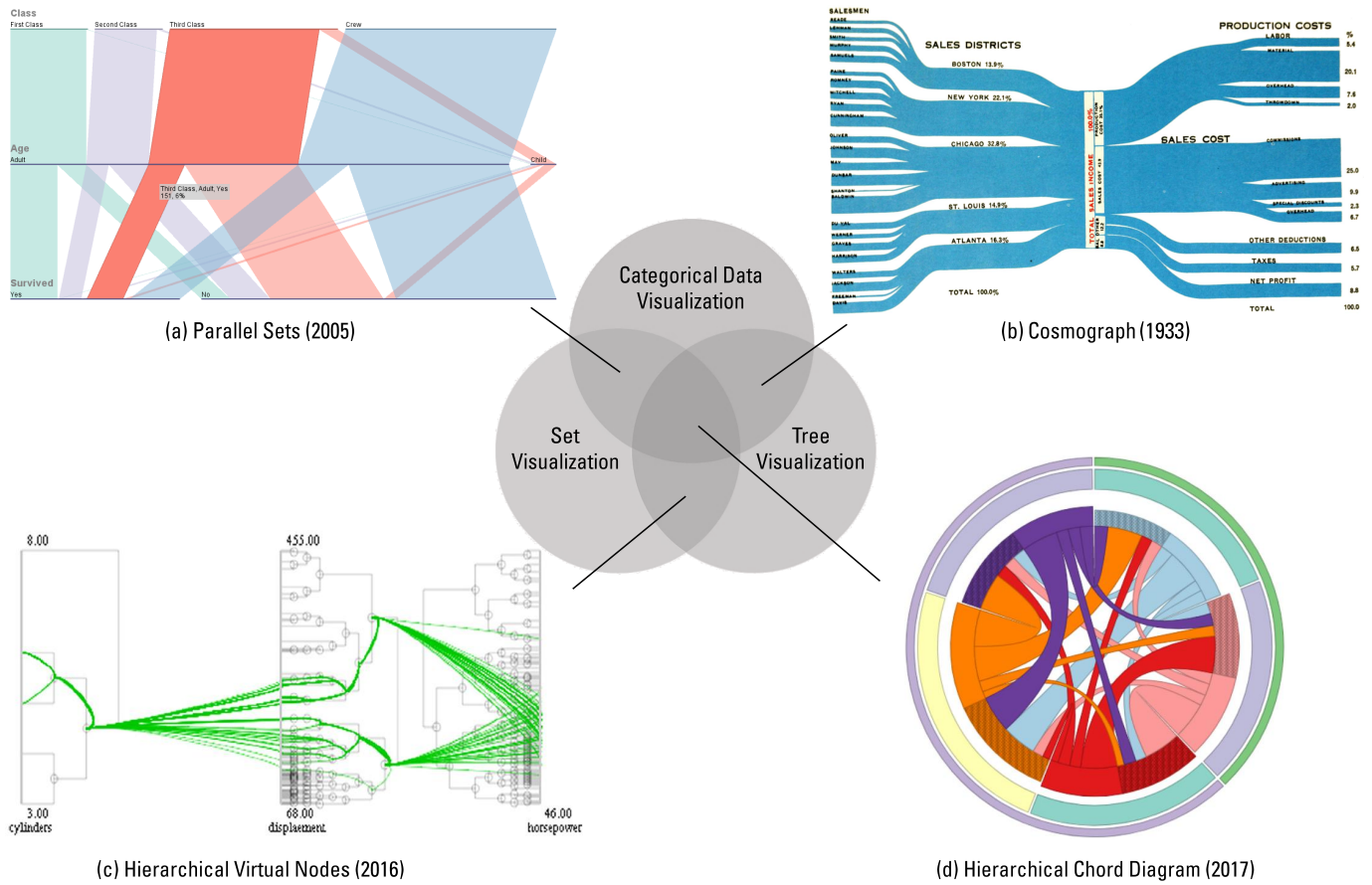


Fig. 2. An overview of the research fields related to Parallel Hierarchies and representative techniques for their different combinations.

constitutes the people aboard the sinking Titanic and the axes denote their categorical properties, such as booked class, age group, and whether they survived. This allows cross-tabulating the various categories to answer questions like *How did first class passengers fare compared to second class passengers?*

What this representation does not convey is any hierarchical structure of the categories along the axes – e.g., the example only distinguishes between *adults* and *children* for the age property, but no further drill down into the large adult category is possible. As a result, we cannot, for example specifically investigate the fate of elderly passengers among the adults.

3.2. Hierarchical Visualization of Categorical Aggregates

Visualizations of this type show two or more hierarchies defined over various categories together with a numerical aggregate. What they miss, though, is an underlying item set. As most data is given in the form of individual data items, visualizations of this type are rare.

One example is a visualization technique/device called *Cosmograph* [32] that was marketed by IBM in the 1930s. It was designed to manually generate flow graphs without the help of an “accomplished draughtsman” [33]. Cosmograph shows hierarchically organized categories on each side – e.g., the salesmen grouped into sales districts on the left and the various costs aggregated into a simple hierarchy on the right in Fig. 2(b). For each category, the numerical aggregate by which it contributes

to the total income/costs is shown as a percentage. This allows identifying the salesmen and districts with high sales volumes, as well as the cost positions contributing most to the expenses.

Yet, as it lacks the underlying set of individual sales transactions, we cannot actually cross-tabulate the categories. Only with the sales transactions – i.e., what was sold and its detailed production costs, how it was sold and its detailed sales costs, and who sold it where – we could relate the different categories to each other to find out, for example, *Which sales district incurs the highest sales costs?* As this information is not given, the strands are bundled in the center of the visualization, making it impossible to cross-tabulate categories from both sides.

3.3. Visualizing Item Sets and Their Hierarchical Properties

These visualizations show an item set as it is distributed across their hierarchical data properties. Yet, they lack a numerical aggregate that would quantify this distribution.

An example of this kind of visualization are *Hierarchical Virtual Nodes* (HVN) [34], which add hierarchical displays to each axis of a *Parallel Coordinates* plot [18]. This way, each item from the dataset is displayed as a sequence of curves routing through the tree structure of each hierarchical axis. The example in Fig. 2(c) shows a dataset of cars and the drawn curves – one for each car – give a rough impression around which attribute values certain cars cluster. A similar technique for textual data are *Parallel hierarchical Coordinates* (PhC) [35]. In

some sense, HVN and PhC can be understood as generalizations of *Hierarchical Parallel Coordinates* (HPC) [36]: where HPC clusters the item set as a whole, so that navigating the cluster hierarchy steers the overall number of polylines across all axes, HVN and PhC do so on a per-axis basis.

Yet, it does not convey the number of data items grouped in the hierarchically structured categories. In the example, this would mean to not only show the cars as individual curves, but also to show how many there are and how their number distributes over the other data properties.

3.4. Hierarchical Visualization of Categorical Set Aggregates

Visualizations of this type actually show all of the mentioned data aspects: the distribution of numerical aggregates of a set of data items over hierarchically organized categories. This makes these visualizations the closest relatives to Parallel Hierarchies.

For only a few hierarchical data properties, the literature makes mention of *Hierarchical Chord Diagrams* [37, 38] – an example of which is depicted in Fig. 2(d). These diagrams basically extend a regular chord diagram by showing an “inverted” *Sunburst* visualization [39] of the different hierarchies on the outside and connecting their categories with ribbons on the inside – somewhat similar to the *Contingency Wheel* [40] with the wheel being a tree visualization, or like Holten’s radial tree visualization with the bundled edges replaced by ribbons [41]. Note that the inversion of the Sunburst scheme into an “outside-in tree visualization” [42] turns its inherent benefit into a drawback: where Sunbursts grow outwards so that with every level more space is available on the circumference to show the increasing details of the hierarchy, drawing Sunbursts outside-in negates this effect so that it actually has less space available with every shown hierarchy level.

Hierarchical Chord Diagrams appear as visual representations in some application domains, such as computer networking and in the life sciences. To the best of our knowledge and very much like Parallel Hierarchies, Hierarchical Chord Diagrams were never formally introduced or evaluated as a visualization technique. The following sections fill this gap for Parallel Hierarchies, positioning them as a well-designed alternative to Hierarchical Chord Diagrams for future applications.

4. The Parallel Hierarchies Technique

Parallel Hierarchies is designed specifically to (1) navigate multiple hierarchies defined over the categorical data properties to find suitable aggregation levels, (2) cross-tabulate pairs of categorical data properties at their respective aggregation level, and (3) switch effortlessly between the two. Together with common guidelines for designing categorical displays [43, 44, 45], these specifications informed our visualization design.

Parallel Hierarchies feature an arrangement of *vertical axes* representing the different hierarchical data properties and allowing for drill-down and roll-up interaction. These axes are connected via curved *horizontal ribbons* that represent the pairwise (weighted) frequency counts for cross-tabulating neighboring axes. The following sections describe the combined visual and interaction design of the base visualization in three

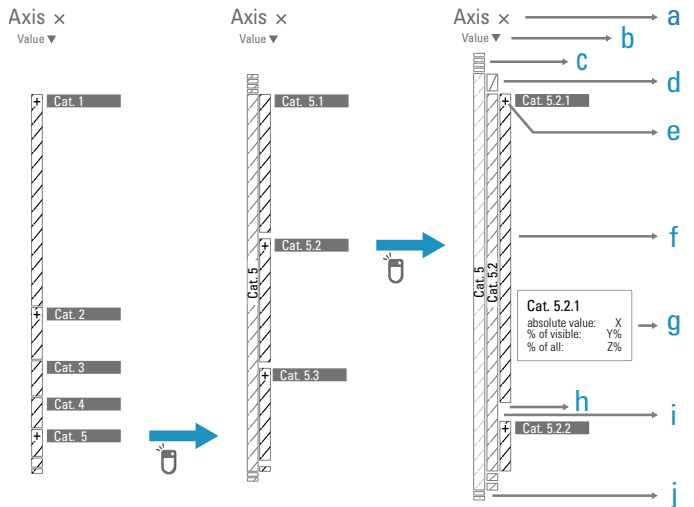


Fig. 3. An individual axis functioning as a simplified Icicle plot. Clicking on a category drills-down into the hierarchy. This condenses all siblings of the clicked category to make space for unfolding the next level of subcategories.

steps: for an individual axis, for pairs of axes linked by ribbons, and for a series of multiple such connected axes. In a fourth step, it details additional customization possibilities for further fine-tuning of Parallel Hierarchies – mainly by means of re-ordering axes, categories, and ribbons.

4.1. The Individual Axis: Showing Hierarchical Categories

Each individual axis in Parallel Hierarchies encodes one hierarchical set collection. To serve as an axis, the hierarchy display cannot take up much screen space and needs to be lean and uncluttered even for large hierarchies. For achieving this, we utilize simplified *Icicle Plots* [46] where “simplified” means that the hierarchy is not shown in full breadth and depth, but only the branch currently in focus is displayed. We thus employ a top-down approach, which is exemplified in Fig. 3 where we perform a 3-step drill-down into a hierarchical property.

On the left of Fig. 3, the topmost level of the data property is shown. The different categories are stacked into a *Spine Plot* [47] to convey the univariate distribution of items among them – i.e., the height of the vertical bar representing a category is proportional to the aggregate value of that category.

A mouse click on one of the categories drills-down and leads to the second view in Fig. 3. Note that the updated view focuses solely on the clicked category, which we call the *active category* and displays its subcategories, which we likewise term *active subcategories*. All siblings of the active category – i.e., those categories that we did not click on in the last step – are now reduced to a stylized representation at the top and/or bottom to provide a contextual indication of their number and their positioning according to the current ordering scheme.

Another click on one of the active subcategories drills-down further by making the clicked subcategory the new active category and displaying its subcategories. The path to the initially selected category is always visible as the clicked categories get stacked from left to right. For a roll-up, the user can simply click on one of these ancestral categories to make it the active

category again. This interplay of the interactive exploration of a tree’s topology with a dynamic adaptation of the tree’s display is reminiscent of the *SpaceTree* technique [48]. Albeit we utilize a more condensed and simplified adaptation mechanism that focuses solely on the path from the root to the active subcategories – i.e., the “ancestor path”, as it is called in *SpaceTrees*.

The directionality of this path is indicated by a color gradient that assigns darker shades to the “older” ancestral categories that have been unfolded and explored a while ago, and lighter shades to the “younger” active subcategories that were just recently added to the stack of categories and that are currently explored. This minimalistic approach for exploring trees under space constraints has been described in the literature, with a preliminary evaluation suggesting that it outperforms other tree views on small displays for tasks involving known targets [49]. In a sense, this form of drill-down/roll-up forms a combination of pivoting certain values on an axis [50], grouping related values on an axis [51, 52], and filtering values on an axis [53].

The following list briefly describes the individual aspects of this hierarchy representation as they are labeled from (a) through (j) in Fig. 3:

- (a) Name of the hierarchical data property, which can also be interpreted as the root of the hierarchy. Note that we do not otherwise show the root node in an axis as it has no added value to show the “distribution” of a singular item.
- (b) Current ordering scheme of the categories, and interaction handle to change the scheme. Details on different orderings are given in Section 4.4.
- (c) Unselected siblings positioned before the selected category according to the currently chosen ordering scheme. The representation is stylized, meaning that nothing is encoded in their height; they merely indicate their number.
- (d) Ancestors of the active category (i). The ancestors represent the path of clicked categories that led to the current view. Ancestors also serve as interaction handles to trigger roll-up operations back to their level.
- (e) Small + marker indicating that this particular active subcategory splits even further into more detailed subcategories. If such a more detailed categorization is needed, an active subcategory with such a marker can be clicked to trigger a drill-down operation.
- (f) Active subcategories split the aggregate value of the current active category (i) and encode these splits proportionally in their respective heights. In addition, they present possible options for further drill-down operations.
- (g) On demand detail information showing up when mousing over any category – ancestor (d), active category (i), or active subcategory (f). The shown details include the full name of the category, and its absolute aggregate value, as well as the relative aggregate value w.r.t. the data subset currently visible and w.r.t. the total dataset.
- (h) Gaps in between active subcategories (f) delineate the subcategories from each other. The size of the gaps varies (cp. second axis in Fig. 3) with the number of active subcategories, as the available whitespace is distributed equally between subcategories.

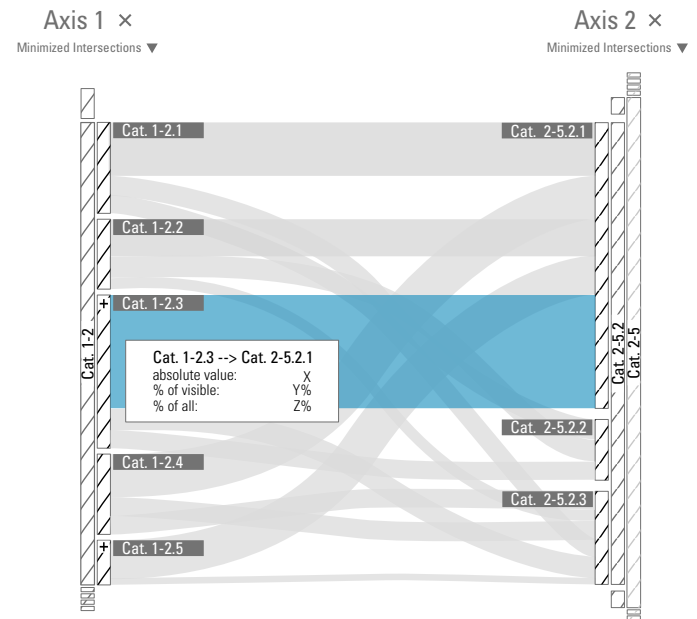


Fig. 4. Connecting the active subcategories of two axes with ribbons to show their pairwise frequency counts. Hovering over a ribbon with the mouse highlights it and displays detail information.

- (i) Active category indicating the current focus of the axis, as every drill-down operation also implies a filtering of sibling nodes of the active category and of its ancestors (c,j).
- (j) Unselected siblings positioned after the selected category according to the current ordering scheme.

4.2. Two Interlinked Axes: Showing Pairwise Frequencies

When combining two of these axes, we can display two different hierarchical set collections, one on each axis. To indicate their pairwise intersections for all active subcategories of both axes, we connect both axes with curved ribbons whose width is proportional to the aggregate value of the intersection they represent. Together, these ribbons provide for a full cross-tabulation of both sets of active subcategories.

Fig. 4 gives an example of two connected axes, cross-tabulating their active subcategories. The two axes stack the hierarchy levels towards each other, so as to make connecting them easier. Curved, light gray ribbons are drawn from one active subcategory on one axis to another active subcategory of the respective other axis if these share at least one data item – i.e., their corresponding aggregate value is larger than 0. We decided for using curved instead of straight ribbons – specifically cubic Bézier curves – as smooth paths and contours are known to be easier to follow than straight ones [54, p.191]. This is particularly important, as the ribbons potentially connect all active subcategories of one axis with all active subcategories of a neighboring axis, leading to a cluttered display that can be challenging for tracing individual ribbons.

The width of the individual ribbons corresponds to the magnitude of the associated frequency count, which means for our example, the more people fall into both subcategories, the wider their connecting ribbon. In sum, the width of the ribbons stack

up to the overall aggregate of both connected active categories. The ribbons generally follow any interactive operation (drill-down, roll-up, reorder,...) performed on the axes. Mousing over an active subcategory highlights all incident ribbons. Mousing over a ribbon highlights both incident active subcategories. The latter also yields more detailed information on demand and shows for example the absolute and relative aggregate value for a selected intersection.

It has to be noted that the space between two axes is constrained by the available screen space, which can be problematic for drilling into particularly deep hierarchies. Resulting layout problems can be that both hierarchies do not leave enough space for routing the ribbons in between them, or the Icicle plots could even meet in the middle. To avoid these problems, we define a maximum number of levels to be stacked onto an axis. If this number is exceeded, ancestor categories further away than this maximum number of levels from the active category become increasingly thinner to save horizontal space.

4.3. Multiple Linked Axes: Propagating Frequencies

Concatenating multiple instances of the bivariate display introduced in the previous section effectively extends it into a full-fledged Parallel Hierarchies visualization. From a visual perspective, this extension is simple: to be able to connect ribbons to intermediate axes from both sides, we simply mirror the Icicle plots for all axes except the leftmost and rightmost ones. This is illustrated in Fig. 5.

Conceptually, this extension brings up a number of new aspects that need to be considered for the axes, as well as for the ribbons. Showing only (weighted) pairwise frequency counts between neighboring axes limits the analysis to 2-way cross-tabulations between those categorical properties shown. To counter this effect, Parallel Hierarchies allow for flexible configuration of the axes – i.e., which properties are shown as axes and in which order – as it is also common for Parallel Coordinates. This configuration is manually adjusted by adding, moving, and removing axes until the view is appropriately configured to answer a given analysis question.

Yet however purposefully configured, the resulting view is still restricted to only cross-tabulating neighboring pairs of axes. To mitigate this restriction to some extent, the hovering/highlighting mechanisms for active subcategories and ribbons is further refined to propagate the highlighted items across axes and to show their spread across the whole view. For example, on the right side of Fig. 5 the ribbon connecting “Cat.2-5.2.1” with “Cat.3-5.2” is moused over. This highlights the ribbon itself and shows how many data items fall into the intersection of these two properties. In addition, this subset of data items is further propagated to the left, showing how these data items are distributed across the different active subcategories of “Cat.1-2”. While all involved ribbons between “Axis 1” and “Axis 2” are highlighted, the distribution of the moused over subset among these ribbons is indicated with a darker highlight at the bottom of each ribbon. In the shown case, we see that most of these data items fall into category “Cat.1-2.3”.

To better distinguish the different axes from each other, our particular realization applies a color coding to the axes that assigns different colors to different axes, as shown in Fig. 1. The

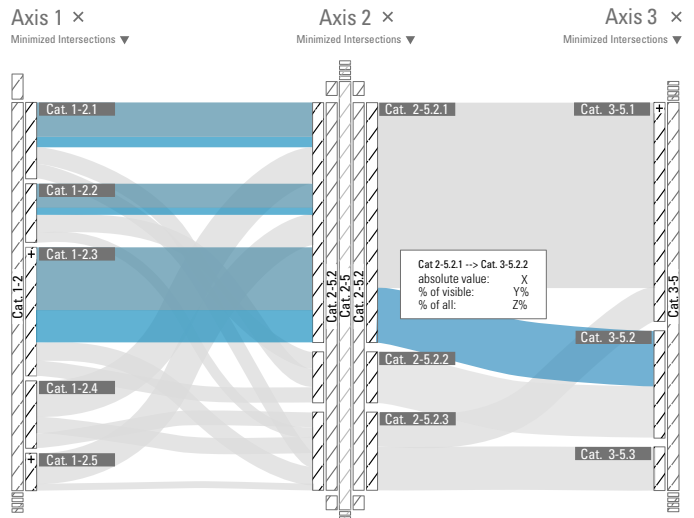


Fig. 5. Concatenating multiple axes and interlinking them with curved ribbons to represent multiple pairwise frequency counts. When highlighting a ribbon in this setup, the highlighted data subset is visually propagated to indicate its distribution across axes.

colors of the axes follow Paul Tol’s categorical color scheme Palette II [55], which is specifically designed to be compatible with both light and dark backgrounds. This makes it easy to plug Parallel Hierarchies into different applications regardless of whether they use a light or a dark UI theme.

It has to be noted that all axes are linked into a unified visualization. This means that any interactive change (roll-up/drill-down) to one of the axes affects the shown item set as whole and not just the one represented by the changed axis. For example, if one was to drill-down further, for example into the subcategory “Cat.3-5.1” in Fig. 5, this would not just reduce the item set on “Axis 3” and its incident ribbons, but for the entire visualization. I.e., the height of the active subcategories on all other axes and the width of their incident ribbons would also adjust to then reflect the particular distribution among those data items falling into “Cat.3-5.1” only. As a result of this interlinkage, a few drill-down operations on some of the axes can reduce the shown dataset significantly, up to the point where the visible item set is no longer large enough to draw meaningful conclusions from it. To give the user an overview of how much of the dataset has already been filtered out and is no longer visible, a small column chart at the top of the visualization gives an indication of this information. See Fig. 6 for the column chart corresponding to the data shown in Fig. 1. The column chart gives the user not only an idea of what portion of the data is missing overall (gray column), but also to which degree the drill-down operations on the different axes are responsible for it, as the columns follows the color coding and axis ordering of the main visualization. If desired, the user can also switch the column chart around to indicate the percentage of the data that remains visible.

4.4. Fine-tuning Parallel Hierarchies through Reordering

The previous sections describe the base visualization of Parallel Hierarchies including interactive means of its adaptation,

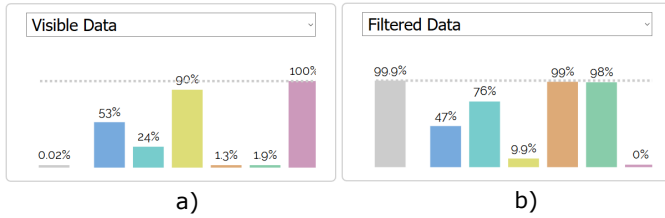


Fig. 6. Column chart indicating how much of the overall dataset is currently visible (a) or filtered out (b). The column in gray gives an overall indication for each, whereas the colored columns indicate to which degree the drill-down on the corresponding axes is responsible for this.

such as drill-down and roll-up of hierarchy levels, or adding and removing axes. This section expands on possible adaptations of the base technique by discussing the finer details of its layout and exploring their degrees of freedom. These degrees of freedom mainly stem from the fact that neither the different data properties (axes), their different categories (active subcategories), nor the pairwise intersections between them (ribbons) have an inherent order that dictates their position on the screen. Instead, we are quite flexible to adjust that order as it is suitable for the analysis task at hand.

4.4.1. Reordering Strategies for the Axes

When deciding for an arrangement of the axes, it depends to what end this arrangement is made. We distinguish between two cases: providing an overview of the data with all data properties being equally important, and looking at details of how a specific data property of interest relates to the rest of them.

In the first case, where no particular data attribute is of more interest than any other, we are free to arrange the axes in whichever way produces the clearest and least cluttered overview. The common approach used for Parallel Coordinates aims to identify correlations, convergences, or other patterns between axes, so that placing these axes next to each other yields a less cluttered output that clearly exhibits the found patterns [19, Sec.6.2]. This approach does not work for Parallel Hierarchies, as we are cross-tabulating all active subcategories between two axes. This means, in many cases we have a complete many-to-many connectivity between the axes, where the connections only differ in their respective pairwise frequency count – i.e., all possible ribbons are present and differ only in their respective widths. To reduce some clutter, we suggest to reduce the number of ribbons by applying one of the following two heuristics:

Alternate axes with many active subcategories and axes with few active subcategories. As potentially all subcategories of an axis are connected to all neighboring subcategories, it makes sense not to place two axes with many subcategories next to each other, as that would produce a high number of ribbons and thus massive clutter between them.

Place axes with a one-to-many relation next to each other. If two data attributes exhibit such a relation, it is even possible to relate them without any ribbon crossings at all. An example would be the sales districts and salesmen from Fig. 2(b), where each salesman is associated with exactly one district.

In the second case, where we want to explore the interrelations of one particular data property to all other properties, we need to prioritize this aim over clutter reduction. This can be done in two ways:

Positioning the axis of interest in the middle. This simple idea stems from the observation that assessing propagated item sets across multiple axes becomes increasingly difficult the more axes the propagation passes. This is as with every axis, the item set gets further spread into increasingly thinner partial ribbons that are further and further away from the originally highlighted subcategory or ribbon. By placing the axis of interest in the middle of the view, this effect is lessened to some degree.

Adding the axis of interest multiple times and interleaving it with the others. Parallel Hierarchies also permit to add the same axis multiple times, so that it can be placed in alternating sequence with the other axes. This approach makes it easier to cross-tabulate the axis of interest with all others, as it does not rely on propagation. Yet, it also requires more horizontal screen space to accommodate the duplicates.

Note that these four strategies can of course be combined with each other. For example, we can position an axis of interest in the middle and still arrange all other axes in an alternating fashion with respect to their number of subcategories around the central axis.

4.4.2. Reordering Strategies for the Categories

Much has been published on how to establish a sensible sorting of items with no inherent order along an axis [56, 57, 58]. As it turns out, there is no universal order that would satisfy all possible visualization needs and support all possible visualization tasks. Instead, it is paramount to be able to switch between different sorting strategies depending on the analysis task, which can be loosely aligned with the task taxonomy by Andrienko and Andrienko [59]:

- Direct look-up tasks (*given: category, sought: corresponding value/count*) benefit from an alphabetical order that allows users to quickly find categories by their name.
- Indirect look-up tasks (*given: aggregate value/count, sought: category*) benefit from an ordering according to the aggregate values or counts of each category.
- Comparison tasks (*given: categories, sought: their relation*) benefit from an ordering of categories that minimizes clutter and crossings when connecting to neighboring axes to visually identify categories with similar and/or different connection patterns.
- Relation-seeking tasks (*given: a relation, sought: categories conforming to that relation*) benefit from any ordering of categories that take the given relation into account – e.g., ordering based on a correspondence analysis when looking for similar categories.

Ordering based on results from a correspondence analysis has been treated in depth by Johansson and Johansson [60],

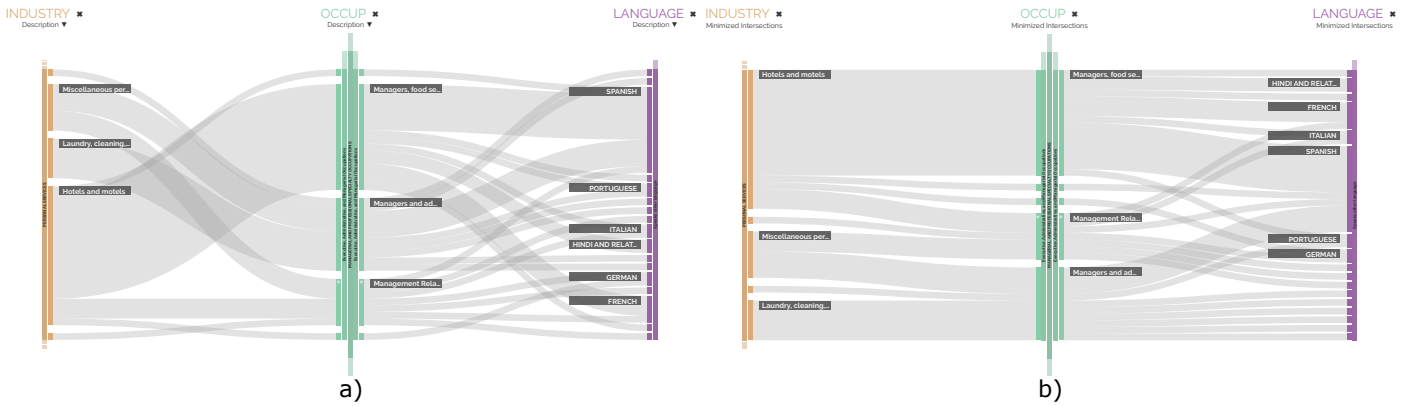


Fig. 7. Before (a) and after (b) applying the crossing minimization to the ordering of categories along the axes. In this example, the crossing minimization reduces the number of ribbon crossings from 134 down to 23. In general, it can be observed that the crossing minimization is most effective for drilled-down views where no longer every category is connected to every other category, as the data gets more sparse.

and we refer the interested reader to their work for details. Re-ordering categorical items on parallel axes to reduce crossings between their connecting lines or ribbons can be formalized as an instance of the *k*-Layer Straightline Crossing Minimization problem, which is known to be NP-hard even for $k = 2$ [61]. For computing optimized orderings, we employ Step II of the well-known Sugiyama layout heuristic [62] in combination with the barycenter method. This has already proven to be an efficient heuristic for crossing minimization for bipartite graphs [63], and it has also been successfully used to reduce ribbon crossings in storyline visualizations [64, 65]. As an additional optimization, a greedy switching heuristic can optionally be used in addition to the barycenter method [66]. While this additional heuristic is able to generate slightly better results in most cases, it also requires a much longer runtime. This makes it ill-suited for use in exploratory analyses, but it is a good option for “pretty printing” a final result for presentation purposes.

Other quality metrics for ribbon-based visualizations aside their crossing number have been proposed in the literature. For example, Perin et al. [67] considered large link heights – i.e., large vertical spans of ribbons – to be disadvantageous to reading a visualization. The reason is that ribbons of large height are more susceptible to the line width illusion if they are straight [68] or to the sine illusion if they are curved [69]. Both of these illusions alter the appearance of a ribbon’s width depending on its slope. This is echoed by observations on the legibility of *Stacked Graphs*, where bands of the same nominal thickness yet with very different slopes appear to be of different width [70]. These considerations are also a concern for Parallel Hierarchies. But since ribbons with a large vertical span are also prone to cross a large number of other ribbons, the applied crossing minimization implicitly also reduces the link heights.

Another often-found consideration in relation to visual clutter is the angular resolution of those lines that nevertheless do cross each other [71]. Taking this into account, one might want to reorder so as to also maximize angular resolution, which is beneficial for tracing crossing lines as it avoids confusing the lines due to narrow crossing angles. Yet, this issue is not as prevalent when ribbons cross each other, as in most cases the

ribbons have different widths and can thus easily be identified when tracing them across parts where they overlap. In case of doubt, a simple mouse over highlights any ribbon showing exactly where it originates and where it leads.

As Parallel Hierarchy visualizations tend to also incorporate axes with an inherent order (e.g., age groups) or a user specified order (e.g., sort by name or value), we keep the categories on those axes fixed and use them as starting points for computing the order of the remaining axes. The effect of reordering the categories along the axes is illustrated in Fig. 7, which shows how this approach unclutters the view. From our experience, the Sugiyama layout heuristic using the barycentric algorithm reduces crossings on average by 20% to 25%. The additional reduction by postprocessing the barycentric order with the greedy heuristic lies around 1%-2%, but can be up to 7% in rare cases.

4.4.3. Reordering Strategies for the Ribbons

The third possibility to fine-tune the appearance of Parallel Hierarchies is to adjust the vertical order of multiple ribbons connecting from/to the same active subcategory of an axis. This order is not predetermined by the data, yet most axes-based visualizations featuring ribbons use a “source-based” ordering that sorts ribbons along an axis according to their order along the neighboring axis to which they connect. This makes sense as it eases tracing ribbons from one axis to another by giving us a rough idea of where a ribbon should land on the far side: if it originates from a subcategory at the bottom, it will also connect to the bottom of any subcategory on the other axis, and vice versa.

As for the rendering order of the ribbons – i.e., which ribbons to draw first and which last – this is only an issue for opaque ribbons that would overplot already drawn ones. For opaque ribbons, one could for example choose to draw wide ribbons first and thin ribbons last. This strategy would ensure that a few thin “outlier ribbons” are not covered up by the wider ones. Or the ribbons could be drawn in the opposite order, if one wants to make sure that a few wide “main trend ribbons” are not drowned out by a crisscross of hundreds of thin ones. In our implementation of Parallel Hierarchies, this point is of lesser interest, as

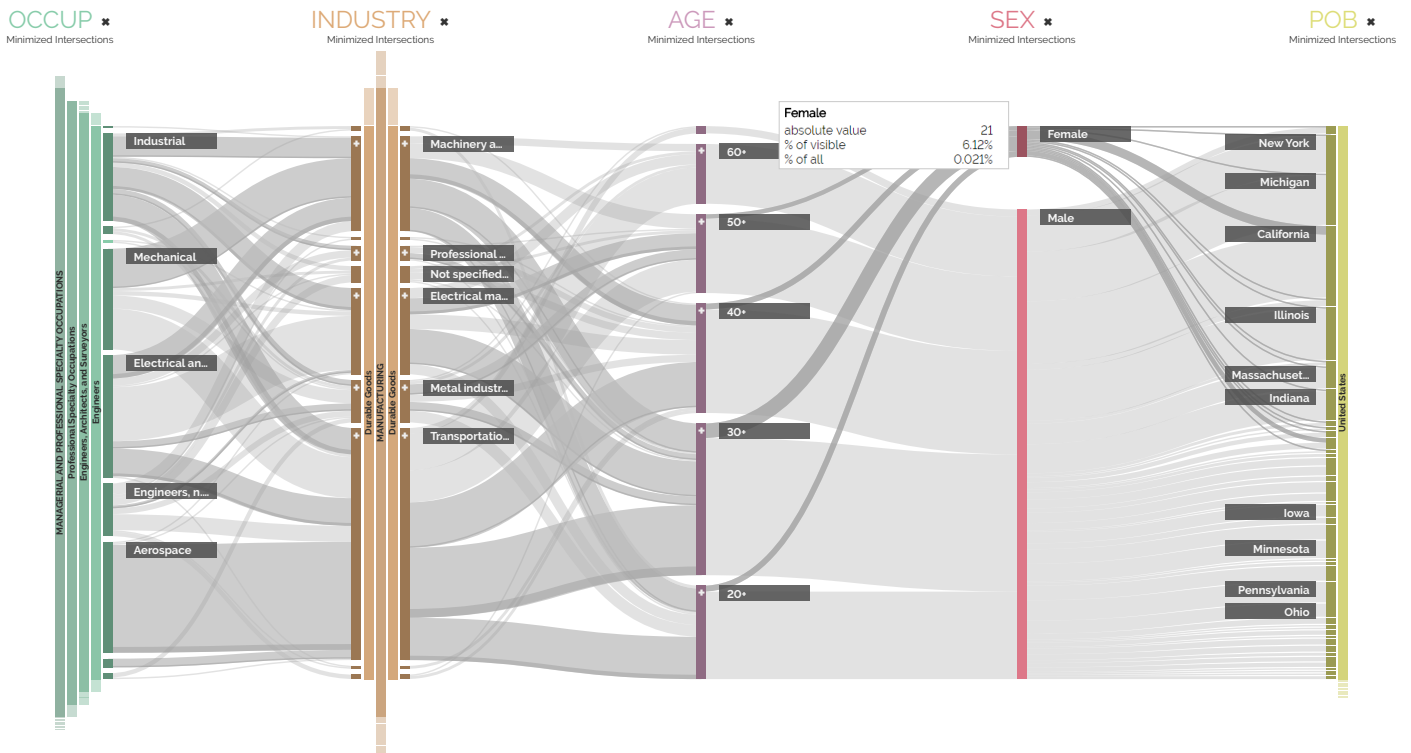


Fig. 8. Parallel Hierarchies visualization of a 100k sample of the US Census 1990 dataset. The highlight emphasizes the few women among the US-born engineers who work in the production of durable goods.

we use semi-transparent ribbons with alpha blending. Yet in the same vein as the different ordering strategies, we could of course assign different alpha values to emphasize certain ribbons more than others, making sure they are well visible.

5. Applying Parallel Hierarchies

The following describes two use cases in which we applied Parallel Hierarchies – one use case looking at demographic data from the US, and the other use case dealing with genome data of yeast. These two examples are to provide a first impression of Parallel Hierarchies in action in two very different fields.

5.1. US Census Data

The US Census dataset was already briefly introduced in Section 2. In this section, we look at a subset of 100k items from that data. The dataset features 68 attributes, some of them numerical, others categorical, and out of the categorical ones a few with a given hierarchy. In this example we focus mainly, but not exclusively on the hierarchical attributes. The view in Fig. 8 shows three of them: POB (indicating the Country in which the person was born), INDUSTRY (the type of industry in which a person works), and OCCUP (a person's primary occupation). Furthermore, we added a hierarchization of the numerical category AGE into age groups and the flat categorical attribute SEX. The aggregate values represented by the height of the categories and by the width of the ribbons are the number of people.

We want to investigate the relation between different work environments as signified by the attributes OCCUP and INDUSTRY, and the AGE and SEX of the people working there.

Hence in Fig. 8, we rearranged the axes so that OCCUP and INDUSTRY are in the center of the view. Moreover, we avoided placing POB and OCCUP next to each other as they both have many subcategories, which would produce a high number of ribbons. In addition, we adjusted the ordering strategies for some axes. For example, the POB axis is ordered descending according to the aggregate value, so that it supports the indirect look-up of states from where many or few people stem. In contrast, the AGE axis benefits from an ordering by description, so that the different age groups are ordered ascending from youngest to oldest, thus supporting a direct look-up strategy.

In this example we drilled-down to the people born in the US, who work as engineers in any industry manufacturing durable goods. We immediately see the small number of women working in this area. Upon hovering over this category, the women in this view are highlighted and we learn from the tooltip that only 6% of all employees in this area are women – 21 in total. From the spread of the highlighted ribbons, we also see that these women are mainly in their 30's and about a third of them stem from California. When changing the aggregate value from the number of people to their income, which is not depicted in the figure, one can further find that while there are roughly 13 times more men than women working in this field, these men make 20 times the money that the women make.

More details on this use case and a showcase of the described interactive analysis is given in the accompanying video. When showcasing the visualization to demographers who work with such data on a daily basis, they immediately noticed the generality of the Parallel Hierarchies approach. In particular for questions involving cross correlations, they were eager to use

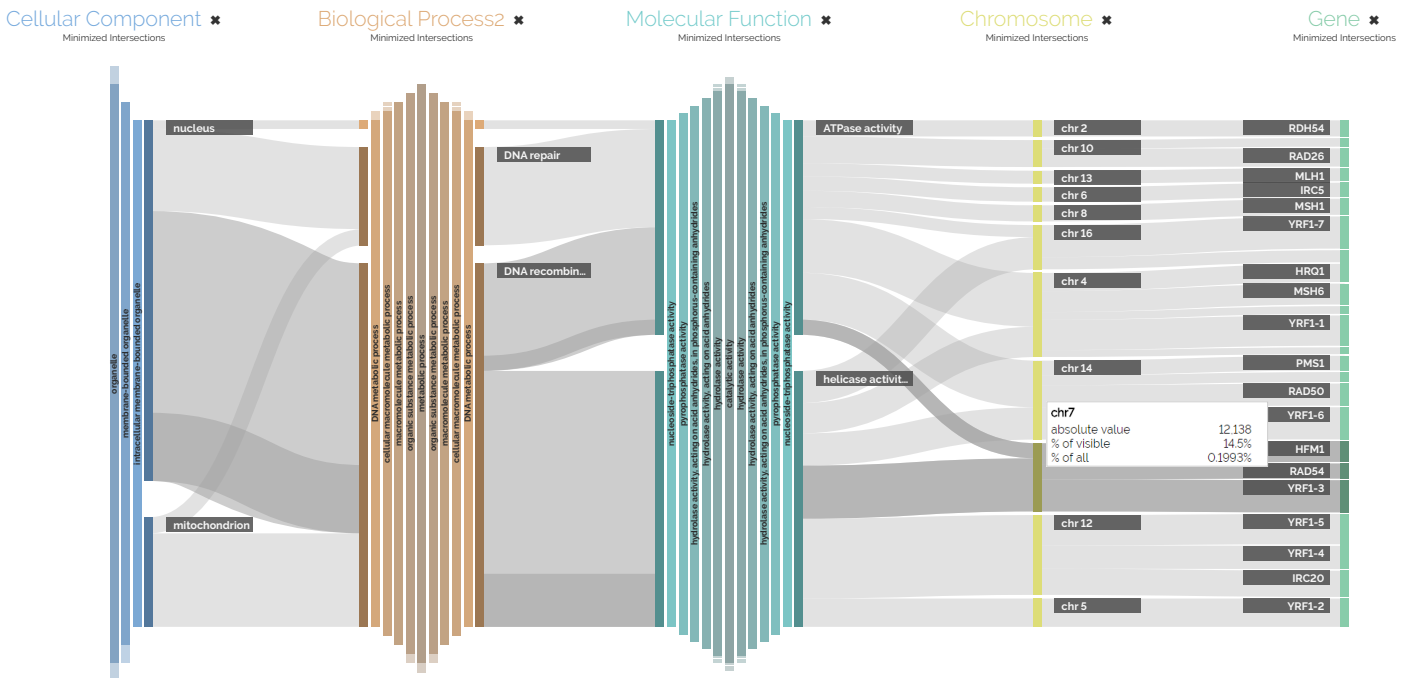


Fig. 9. Parallel Hierarchies visualization of the yeast (*S. cerevisiae*) genome and its Gene Ontology annotations. The view was drilled-down to display only genes that relate to the DNA metabolic process in intracellular membrane-bound organelles whose function involves nucleoside-triphosphatase. The current highlight emphasizes the subset of genes that are encoded on chromosome 7 – which pinpoints exactly three out of the more than 3,000 genes.

the visualization – e.g., to investigate how the degree of education, place of birth, and level of income influence life expectancy and fertility rates as predicted by different statistical models using different parameter settings.

5.2. Yeast Gene Ontology Annotations

Understanding how cells function requires an understanding of the molecular parts of the cell, its genes and the proteins they encode. And while ribbon-based visualizations for biomedical use cases have been proposed in the past (see e.g., *StratomeX* [72] or *CooccurViewer* [73]), they are still far from commonplace in the toolbox of biomedical researchers. In this use case, we consider a dataset of 3,813 genes of the *S. cerevisiae* (yeast) genome, which we obtained together with their chromosomal locations from the Saccharomyces Genome Database (SGD, <https://www.yeastgenome.org>). For the categorical properties, we used their annotations from the Gene Ontology (GO, <http://www.geneontology.org/page/downloads>). These annotations form three hierarchies: *cellular component*, *biological process*, and *molecular function*. As shown in Fig. 9 (right side), Parallel Hierarchies let us visualize the yeast genes and their chromosomal localization. In addition, we can display their Gene Ontology annotations as hierarchical axes in Fig. 9 (left side).

In this use case, we seek to identify genes involved in a specific biological process, having a given molecular function in a particular cellular compartment. Specifically, we want to find the genes that are localized in the intracellular membrane-bound organelle (cellular component), that bind nucleoside-triphosphatase activity (molecular function), and that are involved in DNA metabolic process (biological process). This

task is challenging because it requires simultaneously navigating several hierarchies and performing a joint selection. Identifying such genes would typically require writing a database query or custom script. There are two obstacles to this: First, the actual terms and concepts to be queried against have to be known in the first place, which is not obvious when dealing with an ontology consisting of tens of thousands of concepts. Second, assuming that the parameters of the query are known, it remains necessary to write and perform the query itself – something that requires a certain degree of expertise. In the words of one of our biologist interviewee: “*Asking a biologist to write a complex database query is equivalent to asking computer scientist to run a gel electrophoresis, possible in theory but unlikely in practice.*” Online tools such as genome browsers or Gene Ontology browsers exist to aid biologists, but these cannot visualize the interplay between different concepts, let alone allow to specify the sought genes across different hierarchies.

A first advantage of Parallel Hierarchies is that drilling-down into a specific hierarchy reduces the drill-down choices available for the other hierarchies – a key feature to facilitate interactive exploration of the data. For example, drilling down to a specific cellular compartment can decrease the number of corresponding genes by two orders of magnitude but also reduces the number of relevant molecular functions and biological processes. Indeed, when we asked biologists to test our visualization, they mentioned that “*It is great that selecting a specific term deep in a hierarchy can teach what other terms are relevant in other hierarchies.*”

Another advantage of the joint visualization is that the relationships between hierarchies becomes explicit. For example,

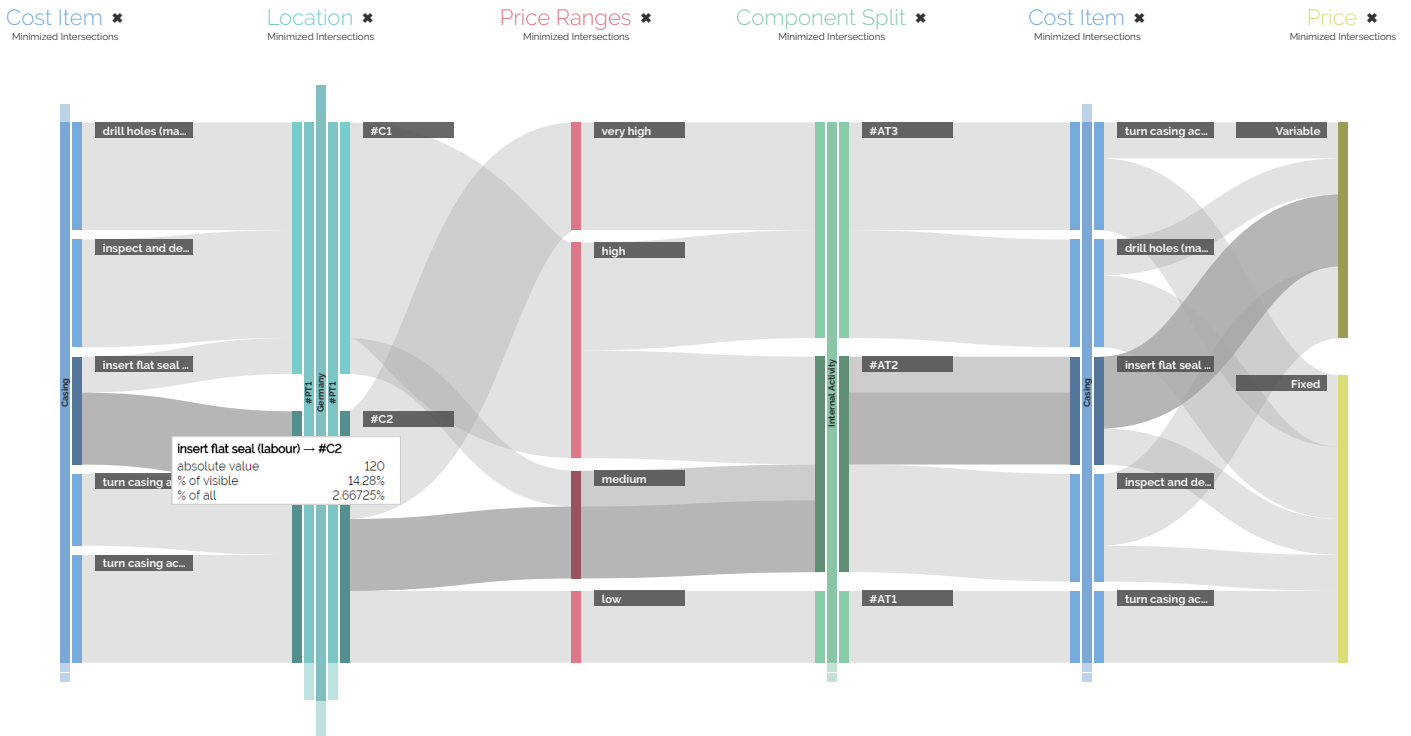


Fig. 10. Parallel Hierarchies visualization of the costing structure of an industrial pump with 92 items. In this view, we show the particular costing structure of the casing. To ease observing its distribution across the other data properties, the axis “Cost Item” has been added twice. The current highlight emphasizes the labor of inserting a flat seal, which is in the medium price range and entirely variable in its price, as wages are often a fluctuating factor.

Fig. 9 shows the preponderance of genes encoding helicases in the Yeast cell nucleus. Moreover, by rolling up the *molecular function* dimension we can visualize the proportion of nucleus genes that function by means of nucleoside-triphosphatase versus all genes localized in the DNA metabolic process. The highlighting feature of Parallel Hierarchies allows us furthermore to highlight specific chromosomes (here chromosome 7) and see which nucleoside-triphosphatase activity, intracellular membrane-bounded organelle genes are located on that chromosome. This facilitates interesting and complex observations such as that all genes involved in nucleoside-triphosphatase activity found in the nucleus except for HFM1 are encoded on chromosome 7, and are directly involved in DNA recombination. Biologists to which we showed this interactive querying mentioned that “*exploring such datasets often requires that we ask an expert such as a bioinformatician to do the analysis for us, write our own scripts, or go through some tedious manual search. With this solution I just have to look and click.*”

The ribbons’ height – which is proportional to the number of genes – helps to convey the relative number of genes involved in different biological processes, molecular functions, localized on different cellular compartments, and encoded on different chromosomes. Besides the interactive visual representation of such a dataset, this use case illustrates how Parallel Hierarchies can help to find a needle in a haystack by providing a faceted search interface through the individual drill-down/roll-up of the different hierarchical axes.

6. Evaluating Parallel Hierarchies

Our particular realization of Parallel Hierarchies, as it was described in the previous section, was initially developed as an interactive visual analysis technique in the domain of product costing. Product costing involves analyses where one wants to break down the overall costs of a product along various aspects, such as cost types (e.g., labor, materials, patent fees, and taxes) and product components (e.g., frame, tires, electronics, engine, and seats) to find cost drivers and thus potential savings when designing a new product. The requirements for a visualization, which is able to support such interactive analyses, were established over the course of multiple formative user studies as part of the *SAP Product Life-cycle Costing (PLC)* co-development program [74].

As part of these formative user studies, we investigated the current use of visualization tools in this particular application domain, which turned out to be mostly pie charts and bar charts as they are offered by today’s spreadsheeting software. Yet none of these diagrams are actually used to perform costing analyses. These analyses are currently conducted by applying filters or by writing functions in spreadsheet-based software solutions. There is no tool available to interactively explore the data and look for patterns or outliers in the data set.

As none of the used diagrams is a good fit for the complex nature of the costing data, we exposed the participants in a second study to Treemaps [75] and Sankey Diagrams [76] to show the hierarchical break-down of costs along the categories. These

visualizations were met with great enthusiasm by the costing experts and the Parallel Hierarchies technique was the logical combination of the Treemap hierarchy display (albeit now shown as Icicle plots) and of the interconnection among categories with ribbons in the Sankey Diagrams.

Since these initial studies were conducted, more than 30 customers and partners who partake in that program have given input on the design of Parallel Hierarchies, which took about 2 years from start to finish. It is in this setting that we conducted an *empirical qualitative user study* [77] of the resultant Parallel Hierarchies technique, on which this section reports.

6.1. Setup of the Evaluation

The Data: The general properties of the dataset used in the evaluation follow the costing scenario: The set of data items contains the individual product parts, including intangible parts such as software licenses and measures for quality control. The hierarchical categories are defined over the categorical attributes of these product parts – e.g., material type, place of production, and the part/whole relationship that via multiple stages forms the overall product from these parts. While *material type* does not strike one as being of hierarchical nature, individual materials can in fact be hierarchically grouped – for example, into raw materials, packaging materials, services, etc. The numerical aggregate was the cost. For our evaluation we used a small realistic dataset for an industrial pump with 92 parts and 6 attributes associated with each part. This dataset was based on a real-world dataset from one of our customers, but was slightly modified by us to obscure its source and to inject a known ground truth for the participants to find. The dataset is shown in Fig. 10.

The Participants: We conducted our evaluation with 15 product costing experts from 9 companies in individual 1-hour sessions over the course of two days. The participants were recruited during a customer workshop and all of them had no prior visualization experience beyond standard charts as they are available in most spreadsheet and BI software. Their application backgrounds are mainly the automotive and machine building industries, where they work in various roles from IT specialists to managers. This group of participants had the following structure, which is also shown in Fig. 11:

- Age: $min= 21$, $avg= 37.7$, $max= 61$
- Years of Experience: $min= 1$, $avg= 12.2$, $max= 35$
- Gender: 2 female, 13 male
- Roles: 6 controllers, 4 IT specialists, 3 managers, 2 consultants

The Tasks: Six tasks were chosen to cover *topology-based* and *attribute-based* tasks, which we “borrowed” from the field of graph visualization [78]. While being a completely different area of visualization, graph visualization tasks work well in our case, as Parallel Hierarchies include topology (the hierarchies and the structure defined by the ribbons) and numerical attributes (the aggregate values). Thus choosing tasks, which demand to traverse the topology and to identify and compare

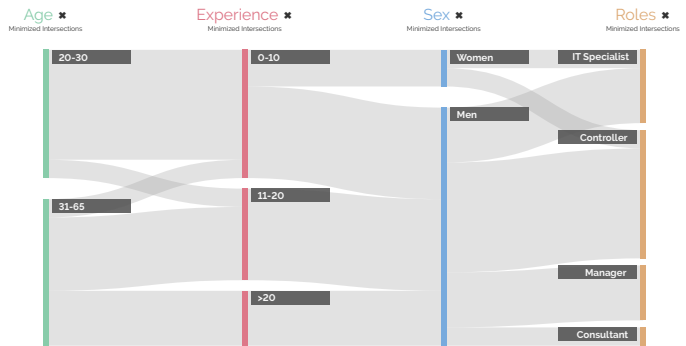


Fig. 11. Structural break-down of the group of study participants.

numerical attributes in any combination, seems to provide a good sample of possible tasks performed with Parallel Hierarchies in the wild. After formulating the tasks, all of the tasks were reviewed by project members who are experts in the field of product costing to make sure that they are granulated enough and reflect the users’ daily base tasks within their current product costing application. Concretely, the six tasks were:

- T1 Which country does the main part of the *Drive* come from?
- T2 What is the price range of most *Shaft* sub-items? And from which country are most of the items in that price range?
- T3 Which item among those with a *manual* price source has the most sub-items?
- T4 Which component split has only *variable* cost portions?
- T5 What percentage of the cost for the *Casing* comes from *Overheads* component split?
- T6 What percentage of the total cost stems from the *fixed* cost portion?

The domain language masks in particular the topology-based nature of some tasks. Yet, for example, it is evident that T2 requires a traversal of ribbons or that T3 requires a drill-down into the hierarchy to identify the most sub-items which cannot be gleaned from the height of the bars as these encode their cost and not their quantity.

6.2. Procedure of the Evaluation

The goal of our study was to check for comprehension and interaction hurdles with the visual representation, as well as to observe how users without prior visualization knowledge actually use Parallel Hierarchies and what they think of it. To that end, our study followed a defined procedure that consisted of five steps:

1. background questions to establish participants’ levels of experience and corporate role;
2. explanation and exemplification of the Parallel Hierarchies visualization and its interactive features;
3. practical warm-up for the participants to familiarize themselves with the technique;
4. performing the six product costing tasks outlined above;
5. wrap-up questionnaire to gather overall user experience indicators and open feedback.

Data was gathered through semi-structured interviews [79] for step 1, through think-aloud protocol [80] for the practical steps 2 to 4, and through the standardized User Experience Questionnaire (UEQ) [81] for step 5. Questions arising during the practical steps were noted down and answered, as helping participants does not introduce bias to their overall experience, since accuracy and timing of their task performance were not measured.

6.3. Results from the Evaluation

The evaluation yielded three types of results: our observations while the participants were solving the tasks, results from the UEQ questionnaire, and the users' free-form feedback.

6.3.1. Observations

Our first observation was an apparent relation between age and learning curve, by which we mean "how fast the user will learn the set of skills required to perform tasks with a given visualization" [82]. After dividing the participants into two age groups, we observed that the first group of 7 participants (from 20 to 30 years old) performed the tasks with more ease and confidence than the second group of 8 participants (from 31 to 65 years old). Where younger participants had no major problems using Parallel Hierarchies after our short 5-minute explanation, older participants still needed a lot of guidance in using it.

This observation is underlined by fact that participants from the younger age group asked for help during one task on average, whereas participants from the older age group required our help during 2.5 tasks on average. Surprisingly this observation was indeed aligned with the age of the participants and not with the years of experience – i.e., it did not really matter for how long a participant had already been working with the current tools of the trade and there was apparently nothing they needed to "unlearn" first to be able to learn the new visualization.

The participants' questions with regard to Parallel Hierarchies were on one hand geared towards understanding the visual mapping – particularly the meaning of the ribbons. Most participants first tried to solve the tasks by looking at and comparing the heights of the active subcategories, which makes sense given their familiarity with bar charts. Only when that did not work they invested the extra efforts of switching their mental map to parse and trace the unfamiliar ribbons. One of the participants mentioned that "*First I do my best to find the answers by the bars, then tool-tip information. If none of them works, then I will try to understand the flows.*"

Questions with respect to the interactive adaptation of Parallel Hierarchies came usually up when the participants were expected to adjust the hierarchy levels using drill-down and roll-up operations, but did not realize this possibility or what could be gained from it. This was mainly the case when working on task T3, where most participants got stuck when trying to solve it with a singular view – i.e., they were trying to find one perfect view that answered the question. Yet this was not possible in this case, as to solve this task they were required to drill-down and roll-up on three different subcategories of the "Cost Item" axis, and then to compare the values they found for each subcategory. This indicates that the participants perceived drill-down and roll-up mainly for adjusting the visualization until the sought information comes into view, which can then be

Dimension	Avg. Value	Std. Error	Alpha
Attractiveness	2.095	0.770	0.86
Perspicuity	1.893	0.944	0.87
Dependability	1.946	1.253	0.75
Efficiency	2.054	0.701	0.74
Stimulation	2.000	0.679	0.80
Novelty	2.482	0.616	0.79

Table 1. Overall UEQ test results from our study.

analyzed. To use them also as part of analysis itself by continuously going back and forth between hierarchy levels to form a larger picture of an insight that cannot be pinpointed on a single level of detail was not part of their repertoire of analysis strategies. In particular this latter observation is highly promising for Parallel Hierarchies, as it opens up a way to gain such insights in this particular domain.

6.3.2. UEQ Results:

The UEQ test uses 26 adjective pairs that are assigned to six user experience factors: attractiveness, perspicuity, efficiency, dependability, stimulation, and novelty. Each adjective pair (e.g., from *attractive* to *unattractive*, from *predictable* to *unpredictable*, or from *easy to learn* to *difficult to learn*) uses a seven point Likert scale where the polarity is determined randomly for each pair. Table 1 shows the averaged results for the six user experience factors in a range between -3 (negative) to $+3$ (positive). Overall, all factors received a value around $+2$ with only novelty being ranked slightly higher. This relative outlier is probably due to having conducted this study with non-experts in the field of visualization, who may not be as current on the visualization state-of-the-art.

These results by themselves give a general indication that Parallel Hierarchies are deemed useful and appealing by the participants. In comparison to the benchmark dataset which currently contains UEQ test data from 246 scientific and industrial user studies of software products with overall 9,905 participants [83], Parallel Hierarchies rank for most factors among the top 10% of the studies. Only for perspicuity, Parallel Hierarchies is in the second tier of 75% – 90% and thus among the top 25% of the products included in the benchmark dataset. While the UEQ test results are difficult to interpret just by themselves, they will allow comparing future enhancements and alterations of the Parallel Hierarchies technique to this baseline.

6.3.3. Feedback

From the free-form feedback given by the participants, two main themes emerged: 6 out of the 15 participants suggested to show a table on the side of the view with further detail information and 7 out of the 15 participants wanted to have the possibility to export snapshots for reporting purposes.

The first suggestion of adding a tabular display was somewhat counter intuitive at first, as the participants had already struggled with all the visual details that go well beyond their common bar chart. Yet it also confirms what we had established in the beginning, namely that costing experts are familiar with spreadsheets and tables. It seems that at least until this user

group has gained a good understanding of what Parallel Hierarchies can and cannot do for them, they feel uneasy with just the visualization and no means to look at the numbers behind it. Thus, adding such a table as a way to ease the transition from their accustomed software tools to Parallel Hierarchies seems like a good path to establish Parallel Hierarchies in this domain.

The second suggestion of being able to take snapshots further underlines this aspect: being asked about this rather common feature request, the participants revealed that they did not see themselves using Parallel Hierarchies in their day to day costing analyses. For their daily analyses, they have their standard tool chain with which they are familiar and which is deeply embedded in their companies' IT infrastructure and general workflows. Instead, they wanted to use Parallel Hierarchies mainly for communication and presentation purposes – in particular with “the higher-ups” who did not understand or care for complicated spreadsheets. They believed that Parallel Hierarchies are a perfect way to break down their analyses for the decision makers. One of the participants who works as a controller said “*I need exactly something like this visualization to communicate my discoveries in the dataset with my managers for future cost optimization decisions*”. The most popular idea brought forth by 1/3 of the participants was to have an iPad version of the visualization, which would preserve the interactivity of the technique. This way, they could even investigate different costing alternatives together with the decision makers using a point&click interface.

In addition, we asked the participants for the top 3 use cases that they could imagine using the visualization for in their daily work. From the responses, we gathered overall 24 different use cases from the costing domain alone. In particular participants with management and consulting roles were eager to use the visualization for their daily tasks. Three companies partaking in the SAP PLC co-development program and thus being introduced to Parallel Hierarchies showed particular interest in the visualization, which lead to the development of a first prototype for customers from the automotive industry.

6.4. Validity of the Evaluation

When designing Parallel Hierarchies, we had one fundamental use case in mind: breaking down a numerical aggregate along multiple hierarchical categories, as one does in the costing scenario. We then evaluated Parallel Hierarchies with a rather homogeneous group of people from the product costing domain and with tasks that work towards the goal of breaking down a large cost aggregate. Their homogeneity is underlined by high Cronbach's alpha-coefficients for the UEQ results given in Table 1, which lie between 0.74 and 0.87. These indicate *acceptable to good* scale consistency among participants [84, p.109], meaning that our participants mostly agree on the interpretation of the 26 UEQ adjective pairs. The evaluation results show that for this scenario, Parallel Hierarchies work well notwithstanding certain learning and adoption hurdles.

Yet when discussing Parallel Hierarchies with the biologists in the context of the Gene Ontology use case from Section 5.2, we found that one could also use the visualization quite differently – namely as an interface for *dynamic queries* [85], *faceted*

search [86], or *cross-filtering* [87]. While being the same fundamental visualization technique, this usage context is quite different as its aim is not to break-down the item distribution across categories, but to find data subsets with very specific properties within the dataset. Because of that difference, we are reluctant to transfer our evaluation results to this very different usage scenario that comes with different requirements and tasks. Establishing the suitability of Parallel Hierarchies in this scenario will require us to conduct a separate evaluation in this particular setting that compares them to the aforementioned interfaces for querying, searching, and filtering.

Finally, it has to be noted that our evaluation did neither compare against the diagrammatic status quo in the field of product costing (i.e., pie charts and bar charts), nor was it conducted using tasks directly formulated by the users themselves. The reason was that pie charts and bar charts lack in many regards the expressiveness for certain data aspects and the effectiveness for certain tasks that Parallel Hierarchies were specifically designed to accommodate. For example, pie charts and bar charts are by themselves not able to express hierarchical data structures and they are thus not effective for pursuing tasks related to navigating the hierarchical categorizations. Hence a comparison between them would have been a rather one-sided endeavour that either used basic tasks that can be accomplished without any of the advanced features that Parallel Hierarchies provide, or that used advanced tasks that cannot be achieved with basic charts. To avoid such an apple-to-oranges comparison, we decided to evaluate Parallel Hierarchies in a stand-alone fashion, as this allowed us to put all of its capabilities to a test. This includes many capabilities that the users did not know about beforehand and for which they could have hardly defined tasks by themselves up front. For a balanced comparison, an evaluation against Hierarchical Chord Diagrams (cf. Fig. 2(d)) could be conducted in the future, albeit these are equally unfamiliar to our users from product costing.

7. Scalability Considerations

Parallel Hierarchies have been designed with scalability in mind, which is realized by means of the simplified Icicle plots that reduce the display of a full hierarchy to only the ancestors of an active subcategory and its immediate children. This design decision alone leads to a clearer view of the currently focused categories, as one can see by comparing Fig. 1 and Fig. 12, where the latter basically illustrates how Fig. 1 would have looked like, if it was not for the simplified Icicle plots. Yet as with any visualization, after surpassing a certain data size Parallel Hierarchies become ill-suited for that data's representation. While our visual and algorithmic design aims to push this point as far out as possible, at some point Parallel Hierarchies reaches a limit of axes and categories that it can no longer faithfully show.

As for its **visual scalability**, Parallel Hierarchies' sweet spot lies according to our experience at 3-5 axes showing hierarchies that are 3-5 levels deep with a branching factor of 2-10 subcategories per category. Already displays of 6-7 axes become hard to read and to interpret, given that no duplicate axes are among

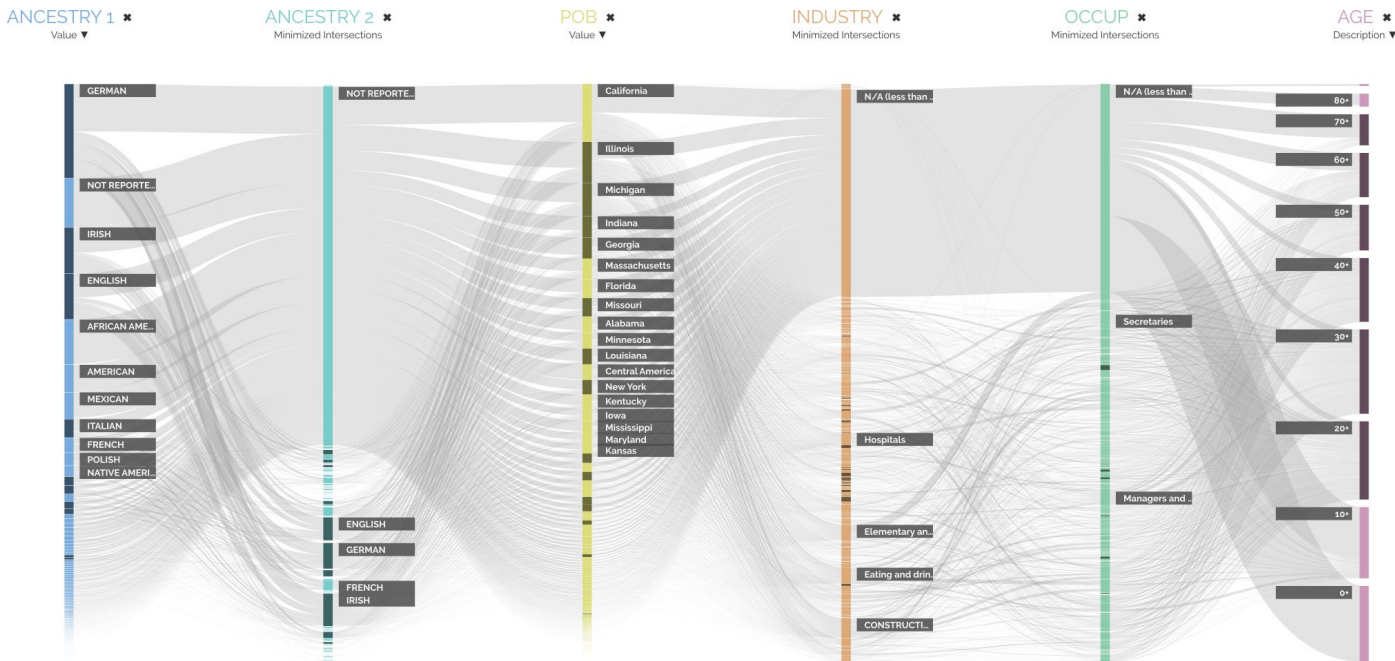


Fig. 12. Flat view of the dataset shown in Fig. 1. The axes have been drilled down to the respective levels of granularity shown in Fig. 1, yet they are displayed without the stacked ancestors and without condensing the siblings, effectively producing a Parallel Sets-like view.

them. We nevertheless decided not to restrict the number of axes, as in some cases one may want to generate wide mural-like Parallel Hierarchy visualizations for illustration purposes, for example. This also leaves the possibility to integrate a more advanced interactive axes management in the future, such as the *dimension composition* known from Parallel Sets [31] or the *hierarchical grouping of dimensions* employed by Aggregate Parallel Coordinates [88].

While deep hierarchies of 10 or more levels are merely tedious to explore as they require many drill-down interactions, wide hierarchies with a branching factor > 20 will turn the visualization indecipherable as the ribbons get thinner. This makes them harder to trace across the visualization, to select or hover over with the mouse, as well as too small to be labeled. To enable the interactive exploration even of these thinner ribbons, we have added an optional fisheye distortion in our Parallel Hierarchies implementation that enlarges small categories and thin ribbons underneath the mouse cursor for simpler selection and temporary label placement.

Lastly, there is the issue of the ribbon clutter in between axes that is left even after applying crossing minimization and re-ordering of axes. To ease tracing ribbons of interest across these cluttered regions, we are currently investigating a suggestion made by one of the demographers who worked with us in the example from Section 5.1. She proposed to completely remove all unrelated ribbons from the view when hovering over a sub-category or ribbon with the mouse. It remains to be seen for which analytic tasks how much of the contextual information provided by the other ribbons is actually necessary, or whether these can indeed just be temporarily cleared while focusing on a category/ribbon of interest.

While not pursued here, common filtering techniques can

easily be added onto the base visualization, such as showing only ribbons above a certain threshold to see the larger trends, or below a certain threshold to see the outliers. Features like these would help to scale the visualization to more and thus smaller categories with even thinner ribbons connecting them, by removing them or putting them in focus, respectively.

As for its **algorithmic scalability**, which includes runtime and memory issues, our browser-based implementation can handle datasets with up to 100k items while maintaining the necessary responsiveness for an exploratory analysis. The main limiting technical reasons are (1) that all computations are currently done entirely on the client side – from parsing, checking, and processing the data to computing the splines for the ribbons; and (2) that we use an SVG-based rendering through the D3 visualization library [89], which inflates the DOM tree to the point where it becomes too complex even for simple jQuery operations. To scale-up our solution to larger datasets, we could follow the current trend in web-based parallel coordinate displays to use an Apache Spark back-end on the server side in combination with WebGL rendering [90, 91]. Another approach would be to utilize a hierarchical data format that follows the hierarchies defined over the categorical values and allows us to selectively load top-down only those pre-aggregated portions of the data that are currently in view.

8. Summary and Outlook

Parallel Hierarchies present a unique way of displaying and exploring categorical aggregates. Its combination of tree visualization elements and set visualization elements in the same display space allows for a rich interaction with hierarchical and categorical aspects of the data at the same time. This interaction

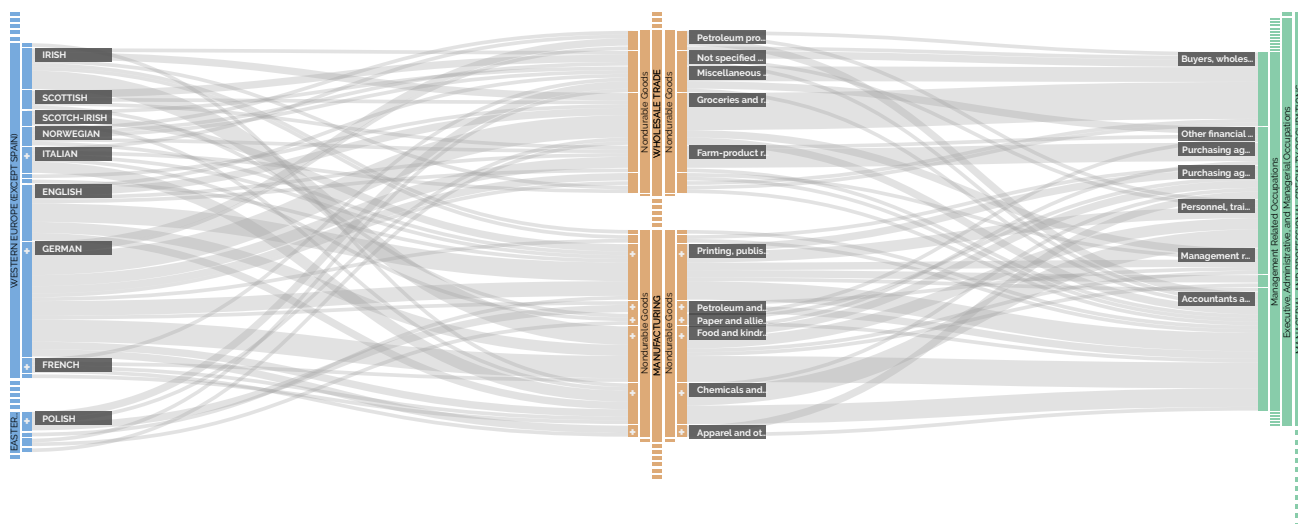
ANCESTRY 1
Description ▾INDUSTRY
Description ▾OCCUP
Description ▾

Fig. 13. Example of the multi-focus selection. Along the left and center axes, two hierarchies are shown. On the leftmost axis, the categories “Western Europe” and “Eastern Europe” are both selected, thus showing all people of European descent. In the center axis the “Durable Goods” category appears in two separate branches, which are both selected. As this example shows, selecting multiple categories on the same axis allows for more complex queries and offers more flexibility when working with complex hierarchies.

can be utilized for a variety of analysis goals: to drill-down into large datasets to find data items with particular characteristics, to identify data items that contribute most or least to a given aggregate, or to trace a subset of data items from a particular category across multiple properties to see how they distribute.

Our plans for future work will pick up where this paper leaves off and explore extensions of the presented base visualization. This includes the possibility of focusing not only at one category at a time (currently via mouse over), but at two or more categories on the same axis. A first example that shows the utility of such a multi-focus selection is given in Fig. 13. This idea can be further expanded to better distinguish the items from multiple foci as they propagate, giving Parallel Hierarchies an entirely new spin in the direction of comparative analysis for categorical data – e.g., Fig. 13 could then show how people with Eastern European ancestors distribute across the other data properties, as compared to people with Western European ancestors. In addition, we aim to provide extensions for uncertain data along the lines of our recent work on showing uncertainties in ribbons [92], as well as for time-dependent data.

As for the practical part of getting Parallel Hierarchies more and more into current visualization practice, we are currently working with the team behind SAP Analytics Cloud, which is SAP SE’s commercial BI framework, to integrate Parallel Hierarchies as a standard diagram type. As an additional benefit from the user study with SAP customers that we reported on in Section 6, several of the study participants now got a taste of using Parallel Hierarchies and have since requested its integration as a product feature in the SAP Product Life-cycle Costing suite.

Acknowledgments

The authors thank Jochen Rode, Stefan Hesse, Martin Luboschik, and the anonymous reviewers for their support and suggestions that helped in shaping this paper into its final form. Thanks also to Isa Usmanov for implementing the first prototype and to Esther Lapczynya for advising us on graphic design aspects of Parallel Hierarchies. Furthermore, we thank all attendees of the SAP PLC co-development workshops, the biologists at the CZ Biohub, and the demographers from the Max Planck Institute for Demographic Research for partaking in different user studies and feedback sessions.

References

- [1] Bureau of Labour Statistics, (BLS). Standard Occupational Classification Manual. US Office of Employment and Unemployment Statistics; 2018.
- [2] Chan, LM, Mitchell, JS. Dewey Decimal Classification: Principles and Application. 3rd ed.; OCLC; 2003.
- [3] World Health Organization, (WHO). International Statistical Classification of Diseases and Related Health Problems, 10th Revision. 5th ed.; WHO Press; 2016.
- [4] Graham, M, Kennedy, J. A survey of multiple tree visualisation. *Information Visualization* 2010;9(4):235–252. doi:10.1057/ivs.2009.29.
- [5] Holten, D, van Wijk, JJ. Visual comparison of hierarchically organized data. *Computer Graphics Forum* 2008;27(3):759–766. doi:10.1111/j.1467-8659.2008.01205.x.
- [6] Telea, A, Auber, D. Code Flows: Visualizing structural evolution of source code. *Computer Graphics Forum* 2008;27(3):831–838. doi:10.1111/j.1467-8659.2008.01214.x.
- [7] Johansson, S, Jern, M, Johansson, J. Interactive quantification of categorical variables in mixed data sets. In: *Proc. of the International Conference Information Visualisation, (IV’08. IEEE; 2008, p. 3–10. doi:10.1109/IV.2008.33.*

- [8] Guha, S, Rastogi, R, Shim, K. Rock: A robust clustering algorithm for categorical attributes. *Information Systems* 2000;25(5):345–366. doi:10.1016/S0306-4379(00)00022-3.
- [9] Ahl, V, Allen, T. *Hierarchy Theory: A Vision, Vocabulary, and Epistemology*. Columbia University Press; 1996.
- [10] Pumain, D, editor. *Hierarchy in Natural and Social Sciences*. Springer; 2006. doi:10.1007/1-4020-4127-6.
- [11] Furnas, GW, Zacks, J. Multitrees: Enriching and reusing hierarchical structure. In: *Proc. of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'94)*. ACM; 1994, p. 330–336. doi:10.1145/191666.191778.
- [12] Robertson, G, Cameron, K, Czerwinski, M, Robbins, D. Polyarchy visualization: Visualizing multiple intersecting hierarchies. In: *Proc. of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'02)*. ACM; 2002, p. 423–430. doi:10.1145/503376.503452.
- [13] Elmqvist, N, Fekete, JD. Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *IEEE Trans on Visualization and Computer Graphics* 2010;16(3):439–454. doi:10.1109/TVCG.2009.84.
- [14] Monroe, M, Lan, R, Lee, H, Plaisant, C, Shneiderman, B. Temporal event sequence simplification. *IEEE Trans on Visualization and Computer Graphics* 2013;19(12):2227–2236. doi:10.1109/TVCG.2013.200.
- [15] Malik, S, Du, F, Monroe, M, Onukwughu, E, Plaisant, C, Shneiderman, B. Cohort comparison of event sequences with balanced integration of visual analytics and statistics. In: *Proc. of the International Conference on Intelligent User Interfaces (IUI'15)*. ACM; 2015, p. 38–49. doi:10.1145/2678025.2701407.
- [16] Stoffel, F, Janetzko, H, Mansmann, F. Proportions in categorical and geographic data: Visualizing the results of political elections. In: *Proc. of the International Working Conference on Advanced Visual Interfaces (AVI'12)*. ACM; 2012, p. 457–464. doi:10.1145/2254556.2254644.
- [17] Schulz, HJ, Hadlak, S, Schumann, H. A visualization approach for cross-level exploration of spatiotemporal data. In: *Proc. of the International Conference on Knowledge Management and Knowledge Technologies (i-Know'13)*. ACM; 2013, p. 2:1–2:8. doi:10.1145/2494188.2494199.
- [18] Inselberg, A. *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer; 2009. doi:10.1007/978-0-387-68628-8.
- [19] Heinrich, J, Weiskopf, D. State of the art of parallel coordinates. In: *Eurographics 2013 - State of the Art Reports*. Eurographics; 2013, p. 95–116. doi:10.2312/conf/EG2013/stars/095-116.
- [20] Johansson, J, Forsell, C. Evaluation of parallel coordinates: Overview, categorization and guidelines for future research. *IEEE Trans on Visualization and Computer Graphics* 2016;22(1):579–588. doi:10.1109/TVCG.2015.2466992.
- [21] Collins, C, Viégas, FB, Wattenberg, M. Parallel tag clouds to explore and analyze faceted text corpora. In: *Proc. of the IEEE Symposium on Visual Analytics Science and Technology (VAST'09)*. IEEE; 2009, p. 91–98. doi:10.1109/VAST.2009.5333443.
- [22] Johansson, J, Ljung, P, Cooper, M. Depth cues and density in temporal parallel coordinates. In: *Proc. of the Eurographics/ IEEE-VGTC Symposium on Visualization (EuroVis'07)*. Eurographics Association; 2007, p. 35–42. doi:10.2312/VisSym/EuroVis07/035-042.
- [23] Ghani, S, Kwon, BC, Lee, S, Yi, JS, Elmqvist, N. Visual analytics for multimodal social network analysis: A design study with social scientists. *IEEE Trans on Visualization and Computer Graphics* 2013;19(12):2032–2041. doi:10.1109/TVCG.2013.223.
- [24] Freiler, W, Matković, K, Hauser, H. Interactive visual analysis of set-typed data. *IEEE Trans on Visualization and Computer Graphics* 2008;14(6):1340–1347. doi:10.1109/TVCG.2008.144.
- [25] Alsallakh, B, Micallef, L, Aigner, W, Hauser, H, Miksch, S, Rodgers, P. The state-of-the-art of set visualization. *Computer Graphics Forum* 2016;35(1):234–260. doi:10.1111/cgf.12722.
- [26] Rusu, A. Tree drawing algorithms. In: Tamassia, R, editor. *Handbook of Graph Drawing and Visualization*. CRC Press; 2013, p. 155–192. doi:10.1201/b15385.
- [27] Schulz, HJ. *Treevis.net: A Tree Visualization Reference*. *IEEE Computer Graphics & Applications* 2011;31(6):11–15. doi:10.1109/MCG.2011.103.
- [28] Blasius, J, Greenacre, M, editors. *Visualization of Categorical Data*. Academic Press; 1998. doi:10.1016/B978-0-12-299045-8.X5000-1.
- [29] Friendly, M. *Visualizing Categorical Data*. SAS Institute; 2000.
- [30] Bendix, F, Kosara, R, Hauser, H. Parallel Sets: Visual analysis of categorical data. In: *Proc. of the IEEE Symposium on Information Visualization (InfoVis'05)*. IEEE; 2005, p. 133–140. doi:10.1109/INFVIS.2005.1532139.
- [31] Kosara, R, Bendix, F, Hauser, H. Parallel Sets: Interactive exploration and visual analysis of categorical data. *IEEE Trans on Visualization and Computer Graphics* 2006;12(4):558–568. doi:10.1109/TVCG.2006.76.
- [32] Brinton, WC. *Graphic Presentation*. Brinton Associates; 1939.
- [33] Strickland, J. *Cosmograph? What's a Cosmograph?* *Computer History Museum Volunteer Information Exchange* 2012;2(16):2–3.
- [34] Huang, ML, Huang, TH, Zhang, X. A novel virtual node approach for interactive visual analytics of big datasets in parallel coordinates. *Future Generation Computer Systems* 2016;55:510–523. doi:10.1016/j.future.2015.02.003.
- [35] Candan, KS, Di Caro, L, Sapino, ML. PhC: Multiresolution visualization and exploration of text corpora with parallel hierarchical coordinates. *ACM Trans on Intelligent Systems and Technology* 2012;3(2):22:1–22:36. doi:10.1145/2089094.2089098.
- [36] Fua, YH, Ward, MO, Rundensteiner, EA. Hierarchical parallel coordinates for exploration of large datasets. In: *Proc. of the IEEE Conference on Visualization (Vis'99)*. IEEE; 1999, p. 43–50. doi:10.1109/VISUAL.1999.809866.
- [37] Argyriou, EN, Symvonis, A, Vassiliou, V. A fraud detection visualization system utilizing radial drawings and heat-maps. In: *Proc. of the International Conference on Information Visualization Theory and Applications (IVAPP'14)*. IEEE; 2014, p. 153–160. doi:10.5220/0004735501530160.
- [38] Arleo, A, Didimo, W, Liotta, G, Montecchiani, F. GiViP: A visual profiler for distributed graph processing systems. In: *Proc. of the International Symposium on Graph Drawing and Network Visualization (GD'17)*. Springer; 2018, p. 256–271. doi:10.1007/978-3-319-73915-1_21.
- [39] Stasko, J, Zhang, E. Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In: *Proc. of the IEEE Symposium on Information Visualization (InfoVis'00)*. IEEE; 2000, p. 57–65. doi:10.1109/INFVIS.2000.885091.
- [40] Alsallakh, B, Aigner, W, Miksch, S, Gröller, E. Reinventing the Contingency Wheel: Scalable visual analytics of large categorical data. *IEEE Trans on Visualization and Computer Graphics* 2012;18(12):2849–2858. doi:10.1109/TVCG.2012.254.
- [41] Holten, D. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Trans on Visualization and Computer Graphics* 2006;12(5):741–748. doi:10.1109/TVCG.2006.147.
- [42] Keahey, TA, Rope, DJ, Wills, GJ. Generating an outside-in hierarchical tree visualization, Patent Application US 20180046690 A1, filed August 12, 2016, published February 15, 2018.
- [43] Guchev, V, Mecella, M, Santucci, G. Design guidelines for correlated quantitative data visualizations. In: *Proc. of the International Working Conference on Advanced Visual Interfaces (AVI'12)*. ACM; 2012, p. 761–764. doi:10.1145/2254556.2254703.
- [44] Fernstad, SJ, Johansson, J. A task based performance evaluation of visualization approaches for categorical data analysis. In: *Proc. of the International Conference on Information Visualisation (IV'11)*. IEEE; 2011, p. 80–89. doi:10.1109/IV.2011.92.
- [45] Zeileis, A, Hornik, K, Murrell, P. Escaping RGBland: Selecting colors for statistical graphics. *Computational Statistics & Data Analysis* 2009;53(9):3259–3270. doi:10.1016/j.csda.2008.11.033.
- [46] Kruskal, JB, Landwehr, JM. Icicle plot: Better displays for hierarchical clustering. *The American Statistician* 1983;37(2):162–168. doi:10.2307/2685881.
- [47] Cox, NJ. Speaking Stata: Spineplots and their kin. *The Stata Journal* 2016;8(1):105–121.
- [48] Plaisant, C, Grosjean, J, Bederson, BB. SpaceTree: Supporting exploration in large node link tree, design evolution and empirical evaluation. In: *Proc. of the IEEE Symposium on Information Visualization (InfoVis'02)*. IEEE; 2002, p. 57–64. doi:10.1109/INFVIS.2002.1173148.
- [49] Band, Z, White, RW. PygmyBrowse: A small screen tree browser. In: *Extended abstracts of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'06)*. ACM; 2006, p. 514–519. doi:10.1145/1125451.1125562.

- [50] Nielsen, M, Grønþæk, K. PivotViz: Interactive visual analysis of multidimensional library trans. data. In: Proc. of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'15). ACM; 2015, p. 139–142. doi:10.1145/2756406.2756937.
- [51] Palmas, G, Bachynskiy, M, Oulasvirta, A, Seidel, HP, Weinkauff, T. An edge-bundling layout for interactive parallel coordinates. In: Proc. of the IEEE Pacific Visualization Symposium (PacificVis'14). IEEE; 2014, p. 57–64. doi:10.1109/PacificVis.2014.40.
- [52] Richer, G, Sansen, J, Lalanne, F, Auber, D, Bourqui, R. Enabling hierarchical exploration for large-scale multidimensional data with abstract parallel coordinates. In: Proc. of the Workshops of the EDBT/ICDT 2018 Joint Conference. CEUR-WS; 2018, p. 76–83. URL: <http://ceur-ws.org/Vol-2083/paper-12.pdf>.
- [53] Siirtola, H, Riih , KJ. Interacting with parallel coordinates. *Interacting with Computers* 2006;18(6):1278–1309. doi:10.1016/j.intcom.2006.03.006.
- [54] Ware, C. *Information Visualization – Perception for Design*. 3rd ed.; Morgan Kaufmann; 2013. doi:10.1016/C2009-0-62432-6.
- [55] Tol, P. Colour schemes. Tech. Rep. SRON/EPS/TN/09-002 v.2.2; SRON Netherlands Institute for Space Research; 2012.
- [56] Ma, S, Hellerstein, JF. Ordering categorical data to improve visualization. In: Late Breaking Hot Topics of the IEEE Information Visualization Symposium (InfoVis'99). IEEE; 1999, p. 15–18.
- [57] Beygelzimer, A, Perng, CS, Ma, S. Fast ordering of large categorical datasets for better visualization. In: Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'01). ACM; 2001, p. 239–244. doi:10.1145/502512.502545.
- [58] Rosario, GE, Rundensteiner, EA, Brown, DC, Ward, MO, Huang, S. Mapping nominal values to numbers for effective visualization. *Information Visualization* 2004;3(2):80–95. doi:10.1057/palgrave.ivs.9500072.
- [59] Andrienko, N, Andrienko, G. *Exploratory Analysis of Spatial and Temporal Data – A Systematic Approach*. Springer; 2006. doi:10.1007/3-540-31190-4.
- [60] Johansson, S, Johansson, J. Visual analysis of mixed data sets using interactive quantification. *SIGKDD Explorations Newsletter* 2009;11(2):29–38. doi:10.1145/1809400.1809406.
- [61] Eades, P, Whitesides, S. Drawing graph in two layers. *Theoretical Computer Science* 1994;131(2):361–374. doi:10.1016/0304-3975(94)90179-1.
- [62] Sugiyama, K, Tagawa, S, Toda, M. Methods for visual understanding of hierarchical system structures. *IEEE Trans on Systems, Man, and Cybernetics* 1981;11(2):109–125. doi:10.1109/TSMC.1981.4308636.
- [63] J nger, M, Mutzel, P. 2-layer straightline crossing minimization: Performance of exact and heuristic algorithms. *Journal of Graph Algorithms and Applications* 1997;1(1):1–25. doi:10.7155/jgaa.00001.
- [64] Liu, S, Wu, Y, Wei, E, Liu, M, Liu, Y. StoryFlow: Tracking the evolution of stories. *IEEE Trans on Visualization and Computer Graphics* 2013;19(12):2436–2445. doi:10.1109/TVCG.2013.196.
- [65] Gad, S, Javed, W, Ghani, S, Elmqvist, N, Ewing, T, Hampton, KN, et al. ThemeDelta: Dynamic segmentations over temporal topic models. *IEEE Trans on Visualization and Computer Graphics* 2015;21(5):672–685. doi:10.1109/TVCG.2014.2388208.
- [66] M kinen, E. Experiments on drawing 2-level hierarchical graphs. *International Journal of Computer Mathematics* 1990;37(3-4):129–135. doi:10.1080/00207169008803941.
- [67] Perin, C, Boy, J, Vernier, F. Using gap charts to visualize the temporal evolution of ranks and scores. *IEEE Computer Graphics and Applications* 2016;36(5):38–49. doi:10.1109/MCG.2016.100.
- [68] Hofmann, H, Vendettuoli, M. Common angle plots as perception-true visualizations of categorical associations. *IEEE Trans on Visualization and Computer Graphics* 2013;19(12):2297–2305. doi:10.1109/TVCG.2013.140.
- [69] VanderPlas, S, Hofmann, H. Signs of the sine illusion – why we need to care. *Journal of Computational and Graphical Statistics* 2015;24(4):1170–1190. doi:10.1080/10618600.2014.951547.
- [70] Byron, L, Wattenberg, M. Stacked graphs – geometry & aesthetics. *IEEE Trans on Visualization and Computer Graphics* 2008;14(6):1245–1252. doi:10.1109/TVCG.2008.166.
- [71] Ellis, G, Dix, A. Enabling automatic clutter reduction in parallel coordinate plots. *IEEE Trans on Visualization and Computer Graphics* 2006;12(5):717–724. doi:10.1109/TVCG.2006.138.
- [72] Lex, A, Streit, M, Schulz, HJ, Partl, C, Schmalstieg, D, Park, PJ, et al. StratomeX: Visual analysis of large-scale heterogeneous genomics data for cancer subtype characterization. *Computer Graphics Forum* 2012;31(3):1175–1184. doi:10.1111/j.1467-8659.2012.03110.x.
- [73] Sarikaya, A, Correli, M, Dinis, JM, O'Connor, DH, Gleicher, M. Visualizing cooccurrence of events in populations of viral genome sequences. *Computer Graphics Forum* 2016;35(3):151–160. doi:10.1111/cgf.12891.
- [74] Vosough, Z, Groh, R, Schulz, HJ. On establishing visualization requirements: A case study in product costing. In: Short Paper Proc. of the Eurographics Conference on Visualization (EuroVis'17). Eurographics Association; 2017, p. 97–101. doi:10.2312/eurovisshort.20171140.
- [75] Johnson, B, Shneiderman, B. Tree-Maps: A space-filling approach to the visualization of hierarchical information structures. In: Proc. of the IEEE Conference on Visualization (Vis'91). IEEE; 1991, p. 284–291. doi:10.1109/VISUAL.1991.175815.
- [76] Riehmann, P, Hanfler, M, Froehlich, B. Interactive Sankey diagrams. In: Proc. of the IEEE Symposium on Information Visualization (InfoVis'05). IEEE; 2005, p. 233–240. doi:10.1109/INFVIS.2005.1532152.
- [77] Tory, M, M ller, T. Human factors in visualization research. *IEEE Trans on Visualization and Computer Graphics* 2004;10(1):72–84. doi:10.1109/TVCG.2004.1260759.
- [78] Lee, B, Plaisant, C, Parr, CS, Fekete, JD, Henry, N. Task taxonomy for graph visualization. In: Proc. of AVI Workshop “BEyond Time and Errors: Novel Evaluation Methods for Information Visualization” (BELIV'06). ACM Press; 2006, p. 1–5. doi:10.1145/1168149.1168168.
- [79] Lindlof, TR, Taylor, BC. *Qualitative Communication Research Methods*. 3rd ed.; SAGE Publications; 2011.
- [80] Boren, MT, Ramey, J. Thinking aloud: reconciling theory and practice. *IEEE Trans on Professional Communication* 2000;43(3):261–278. doi:10.1109/47.867942.
- [81] Laugwitz, B, Held, T, Schrepp, M. Construction and evaluation of a user experience questionnaire. In: Proc. of the Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society (USAB'08). Springer; 2008, p. 63–76. doi:10.1007/978-3-540-89350-9_6.
- [82] Lall , S, Conati, C, Carenini, G. Prediction of individual learning curves across information visualizations. *User Modeling and User-Adapted Interaction* 2016;26(4):307–345. doi:10.1007/s11257-016-9179-5.
- [83] Schrepp, M, Hinderks, A, Thomaschewski, J. Construction of a benchmark for the user experience questionnaire (UEQ). *International Journal of Interactive Multimedia and Artificial Intelligence* 2017;4(4):40–44. doi:10.9781/ijimai.2017.445.
- [84] DeVellis, RF. *Scale development: Theory and applications*. 3rd ed.; Sage; 2012.
- [85] Shneiderman, B. Dynamic queries for visual information seeking. *IEEE Software* 1994;11(6):70–77. doi:10.1109/52.329404.
- [86] Tunkelang, D. *Faceted Search*. Morgan and Claypool Publishers; 2009. doi:10.2200/S00190ED1V01Y200904ICR005.
- [87] Weaver, C. Cross-filtered views for multidimensional visual analysis. *IEEE Trans on Visualization and Computer Graphics* 2010;16(2):192–204. doi:10.1109/TVCG.2009.94.
- [88] Andrews, K, Osmi , M, Schagerl, G. Aggregated parallel coordinates: Integrating hierarchical dimensions into parallel coordinates visualisations. In: Proc. of the International Conference on Knowledge Technologies and Data-driven Business (i-Know'15). ACM; 2015, p. 37:1–37:4. doi:10.1145/2809563.2809588.
- [89] Bostock, M, Ogievetsky, V, Heer, J. D3: Data-driven documents. *IEEE Trans on Visualization and Computer Graphics* 2011;17(12):2301–2309. doi:10.1109/TVCG.2011.185.
- [90] Heinrich, J, Broeksema, B. Big data visual analytics with parallel coordinates. In: Proc. of Big Data Visual Analytics (BDVA'15). IEEE; 2015, p. 121–122. doi:10.1109/BDVA.2015.7314286.
- [91] Sansen, J, Richer, G, Jourde, T, Lalanne, F, Auber, D, Bourqui, R. Visual exploration of large multidimensional data using parallel coordinates on big data infrastructure. *Informatics* 2017;4(3):21:1–21:22. doi:10.3390/informatics4030021.
- [92] Vosough, Z, Kammer, D, Keck, M, Groh, R. Visualizing uncertainty in flow diagrams: A case study in product costing. In: Proc. of the International Symposium on Visual Information Communication and Interaction (VINCI'17). ACM; 2017, p. 1–8. doi:10.1145/3105971.3105972.