A Model for Structure-based Comparison of Many Categories in Small-Multiple Displays

Johannes Kehrer, Harald Piringer, Wolfgang Berger, and M. Eduard Gröller, *Member, IEEE CS*

Abstract—Many application domains deal with multi-variate data that consist of both categorical and numerical information. Small-multiple displays are a powerful concept for comparing such data by juxtaposition. For comparison by overlay or by explicit encoding of computed differences, however, a specification of references is necessary. In this paper, we present a formal model for defining semantically meaningful comparisons between many categories in a small-multiple display. Based on pivotized data that are hierarchically partitioned by the categories assigned to the x and y axis of the display, we propose two alternatives for structure-based comparison within this hierarchy. With an *absolute reference specification*, categories are compared to a fixed reference category. With a *relative reference specification*, in contrast, a semantic ordering of the categories is considered when comparing them either to the previous or subsequent category each. Both reference specifications can be defined at multiple levels of the hierarchy (including aggregated summaries), enabling a multitude of useful comparisons. We demonstrate the general applicability of our model in several application examples using different visualizations that compare data by overlay or explicit encoding of differences.

Index Terms—Comparative visualization, small-multiple displays, Trellis displays, categorical data.

1 Introduction

Many application domains deal with multi-variate data that consist of both categorical and numerical information. A common analysis approach for such data is based on pivotization. The data are split by categorical attributes before computing aggregations of numerical attributes. The application of this concept in On-Line Analytical Processing [28] (OLAP) has established the terms *dimensions* for categorical attributes and *measures* for numerical attributes. Many visualization approaches for analyzing pivotized data rely on comparison by juxtaposition in small-multiple displays [30], also called Trellis displays [5]. Polaris [27] and the success of its commercial version Tableau [1, 2, 19] demonstrate the usefulness of such approaches. The display is subdivided into rows and columns by categories of dimensions which can also be organized hierarchically (e.g., months within years). Graphics then show the pivotized data for each combination of categories. A particular advantage of this concept is that it unifies a variety of possible comparisons between dimensions and measures [19, 27]. Also, it can be easily applied to common visualizations such as scatterplots, function graphs, or geographic maps. The visual results are easily comprehensible also for non-visualization experts.

Our work focuses on concepts for comparison of many categories in a small-multiple display. A motivation is that comparisons by juxtaposition become increasingly ineffective in case of a large number of categories. Users are forced to "remember" one graphic while shifting the focus to other potentially distant plots. This is especially challenging when comparing complex graphics with many objects [3]. Moreover, precise comparisons with respect to the horizontal position are usually limited to graphics within the same column, and precise vertical comparisons require plots to share the same row.

According to Gleicher et al. [12], there are two alternatives to comparison by *juxtaposition*: 1) data can be *overlaid* within the same frame of reference using different visual styles, and 2) computed differences between the data can be *explicitly encoded*. With comparison by overlay, data can then be compared within the same coordinate system and differences can be read-out in the data unit. Explicit encoding, in contrast, enables the analyst to compare data differences more precisely. Computed differences can be shown instead of the original data or can be encoded in addition, for example, using color. A special and common case of explicit encoding are *percentual measures* which express the relative difference of two measures.

In contrast to juxtaposition, both comparison by overlay and explicit encoding require a specification of which graphics should be compared to each other within the small-multiple display. Considering the hierarchical organization of the partitioning categories, there are many options for comparing graphics in a semantically meaningful way. For example, the months within a year can be compared to the preceding month each or to a user-defined reference month. Also, different years can be related by comparing their individual months. In this paper, we investigate the design space of exploiting the hierarchical structure of small-multiple displays for flexibly defining meaningful comparisons between graphics (referred to as *structure-based comparison*). Specifically, we present the following contributions:

- We propose a formal model for structure-based comparison of graphics, where the corresponding categories can be related both within and across multiple hierarchy levels.
- We incorporate aggregated summaries of parent categories in our model, which can be specified as references for comparison.
- We demonstrate our model through several application examples for visual comparison by overlay and explicit encoding.

2 Related Work

Comparative Visualization. Comparative visualization investigates data for similarities and differences. According to Kehrer and Hauser [17], comparison is one of six typical tasks in the visual analysis of scientific data. In this context, the combination of interaction, computational analysis, and appropriate visual encoding are highly important. Examples include the comparison of different time steps, spatial locations, variables, or data modalities. According to Verma and Pang [32], scientific data can be compared at the image, data, or feature level, depending on the degree of data abstraction. Gleicher et al. [12] proposes a complementary taxonomy for information visualization. The authors distinguish three categories, which can also be combined: 1) spatial or temporal juxtaposition, 2) superposition (or overlay), and 3) explicit encoding of computed differences. Our model supports the latter two alternatives for visual comparison in a small-multiple display.

- *J. Kehrer is with the VRVis Research Center, Vienna, and the Institute of Computer Graphics and Algorithms, Vienna University of Technology, Austria. E-mail: kehrer@cg.tuwien.ac.at*
- *H. Piringer and W. Berger are with the VRVis Research Center, Vienna, Austria. E-mail: {piringer, berger}@VRVis.at*
- *M.E. Gröller is with the Institute of Computer Graphics and Algorithms, Vienna University of Technology. E-mail: groeller@cg.tuwien.ac.at*

Data are often compared by juxtaposition using coordinated multiple views with linking and brushing [4, 25]. Lex et al. [18] propose a focus+context visualization for comparing separately clustered groups of variables of biomolecular data. Clustered records are connected across multiple groups of variables using bundled curves and ribbons. VisLink [9] draws connections between related data items in visualizations that can be placed in 3D. Malik et al. [20] discuss a multi-image view that supports the comparison of series of 3D scans from the same specimen. The approach incorporates also computed differences between the data, which is usually more precise than pure image level comparison. Tominski et al. [29] discuss interaction concepts that are inspired by real-world behavior for comparing printed information. The user can interactively specify and rearrange parts of the visualization via drag and drop. Similar to our model, data can then be compared by juxtaposition, explicit encoding, or overlay. With the latter approach, the occluded information can either shine through using alpha blending or the occluding part can be folded back and forth using a folding interaction. Additionally, computed differences can be encoded in color on-demand.

Most approaches for comparing hierarchically organized data are based on trees [14]. Graham and Kennedy [13] propose linking and focusing across multiple hierarchies. Nodes can be interactively selected and are linked to similar nodes across the tree (highlighted in color). The linking leads to multiple focal points that receive more space according to a degree-of-interest value. TreeJuxtaposer [23] allows structural comparison by finding corresponding nodes between two trees. Structural differences are automatically highlighted, and brushing helps finding similar structures in the other tree. Holten and van Wijk [16] show two trees at opposite sides of the display. Leaf nodes are matched to each other and connected by hierarchical edge bundles. Bremm et al. [7] propose similarity measures for comparing multiple trees. A matrix view gives an overview of pairwise tree similarities. A tree can be selected as a reference, which is then compared in detail to other trees. While the described approaches focus on the comparison of tree structures, we use the hierarchical categorization of the data to define which data portions should be compared.

Pivot Tables and Small-Multiple Displays. Pivot tables have long been used to summarize measures with respect to a classification given by categories. The concept of pivoting data is also important for databases, where the predominant structured query language (SQL) offers, for example, the “GROUP BY” clause for select statements. However, SQL statements have limitations with respect to drill-down and roll-up operations. Gray et al. [15], therefore, propose to treat multi-variate databases as *n*-dimensional data cubes, which have widely been adopted by On-Line Analytical Processing [28] (OLAP). OLAP systems typically use aggregated summaries as a starting point to drill down into interesting data subsets.

While most OLAP front-ends provide only selected business graphics, Polaris/Tableau [19, 27] uses a formal algebra for specifying pivot tables and their visual representation. The table algebra is based on the grammar for describing statistical graphics by Wilkinson [33]. The user can incrementally construct complex queries by intuitive manipulations in this algebra. The layout is based on small-multiple displays [5, 30]. With respect to comparison, the current version 7 of Tableau [1, 2] supports the computation of absolute and percentage differences between categories. In a dialog window, the reference can be specified either as a certain category or the previous or next element of a dimension. The comparisons, are not formally described and only the resulting differences can be displayed in Tableau. In contrast, we propose a formal model which is general enough to also support comparison by overlay.

Robertson et al. [26] study the effectiveness of small multiples, animation, and trace lines for analyzing trends in scatterplots. Their results suggest that animation is the least effective technique and small multiples are more accurate. Daae Lampe et al. [11] compare differences in movement data in a small-multiple display. Each view shows the difference to the average movement for a different category. The user can interactively drill down into subcategories. The resulting views replace the previous ones and depict the differences to the parent category. In contrast, our model can simultaneously display and compare multiple levels of the hierarchy and is not limited to comparison to aggregated summaries.

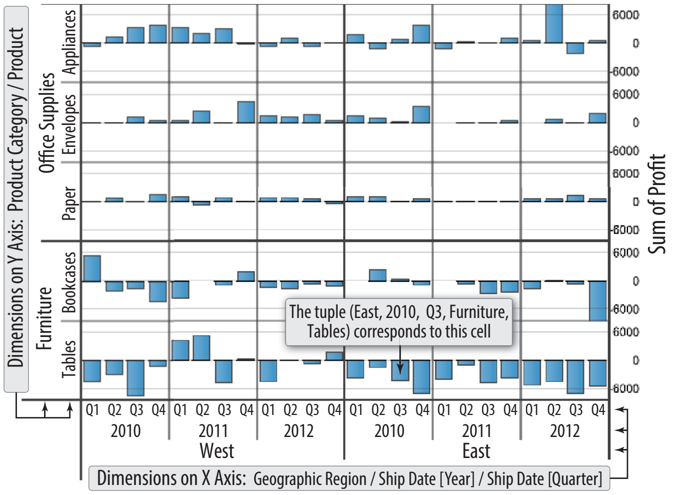


Fig. 1. Visualization of sales data. Based on the hierarchical partitioning of the small-multiple display, the data can be compared in many ways: East to West, all years to 2010, each quarter to the previous one, etc.

3 A Model for Structure-based Comparison

This section describes our model for structure-based comparison of graphics in a small-multiple display. Such a display represents a matrix of graphics which is subdivided into rows and columns according to the dimensions assigned to its axes (see Fig. 1 for an example). For each axis of the matrix, the user can define a hierarchy of categories by dimension composition. For example, categories from the dimensions “year” and “quarter” can be combined to categories like “first quarter of 2010” (see the x axis in Fig. 1). The graphics of the matrix (subsequently called *cells*) then show the data for the respective combination of categories [5, 19, 27]. In order to formally define configurations of the matrix, we build upon the table algebra by Stolte et al. [27].

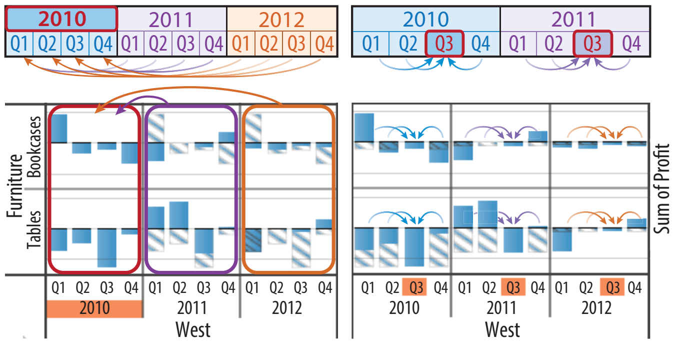
Small-multiple displays inherently facilitate comparison by juxtaposition. However, comparing different cells by overlay or explicit encoding (see the categorization of Gleicher et al. [12]) requires the specification of a *reference graph* between the cells. For large matrices, there is in theory a very large number of potential reference graphs between the cells. Some of these reference graphs are semantically meaningful for different tasks, others are not. Our goal is to exploit the hierarchical organization of the matrix in order to support a flexible definition of semantically meaningful reference graphs for comparing cells. The subsequent sections investigate the design space of structure-based comparison of cells in a matrix. Before formally describing our model, we start with an illustrative example.

3.1 Illustrative Example: Sales Data Analysis

We use the Superstore sales data which comes with the Tableau Desktop software [1] to illustrate our model throughout this section. The data consists of product orders by costumers which are described by measures such as sales, profit or unit price, categorical attributes (e.g., product category, geographic region, or customer segment) as well as the timestamps order and ship date. In Fig. 1, for example, the sum of profit is shown in a matrix that is horizontally partitioned by geographic region and ship date (year and quarter), and vertically partitioned by product category and product. Based on this example, typical comparative tasks an analyst may want to perform include:

T1 Compare the profit per quarter and product to the maximum profit in the matrix (i.e., the highest bar).

T2 Compare one geographic region to the other one, i.e., East to West.



(a) (b)

Fig. 2. Comparison to an absolute reference: (a) Each year is compared to 2010 on a quarterly basis (compare to sample task T3 in Sec. 3.1). (b) The quarters within each year are compared to a specified reference (compare to sample task T4). In the comparison by overlay, the data from the references are depicted with a hatching texture.

T3 Compare each year's profit to the profit of a particular year on a quarterly basis (per product and geographic region, see Fig. 2a).

T4 Compare each quarter's profit to a specific quarter of the same year (per product and geographic region, see Fig. 2b).

T5 Compare each year's profit to the following year's profit on a quarterly basis (per product and geographic region, see Fig. 5a).

T6 Compare each quarter's profit to the previous quarter in the same year (per product and geographic region, see Fig. 5b).

T7 Compare each quarters's profit to the average profit of the year (per product and geographic region, see the blue bars in Fig. 9).

T8 Compare each product's profit to the average profit of the corresponding product category (per quarter, year, and geographic region, see the top row in Fig. 9).

The subsequent sections describe a formal model that is expressive enough to specify relationships between cells in a small-multiple display in order to answer these and similar tasks. Before formally describing this model, Sec. 3.2 summarizes the table algebra [27] as the formalism on which our model is based. Section 3.3 then discusses basic considerations with respect to our model. Sections 3.4 to 3.6 cover three general types of comparison we have identified: *absolute references* relate cells to fixed other cells within the matrix (e.g., tasks T1 to T4, see Sec. 3.4); *relative references* are based on adjacencies between categories with respect to some order (e.g., tasks T5 and T6, see Sec. 3.5); and *comparisons between hierarchy levels* relate the data at different levels of aggregation (e.g., tasks T7 and T8, see Sec. 3.6).

3.2 Table Algebra

Our reference specification in sections 3.4 and 3.5 is based on a matrix configuration T resulting from a *cross operation* ($\times$) in the table algebra by Stolte et al. [27]. The operation performs the Cartesian product of the categories of n dimensions $X_1,\dots,X_n$, which are assigned to the x and y axis of the matrix and partition it into columns and rows:

$$T = X_1 \times \cdots \times X_n = \{(t_1,\dots,t_n) \mid t_i \in X_i\}\,, \qquad (1)$$

where $X_i = \{x_{i,1}, x_{i,2},\dots, x_{i,l_i}\}$ denotes a dimension with l_i categories. The hierarchy of categories on the x axis in Fig. 1 is such an example that results from a cross operation. The cells of the matrix represent the leaf nodes in the combined hierarchy of dimensions. Each cell is formally described by an n-tuple $t = (t_1,\dots,t_n)$ which represents a unique combination of categories of the partitioning dimensions (e.g., see the annotated cell in Fig. 1). In our formalism, we use these n-tuples to unambiguously refer to the cells/graphics of the matrix.

3.3 Structure-based Comparison

The goal of our model is to facilitate the definition of meaningful references between the categories for comparison based on the hierarchical structure of the matrix. A key idea of our model is the discrimination between three roles of dimensions. The role of a dimension depends on the overall structure of the small-multiple display and the intent of the comparison, and it may change in the course of the analysis:

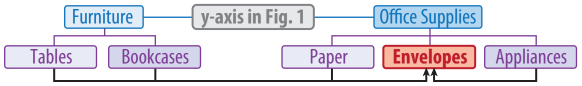


Fig. 3. Products are grouped by the nesting dimension "product category" and are compared to envelopes. In this hierarchy, however, furniture cannot be compared to office supplies because the structure and categories of the subtrees do not match.

Comparing dimensions relate different categories of the same dimension to each other, that is they specify what should be compared to what (e.g., the years in Fig. 2a or quarters in Fig. 2b). Sec. 3.6 introduces intermediate aggregates as a concept that extends comparing dimensions to also enable comparisons between hierarchy levels (see sample tasks T7 and T8).

Refining dimensions relate equivalent categories of the same dimension to each other, i.e., they subdivide the data for increasing the level of detail of the comparison (e.g., the quarters in Fig. 2a).

Nesting dimensions are different from comparing and refining dimensions. They do not introduce an additional refinement of the data, but rather provide a disjunct grouping of the subcategories which are *nested within* their parent categories (compare to the nest operation in the table algebra [27]). Such an example are the dimensions partitioning the y axis in Fig. 1. Since the categories and/or structure of the subtrees are different, a nesting dimension cannot be a comparing dimension in our model (see Fig. 3).

Our model for structure-based comparison is based on the following elementary guidelines. We have derived these guidelines in a bottom-up manner from a larger list of tasks (including T1 to T8) in different application fields (see Sec. 4).

G1 Scalability. The model should not impose a limit on the number of involved dimensions or categories.

G2 Generality. The model should allow many types of visualization per cell (e.g., bar chart, scatterplot, or geographic map).

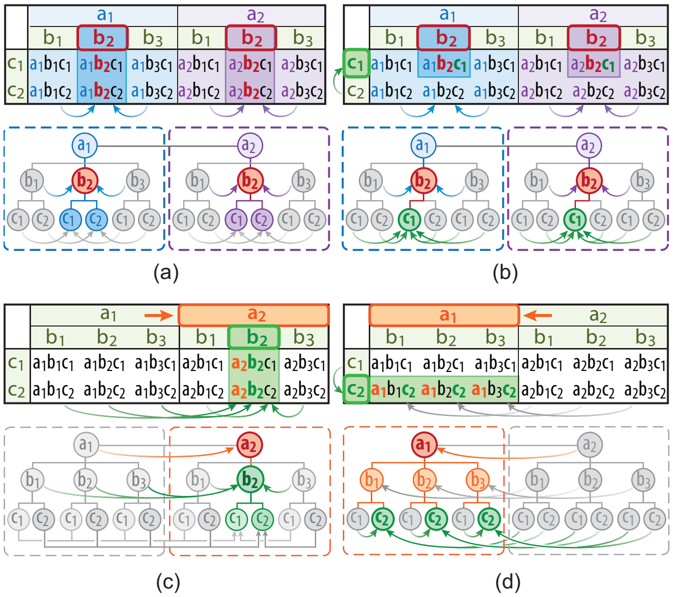
G3 Surjectivity. Each cell of the matrix can be compared to only one reference, but a cell can serve as reference for multiple cells. This restriction to one-to-many relations keeps the model simple enough to be comprehensible for a user, and it facilitates the design of visual comparisons where only two visualizations need to be overlaid per cell (see Fig. 2, for example).

G4 Hierarchical consistency. At each level of the hierarchy, the reference graph should be consistent for all sub-categories (e.g., see the reference graphs in Figs. 2 and 4). This makes it easier for the user to understand which cells are related to each other and enables a simple specification of the reference graph.

G5 Order independence. The possibilities for specifying a reference should be independent of the order of the dimensions within the hierarchy in general, and they should be independent of the assignment to the x and y axis of the matrix in particular. Changes in the configuration of the matrix (e.g., swapping two dimensions) should thus not restrict the model. An exception to this guideline are comparisons between hierarchy levels in Sec. 3.6.

G6 Compactness. The references between the categories of a comparing dimension should be compact, i.e., they should be described by at most one target category per comparing dimension and—in case of ordinal data—the order of the categories. This facilitates the specification of a reference graph for the user.

We consider especially guidelines G3 and G4 as a useful compromise between being able to cover every possible reference graph between cells in a matrix and keeping the model simple and usable. The subsequent sections describe a formalism for defining structure-based comparisons which adhere to these guidelines.


a1
a2
b1
b2
b3
c1
c2
a1b1c1 a1b2c1 a1b3c1 a2b1c1 a2b2c1 a2b3c1
a1b1c2 a1b2c2 a1b3c2 a2b1c2 a2b2c2 a2b3c2

(a)

(b)

(c)

(d)

Fig. 4. Absolute reference specification for visual comparison. The matrix/hierarchy results from a cross operation of the partitioning dimensions $A = \{a_1, a_2\}$ and $B = \{b_1, b_2, b_3\}$ on the x axis as well as $C = \{c_1, c_2\}$ on the y axis. (a) Category b_2 within the comparing dimension B is defined as reference. (b) The reference definition is further refined on the y axis. (c) A reference is specified within both dimensions on the x axis. (d) Categories on both x and y axis define a reference.

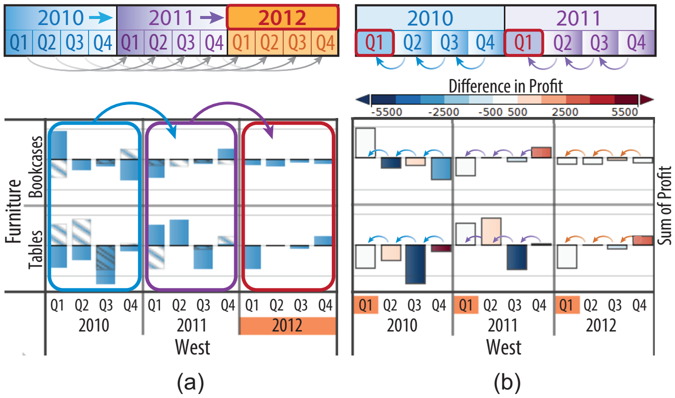
3.4 Absolute Reference Specification

With an absolute reference specification, cells can be compared to a fixed reference at different levels of the hierarchy as illustrated in Fig. 2. For a matrix configuration T that is defined by n partitioning dimensions $X_1, \dots, X_n$, such a reference can be expressed by an n-tuple $r = (r_1, \dots, r_n)$. Here, r_i can either be undefined (denoted as $\varnothing$) or one category of a dimension $X_i \in T$ which then becomes a comparing dimension $\tilde{X}_i$. In Fig. 2b, for example, the third quarter (Q3) has been defined in the reference tuple $r = (\varnothing, \varnothing, \text{Q3}, \varnothing, \varnothing)$, and the corresponding dimension "ship date [quarter]" thus becomes a comparing dimension. For each matrix cell described by an n-tuple t (Eq. 1), we can obtain the n-tuple describing the related *absolute reference* cell as

$$R_{abs}(t,r) = (x_1, \dots, x_n) \quad \text{with} \quad x_i = \begin{cases} t_i & \text{if } r_i = \varnothing \\ r_i & \text{if } r_i \in \tilde{X}_i \end{cases}$$

Our definition specifies exactly one reference per cell. The reference is thereby identical to t in the dimensions where r is undefined (refining and nesting dimensions) and equals the defined category for comparing dimensions (e.g., Q3 in Fig. 2b). Accordingly, the cells that are identical to r in all the comparing dimensions are references.

Figure 4 illustrates the reference graph between cells using an absolute reference specification. The matrix is partitioned by dimensions A, B and C, and the cells represent the leaf nodes in the combined hierarchy of categories. Colored arrows illustrate when categories are related to a reference category within a comparing dimension. Gray arrows illustrate when categories refer to themselves, but within another subtree (refining dimension). In Fig. 4a, the dimension B on the x axis of the matrix is a comparing dimension, i.e., $r = (\varnothing, b_2, \varnothing)$. Such a relation, for example, would compare the quarters within each year to a specific reference quarter (see Fig. 2b and task T4). We define $B(t)$ to be the category of the dimension B for a cell described by t. The reference cells in Fig. 4a are highlighted and satisfy the predicate $B(t) = b_2$, i.e., all cells where the category of dimension B is b_2. Since the reference tuple is undefined for A (refining dimension), the cells with a blue background are compared to a different column than those shown with a purple background (see colored arrows). In Fig. 4b, the reference definition is extended on the y axis, i.e., $r = (\varnothing, b_2, c_1)$. All cells are thus compared to the reference with $B(t) = b_2$ and $C(t) = c_1$.



(a)

(b)

Fig. 5. Comparison to a relative reference. (a) Each year is compared to the next year on a quarterly basis (see sample task T5 in Sec. 3.1). (b) Each quarter within a year is compared to the previous one (see sample task T6). Differences are encoded in color.

In Fig. 4c, both dimensions on the x axis are comparing dimensions, i.e., $r = (a_2, b_2, \varnothing)$. The cells in each row of the matrix are then compared to references, where $A(t) = a_2$, $B(t) = b_2$, and $C(t)$ remains unchanged. Such a comparison, for example, could compare each month to a specific month of a specific year. If all categories in r are defined, all cells would be compared to a single reference cell in the matrix (T1, not illustrated here). In Fig. 4d, the reference is $(a_1, \varnothing, c_2)$. The cells are thus compared to references, where $A(t) = a_1$, $C(t) = c_2$, and $B(t)$ remains the same. For example, each quarter of a year could be compared to the same quarter in a reference year (see Fig. 2a and T3).

An absolute reference specification is independent for each dimension, i.e., the reference categories defined at different levels of the hierarchy do not affect each other (compare to guideline G5 in Sec. 3.3). They only influence comparing dimensions where all other categories are compared to the specified reference. Changing the order of the dimensions on an axis only affects the location of the graphics, but not the comparison.

While the focus of this paper is not on interaction, the compact representation of the reference tuple (see guideline G6) suggests concepts for an interactive specification of structure-based comparisons. The user can click on the category labels on the x and y axis of the matrix to toggle the state of the category in the reference tuple, possibly replacing a previous reference category for the same dimension.

3.5 Relative Reference Specification

With a relative reference specification, cells are compared to a moving reference instead of a fixed one. The comparing dimension can be defined at different levels of the hierarchy as illustrated in Fig. 5. Each category is then compared to the preceding or following category within the comparing dimension. Such a comparison implies that the categories involved in defining a relative reference have a *logical ordering*. This ordering is inherently given for ordinal dimensions or can be defined by ranking categories according to a user-specified measure, for example. We first describe the simple case, where the reference is defined at only one hierarchy level. A reference definition for composite categories is then discussed in Sec. 3.5.2.

3.5.1 Reference specification on one hierarchy level

In principle, there are three possible cases for a relative comparison of cells. Defining a global direction of comparison, categories can either be compared to the 1) previous or 2) subsequent category within a comparing dimension as shown in Fig. 5. As a third alternative, cells can be compared based on a specified reference category. This provides additional flexibility by intermixing the comparison direction within a subtree as illustrated in Fig. 6. Additionally, the latter case can express the first two cases as well and can be defined analogous to an absolute reference definition. Consequently, we define a relative comparison by an n-tuple $r = (r_1, \dots, r_n)$, where r_i can either be undefined

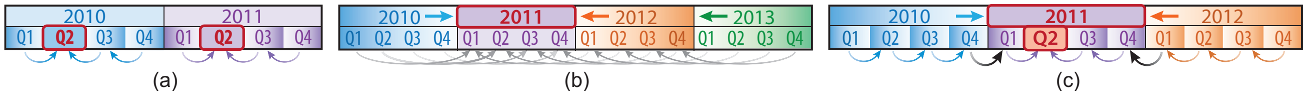


Fig. 6. Depending on the location of the defined reference category, categories are compared relatively to the previous or next category. (a) Relative comparison within each subtree. (b) Relative comparison of entire subtrees. (c) Relative comparison for composite categories.

for any refining or nesting dimension, or it can represent a category of an ordinal dimension $\hat{X}_i \in T$ which then becomes a comparing dimension. For each matrix cell described by an n-tuple t, we can obtain the *relative reference* as $R_{rel}(t,r) = (x_1, \ldots, x_n)$ with

$$x_i = \begin{cases} \text{decrement}(t_i) & \text{if } (r_i \in \hat{X}_i) \wedge (r_i < t_i) \\ \text{increment}(t_i) & \text{if } (r_i \in \hat{X}_i) \wedge (r_i > t_i) \\ t_i & \text{otherwise.} \end{cases} \tag{2}$$

In case r_i is defined, both r_i and t_i stem from the same ordinal dimension $\hat{X}_i$ and can thus be compared. If r_i is less than t_i, we decrement t_i to the previous element in $\hat{X}_i$. If r_i is greater than t_i, we increment t_i.

In Fig. 6a, for example, $r = (\varnothing, \text{Q2})$ and the ordinal dimension "quarter" is thus a comparing dimension (compare to Fig. 5b and task T6). Cells with a quarter before Q2 are compared to the subsequent cell each, and cells after Q2 to the preceding one (see arrows). With such a comparison, for example, one can check time-dependent data for continuity around a peak. In Fig. 6b, the reference tuple is $(2011, \varnothing)$ and the years are either incremented or decremented as illustrated with arrows (compare also to Fig. 5a and task T5). Since the tuple is undefined for the refining dimension "quarter," each cell is compared to a reference where the quarters remain unchanged.

3.5.2 Reference specification for composite categories

In some cases, inferring an order for a relative reference specification involves the composition of multiple dimensions. For a meaningful comparison, however, the comparing dimensions need to have a semantical ordering, i.e., the categories of one dimension must be semantically *embedded* within the categories of another one. A typical example is time where hours are embedded within days, days within months, months within years, etc. It should be noted that this ordering is independent of the ordering of the dimensions in the matrix/hierarchy (compare to guideline G5 in Sec. 3.3).

The main idea of a reference specification for composite categories is to define the references between the categories of the comparing dimension with the lowest rank (e.g., quarters that are embedded within years). The comparison direction for each cell is then determined in a top-down manner as illustrated in Fig. 6c. For example, this enables to compare the last quarter of a year to the first quarter of the next year. Specifically, we adapt Eq. 2 to work on the tuples r and t as a whole instead of considering each dimension independently. Note that only the categories of ordinal comparing dimensions $\hat{X}_i$ are compared and possibly modified—the categories of the other dimensions remain unchanged:

$$R_{rel}(t,r) = \begin{cases} \text{decrement}(t) & \text{if } r < t \\ \text{increment}(t) & \text{if } r > t \\ t & \text{otherwise.} \end{cases} \tag{3}$$

We first compare the categories of the comparing dimension with the highest rank (e.g., years in Fig. 6c). In case the corresponding categories of r and t are equal, the categories of the next-ranked comparing dimension are compared, and so on. When t is incremented or decremented as a result, we start with the category of the comparing dimension with the lowest rank (quarters in Fig. 6c). In case we are at the boundary between two subtrees (first or last element in the embedded dimension), we jump to the neighboring category in the subtree of the decremented or incremented category of the next-ranked comparing dimension (illustrated by black arrows in Fig. 6c). For example, the last quarter of the year 2010 refers to the first quarter of 2011.

Figure 7 illustrates a setup with three partitioning dimensions on the x axis. In Fig. 7a, the middle and inner dimension are comparing dimensions, i.e., $r = (\varnothing, b_1, c_2)$. Dimension B is highest ranked, and cells are principally compared based on the corresponding categories. Accordingly, all cells with category $B(t) = b_2$ are compared to the preceding composite category of dimensions B and C. For cells with $B(t) = b_1$ also the categories of the next-ranked dimension C are considered. Fig. 7b shows a similar comparison, however, the comparing dimensions are not adjacent, i.e., $r = (a_1, \varnothing, c_2)$. The categories of dimensions A and C form composite categories, which are intersected by the categories of dimension B which acts as a refining dimension. Cells with category a_2 are then compared to the previous composite category, where the categories of B remain unchanged. At the boundary between the subtrees, the cell $a_2b_1c_1$ is compared to $a_1b_1c_3$, and $a_2b_2c_1$ is compared to $a_1b_2c_3$ (indicated with black arrows). In Fig. 7c, the categories of the comparing dimension A and B form a combined hierarchy with $r = (a_1, b_2, \varnothing)$, and the categories of the refining dimension C remain unchanged (see gray arrows).

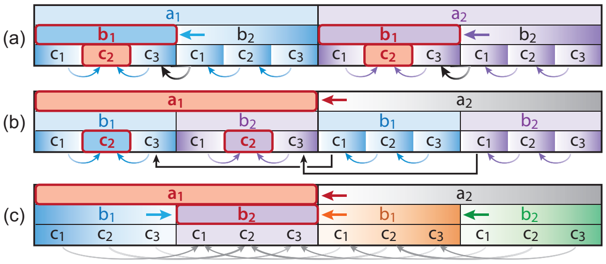


Fig. 7. Various configurations of a relative comparison for composite categories within a hierarchy.

3.6 Comparison between Hierarchy Levels

Many tasks require comparing subcategories to the parent category. An example would be comparing the average profit per quarter to the average profit of the entire year (see sample task T7 in Sec. 3.1). In general, comparisons between hierarchy levels require the aggregation of an intermediate node of the hierarchy. We thus refer to such aggregates as *intermediate aggregates* (IAs), which are represented as additional nodes in the hierarchy (see Fig. 8). By preserving the structure of the subtrees when computing IAs, we can apply our model to references between hierarchy levels in a natural way. Conceptually, the IAs can be considered as additional nodes of the hierarchy. They can thus be referred to by a reference tuple like any other node even though the subset of the data represented by IAs is by definition not disjunctive to the other nodes of the respective level of the subtree representing actual categories.

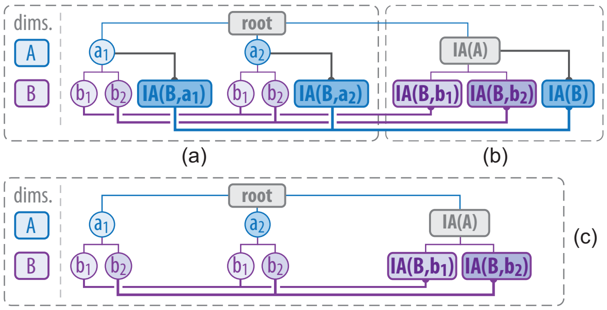


Fig. 8. Calculating intermediate aggregates (IAs) within the hierarchy. (a) IAs are computed for dimension B with respect to each category of dimension A. (b) Each category of dimension B is summarized by an IA (purple). Additionally, $IA(B)$ represents the total summary. (c) Only the categories of dimension B are summarized.

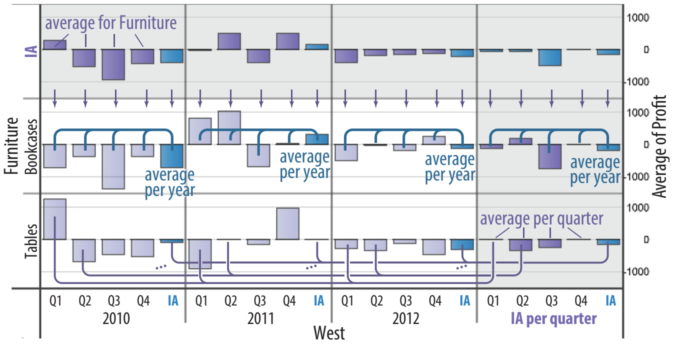


Fig. 9. Example with intermediate aggregates. For each year, IAs summarize the data of the corresponding quarters (i.e., average per year). Additional IAs summarize the data for each row and column.

We denote such intermediate aggregates as $IA(X_i, x_j)$, where X_i represents the dimension in the hierarchy that is aggregated, and x_j represents an optional filter category from a dimension X_j. In Fig. 8a, for example, IAs are computed for the inner dimension B with respect to the parent categories. The resulting node $IA(B, a_1)$ summarizes the data of the subtree of a_1, and $IA(B, a_2)$ summarizes the subtree of a_2 (illustrated with black lines above the nodes). In Fig. 8b, the node $IA(A)$ is added to the hierarchy, which summarizes the data of the entire tree. In order to comply with our model for structure-based comparison, additional IAs are added for dimension B such that each subtree has the same number of nodes. In this context, $IA(B, b_1)$ summarizes the data for category b_1, and $IA(B, b_2)$ summarizes the data for b_2 (see the purple bold lines below the nodes). The node $IA(B)$ has the same value as $IA(A)$ (total summary) and aggregates the data of $IA(B, a_1)$ and $IA(B, a_2)$. In Fig. 8c, only the intermediate aggregates for each category in B are computed. In general, the decision to summarize the categories of a hierarchy level by adding an IA-node is independent from other hierarchy levels. That is, IAs can be computed for single levels, multiple levels, or all levels. Fig. 9 shows an example setup with intermediate aggregates.

4 Application Examples

In this section, we present examples for the versatile applicability of our model in different scenarios. The examples are based on real-world data and tasks in different application domains, and involve a visual comparison by overlay and explicit encoding (compare to Gleicher et al. [12]). These types of comparison benefit from our model, since they both require an explicit definition of references.

4.1 Parameter Space Analysis

The first application example addresses the analysis of the parameter space of a complex system. This example is based on real data and a real task in the automotive industry. The application background is the development process of car designs by means of 1D-CFD multi-run simulations as described in previous work [6, 21, 24]. Specifically, the data represents a study of a car engine where four parameters have been varied, which can be grouped into two types. *Operating parameters* describe conditions varying during the operation. In this example, this includes "Engine Speed" (abbreviated as speed) measured in rotations per minute (rpm) and the "Load Signal" (abbreviated as load) corresponding to the percentage at which the gas pedal is pressed. As all combinations of values may potentially occur during operation, it is essential for the analysis to overview the entire 2D space of these parameters. In contrast, *design parameters* represent choices of the engineer. In this example, this includes "Intake Valve Closing Shift" (abbr. IVCS) measuring a specific timing in milliseconds (ms) and different configurations of the engine as a categorical parameter. Continuous parameters have been discretized individually and each combination of values has been simulated (also known as full-factorial design). In this example, relevant simulation outputs comprise scalar values measuring torque in Newton meters (Nm) which is to be maximized and trapped fuel in kilograms (kg) which is to be minimized.

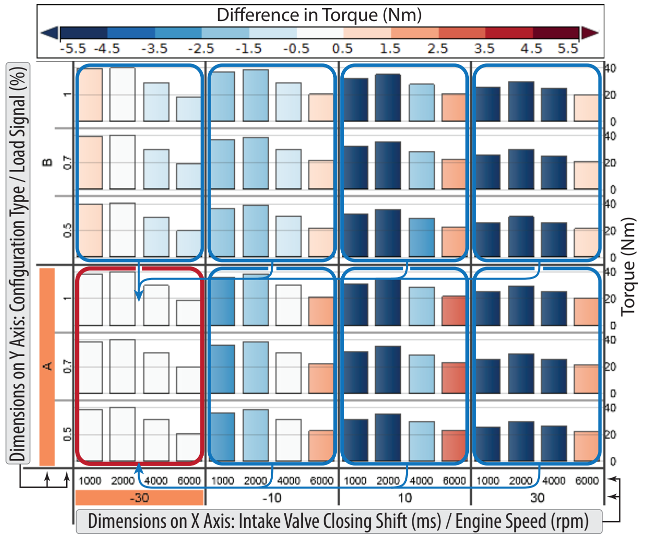


Fig. 10. Parameter space analysis of a car engine simulation. Differences of torque with respect to the combination of design parameters with maximal values (red rectangle) are encoded in color.

The first step of the analysis aims at identifying parameter combinations that maximize torque (compare to task T1 in Sec. 3.1). Fig. 10 shows a visual representation of the four-dimensional parameter space. The design parameters define the outer subdivision of the layout, while the operating parameters define the inner subdivision, enabling a coherent representation of their 2D domain. Visualizing torque using bar charts shows maximal values for configuration A and IVCS = -30 with speed set to 2000. However, the domain of design parameters has to be considered in its entirety as reasoned above. Our model thus helps to relate all combinations of design parameters to all other points of the design space, i.e., $r = (\text{-}30, \varnothing, \text{A}, \varnothing)$. While the bars represent the actual values as context, color explicitly encodes the differences in torque to the reference. We use a diverging color map to distinguish positive and negative differences [8]. This visualization shows that for IVCS = -30, torque is nearly identical for configuration B for speed = 2000 while larger speeds generate less torque in configuration B. For speed = 1000, however, configuration B is superior. The engineer thus has to decide whether to optimize the engine for low or high values of speed. Other values of IVCS generate considerably less torque for speed $\leq$ 4000, but all of them exceed IVCS = -30 at speed = 6000.

For a more detailed investigation of configuration A at IVCS = -30, the second step concerns an analysis of both torque and trapped fuel using scatterplots (see Fig. 11). The view layout is similar as before, but different values of speed are now represented by color. This increases the spatial resolution of the x axis representing trapped fuel. Defining references in the same way as in the first step supports a comparison by overlay which directly visualizes the trade-off between the two objectives. The current data are depicted as filled dots and the reference data as circles which are connected by a line (compare to Turkay et al. [31] and Robertson et al. [26]). This shows similar yet not identical gradients of the trade-off across the parameter space. Comparing the configurations A and B at IVCS = -30, the lower values of torque for configuration B at high speeds are not equally compensated by reductions of trapped fuel.

The third step analyzes the sensitivity of torque regarding variations of IVCS (see Fig. 12). Based on the same layout as Fig. 10, our model is now used to specify relative references with $r = (\text{-}30, \varnothing, \varnothing, \varnothing)$. Each value of IVCS is thus related to the respectively smaller one (compare to task T5 in Sec. 3.1). Directly representing the difference in torque by the height of the bars mimics the first derivative of torque with respect to IVCS. This shows gradients more clearly than Fig. 10 while color is used to represent the absolute values in this case. The visualization directly shows that torque is decreasing sharply for increasing IVCS for speed = 1000. For speed = 6000, however, torque is increasing before also starting to decrease for IVCS = 30.

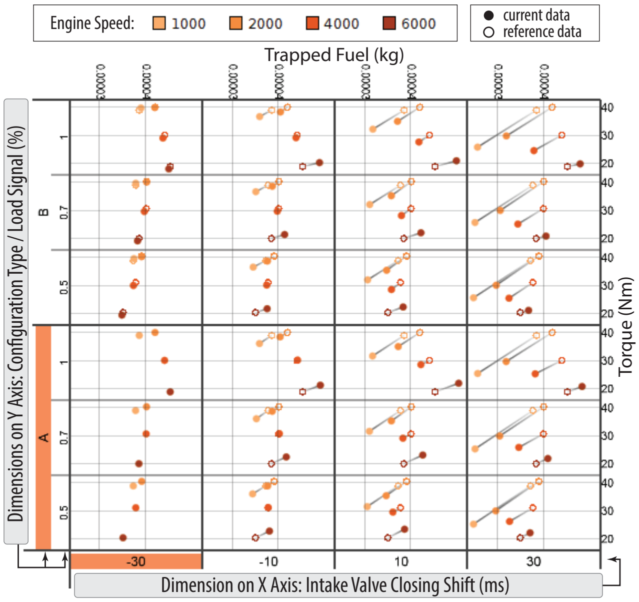


Fig. 11. Comparison by overlay to the same reference as in Fig. 10.

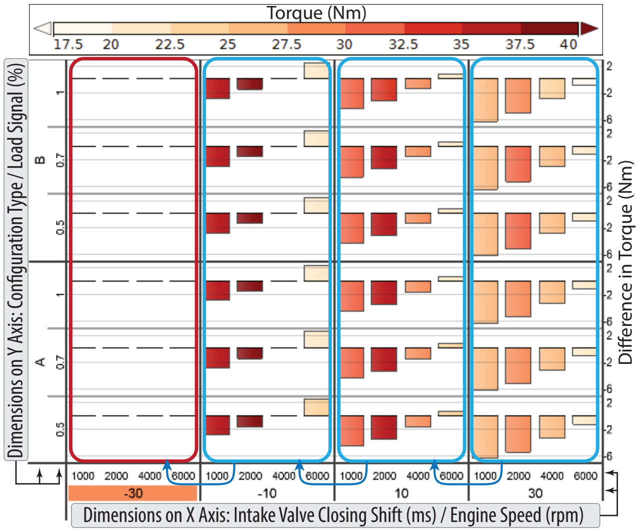


Fig. 12. Relative comparison to study the sensitivity with respect to variations in "Intake Valve Closing Shift." Color encodes torque and the bar height represents the respective difference in torque.

Concluding, this example shows an application of small multiple displays to support the analysis of a multi-dimensional parameter space using dimension composition. In this context, our model is generally useful to relate arbitrary subspaces of the parameter space as well as to define derivatives using relative references. As another option, comparisons between hierarchy levels could be helpful to emphasize the local effect of single parameters in the context of the others by subtracting the average of the intermediate aggregate.

4.2 Analysis of Natural Gas Consumption

This example deals with the effect of temporal categories such as months, hours, and days of the week on the average consumption of natural gas of a large European city. The data comprises hourly measurements for approximately five years (42,869 data samples). The background of this data is the statistical modeling of natural gas consumption as a regression model for prediction and sensitivity analysis [22]. In this context, the structural effects of time are important for feature selection and feature transformation (e.g., for identifying good splits of regression trees). While statistical modeling is beyond the scope of this paper, we now analyze variations in the consumption of natural gas with respect to hour of the day and day of the week.

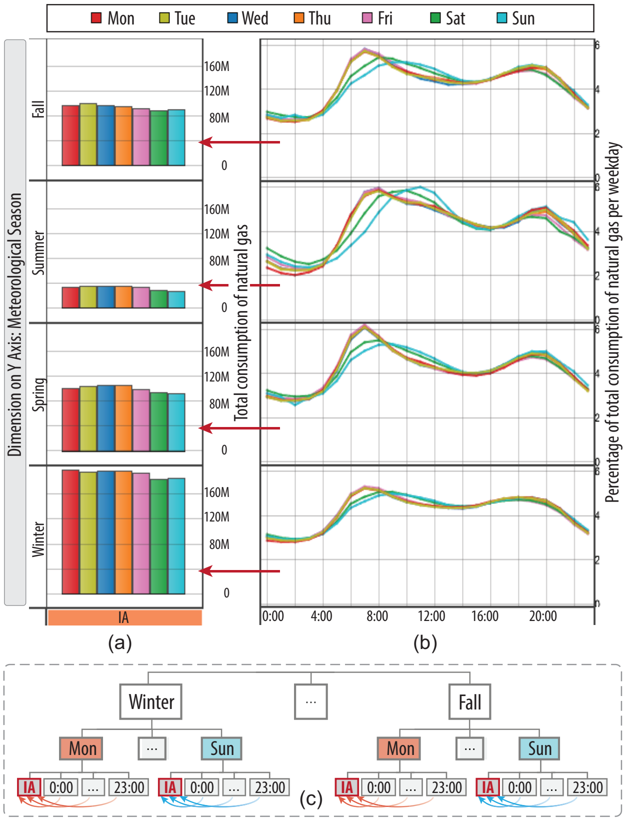


Fig. 13. Analysis of the natural gas consumption in a large city. (a) IAs show the total consumption per weekday (color). (b) Percentage of total consumption per weekday over hour of day. (c) Hierarchy of categories with "hour of day" as a comparing dimension.

The first goal is to identify peaks of consumption throughout the day and to compare them by their occurrence in time and distinctness for different weekdays. In order to account for seasonal effects, the matrix in Fig. 13 is vertically partitioned by meteorological seasons. Fig. 13a shows the total consumption per weekday (color), which is computed as intermediate aggregates (IAs), and Fig. 13b shows the corresponding relative consumption over the course of the day (compare to sample task T7 in Sec. 3.1). The relative consumption is shown per hour in percentage of the total consumption per weekday, e.g., $100 \cdot \frac{\text{consumption[Fall, Sat, 12:00]}}{\text{total consumption[Fall, Sat]}}$. As illustrated in Fig.13c, the reference for comparison is $(\varnothing, \varnothing, \text{IA})$, where "hour of day" is a comparing dimension. We can see that workdays have a higher total consumption than weekends (Fig. 13a) and the morning peak in consumption is later for weekends (Fig. 13b). With respect to seasonal differences, for example, Sundays have a peak around 11:00 in summer and around 9:00 in winter. In summer, moreover, the morning peak is similarly high for workdays and weekends, but lower for weekends in the other seasons.

The second goal is to quantify the hourly effect of different weekdays on the consumption of natural gas. This task is related to estimating the error a regression model could have by not distinguishing between weekdays. Fig. 14a shows the average consumption per hour as an intermediate aggregate for each season (compare to sample task T8 in Sec. 3.1). The deviation from this graph per hour and weekday is shown in Fig. 14b, e.g., average consumption[Fall, Sat, 12:00] − average consumption[Fall, 12:00]. As illustrated in Fig. 14c, "weekday" is a comparing dimension where $r = (\varnothing, \text{IA}, \varnothing)$. Especially in the morning (around 7:00), we can see large differences in consumption between workdays and weekends. Compared to other seasons, daily variations are relatively small in summer. Moreover, Friday evenings have a similar consumption like weekends (indicated by ellipses), and Sunday evenings (after 22:00) are comparable to workdays.

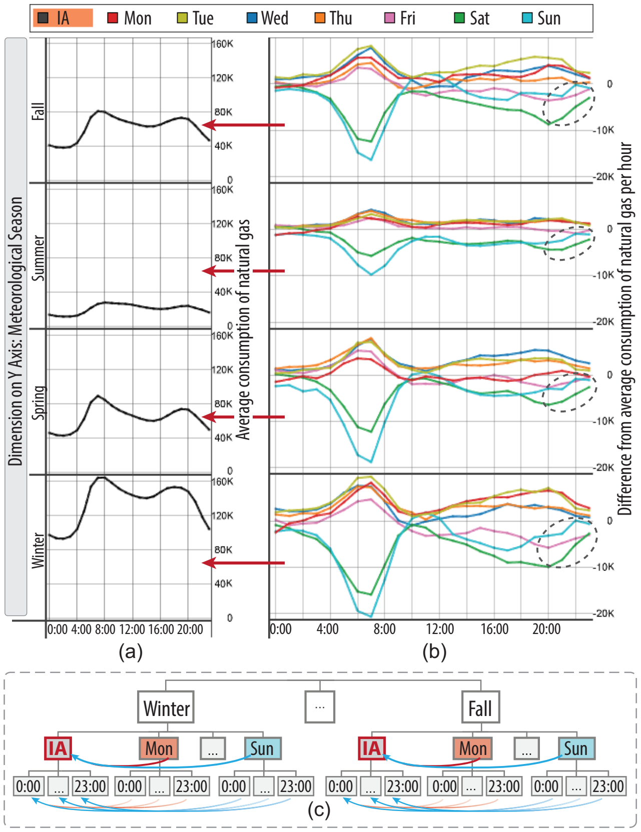


Fig. 14. (a) Intermediate aggregates show the average consumption of natural gas per season over the course of the day. (b) Difference from the average consumption per hour. (c) Hierarchy of categories where "weekday" is a comparing dimension.

In this example, our model is useful to study percentual measures and to relate data across different levels of the hierarchy (including intermediate aggregates). This enables the analyst to flexibly define comparisons to investigate variations in the consumption of natural gas with respect to different seasons, weekdays, and hours of the day.

4.3 Differences in Movement Data

As an example in the context of scientific data, this section describes an analysis of records of vessel movements around the coast of Norway. The data stems from an Automatic Identification System (AIS) which is used by the Norwegian Coastal Administration (NCA) to track and monitor vessels. The visualizations are based on *Kernel Density Estimates* (KDE) and are generated in a system as described by Daae Lampe et al. [11, 10]. Here, individual movement trajectories are convoluted with a line kernel instead of a point spread function.

Our first step of the analysis is to identify peaks and study changes in traffic with respect to different time periods of the day (compare to sample task T7 in Sec. 3.1). The average movements are computed as an intermediate aggregate and selected as an absolute reference (Fig. 15 to the right). The other graphics show the per-pixel differences to the average for different time spans (compare to difference views [11]). We can see that most traffic is between 6:00 and 18:00 (dark red color). There is generally less traffic compared to the average for certain routes between 18:00 and 6:00 (blue color). Between midnight and 6:00, moreover, there is more traffic between Skudeneshavn and Kvitsøy (red arrow) compared to the other time intervals.

For a more detailed analysis, our second step studies the effects of different weekdays on the traffic. In Fig. 16, the average traffic per time span is computed as an intermediate aggregate (top row), which is defined as an absolute reference. Each column then depicts the difference to the respective average for different weekdays using a diverging color map (compare to task T8 in Sec. 3.1). Between 0:00 on Saturday and 6:00 on Sunday, for example, there is less traffic between Stavanger and Tau (see purple arrows). On Sunday between 0:00 and 6:00 as well as 12:00 and 18:00, there are less ships leaving southwards from Stavanger (green arrows). We also see some routes with higher traffic (dark red colors), for example, on Thursday and Friday between 12:00 and 18:00.

In this example, we apply our model to study differences in traffic across different levels of aggregation. We can investigate traffic patterns with respect to different time periods of the day as well as days of the week.

5 Discussion and Future Work

A key advantage of our model is that it enables a multitude of useful comparisons of hierarchically structured categorical data within small-multiple displays. Using a generic formalism, categories can be compared to an absolute or relative reference, both within and across subtrees. In consistence with the guidelines in Sec. 3.3, the model scales for many dimensions and it does not impose any limit on the number of involved categories (G1). Moreover, our model is independent of the underlying type of visualization (G2) as well as the order of the dimensions (G5) in the small-multiple display. Being able to express a large set of tasks, we consider guidelines G3 and G4 as a useful compromise between (too much) flexibility and (too restrictive) simplification. As an important goal for practical reasons, the model enables a simple specification of complex reference graphs based on a single tuple (G6). In addition to business intelligence, Sec. 4 has shown the applicability to comparison tasks in very diverse application domains.

We demonstrated the applicability in a comparison by overlay, through an explicit encoding of differences, and through defining percentual measures. As discussed in the visualization literature [12, 29], these options successfully overcome shortcomings of comparison by juxtaposition. In particular, these options significantly increase the precision of comparisons especially for cells which are not vertically or horizontally aligned as well as for large matrices in general, where comparing two cells may even require scrolling. Besides visual comparison, a simple and comprehensive specification of percentual measures is a benefit by itself for many applications.

As a practical consideration, our model specifies a reference graph between cells regardless of the actual availability of data, i.e., also for non-existing combinations of categories. Applications of the model will typically need to deal with such cases. In our experience, appropriate strategies depend on the task, the visualization, and potentially

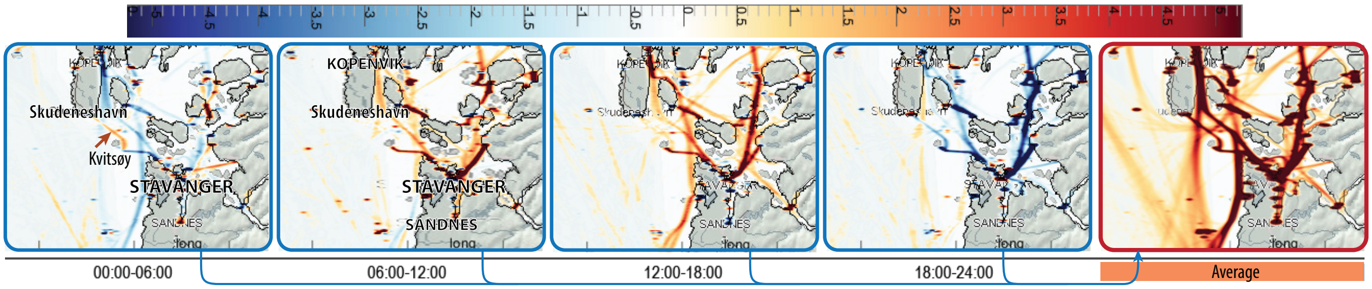


Fig. 15. Analysis of vessel movement data. Different time spans are compared to the average traffic (image courtesy of O. Daae Lampe).

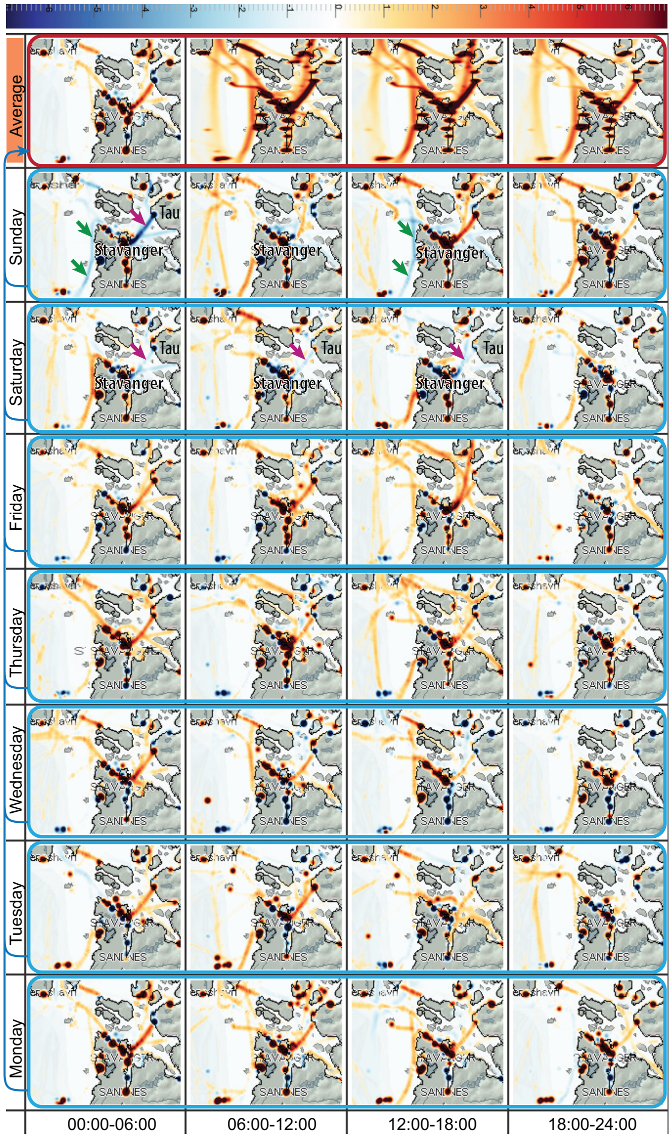


Fig. 16. Comparison of movement data for different time spans (horizontal) and weekdays (vertical). The top row shows the average traffic per time span as an intermediate aggregate which is defined as an absolute reference. The columns then show the difference to the respective average traffic for different weekdays (image courtesy of O. Daae Lampe).

involved types of aggregation. For example, assuming a value of zero might be possible for measures like the sum while visually indicating the absence of reference data might be necessary in other cases.

The focus of this paper is on the formal definition and application-oriented discussion of our model for structure-based comparison. However, implementations of the model will also need to deal with other aspects including an interactive specification and the selection of an appropriate visual encoding for a given task. Concerning the specification, the compact definition based on a single tuple (see guideline G6) makes using standard controls of graphical user interfaces straightforward. An alternative—and presumably more intuitive—option involves marking categories or combinations of categories directly within the small-multiple display (e.g., by clicking on category labels or cells). Therefore, an evaluation of different possibilities to specify the reference tuple provides an important aspect for future work. Additionally, we want to enable a task-based specification of a reference such as comparison to the minimum or maximum value within each subtree. Other issues for future work include a study of different approaches for visually indicating the references between cells in a general and scalable way. We also want to design and evaluate further visualizations for comparison by overlay such as function graphs.

6 Conclusions

This paper proposes a formal model for structure-based comparison of cells within a hierarchically organized small-multiple display. The model enables the definition of a multitude of practically relevant types of reference graphs while being based on a simple and compact formal specification. Categories can be compared absolutely to a fixed reference or relatively with respect to a semantical order of the categories. References can be defined within and across subtrees and also support comparisons between hierarchy levels based on intermediate aggregates. We have demonstrated the general applicability of our model in the context of business data as well as for analyzing simulation data, time series data, and movement data, involving different types of visualizations and comparisons. We therefore believe that our model will be of high practical relevance to support a comparative visual analysis in many applications domains.

Acknowledgments

The authors thank A. Pobitzer for his help with respect to the formal description of our model. We are indebted to O. Daae Lampe (Christian Michelsen Research, Norway) for providing the visualizations of the movement data. The Superstore Sales data is courtesy of Tableau Software, Inc., the multi-run simulation data is courtesy of AVL List GmbH, the gas consumption data is courtesy of HAKOM Solutions GmbH & Co KG, and the AIS data is courtesy of the Norwegian Coastal Administration. This work has been supported by the Austrian Funding Agency (FFG) within the scope of the COMET K1 program as well as the Austrian Science Fund (FWF) through the ViMaL project (No. P21695).

References

[1] Tableau Software. www.tableausoftware.com, accessed March 2013.

[2] Tableau Software. Tableau Desktop Help, 2012.

[3] G. Alvarez and P. Cavanagh. The capacity of visual short-term memory is set both by visual information load and by number of objects. *Psychol. Sci.*, 15(2):106–111, 2004.

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

[5] R. A. Becker, W. S. Cleveland, and M.-J. Shyu. The visual design and control of trellis display. *J. Computational and Graphical Statistics*, 5(2):123–155, 1996.

[6] W. Berger, H. Piringer, P. Filzmoser, and M. E. Gröller. Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction. *Computer Graphics Forum*, 30(3):911–920, 2011.

[7] S. Bremm, T. von Landesberger, M. Heß, T. Schreck, P. Weil, and K. Hamacherk. Interactive visual comparison of multiple trees. In *Proc. IEEE Conf. Visual Analytics Science and Technology (VAST)*, pages 31–40, 2011.

[8] C. Brewer. Color use guidelines for data representation. In *Proc. Section on Statistical Graphics*, pages 55–60, 1999.

[9] C. Collins and S. Carpendale. VisLink: revealing relationships amongst visualizations. *IEEE Trans. Visualization and Computer Graphics*, 13(6):1192–1199, 2007.

[10] O. Daae Lampe and H. Hauser. Interactive visualization of streaming data with kernel density estimation. In *Proc. IEEE Pacific Visualization Symp.*, pages 171–178, 2011.

[11] O. Daae Lampe, J. Kehrer, and H. Hauser. Visual analysis of multivariate movement data using interactive difference views. In *Proc. Vision, Modeling, and Visualization (VMV)*, pages 315–322, 2010.

[12] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. Hansen, and J. Roberts. Visual comparison for information visualization. *Information Visualization*, 10(4):289–309, 2011.

[13] M. Graham and J. Kennedy. Combining linking and focusing techniques for a multiple hierarchy visualisation. In *Proc. IEEE Symp. Information Visualization*, pages 425–432, 2001.

[14] M. Graham and J. Kennedy. A survey of multiple tree visualisation. *Information Visualization*, 9(4):235–252, 2010.

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

[16] D. Holten and J. van Wijk. Visual comparison of hierarchically organized data. *Computer Graphics Forum*, 27(3):759–766, 2008.

[17] J. Kehrer and H. Hauser. Visualization and visual analysis of multifaceted scientific data: A survey. *IEEE Trans. Visualization and Computer Graphics*, 19(3):495–513, 2013.

[18] A. Lex, M. Streit, C. Partl, K. Kashofer, and D. Schmalstieg. Comparative analysis of multidimensional, quantitative data. *IEEE Trans. Visualization and Computer Graphics*, 16(6):1027–1035, 2010.

[19] J. Mackinlay, P. Hanrahan, and C. Stolte. Show me: Automatic presentation for visual analysis. *IEEE Trans. Visualization and Computer Graphics*, 13(6):1137–1144, 2007.

[20] M. Malik, C. Heinzl, and M. E. Gröller. Comparative visualization for parameter studies of dataset series. *IEEE Trans. Visualization and Computer Graphics*, 16(5):829–840, 2010.

[21] K. Matković, M. Jelović, J. Jurić, Z. Konyha, and D. Gračanin. Interactive visual analysis and exploration of injection systems simulations. In *Proc. IEEE Conf. Visualization*, pages 391–398, 2005.

[22] T. Mühlbacher and H. Piringer. A partition-based framework for building and validating regression models. *IEEE Trans. Visualization and Computer Graphics (accepted for IEEE VAST)*, 2013.

[23] T. Munzner, F. Guimbretière, S. Tasiran, L. Zhang, and Y. Zhou. TreeJuxtaposer: Scalable tree comparison using focus+context with guaranteed visibility. *ACM Trans. Graph.*, 22(3):453–462, 2003.

[24] H. Piringer, S. Pajer, W. Berger, and H. Teichmann. Comparative visual analysis of 2D function ensembles. *Computer Graphics Forum*, 31(3):1195–1204, 2012.

[25] J. C. Roberts. State of the art: Coordinated & multiple views in exploratory visualization. In *Proc. Coordinated & Multiple Views in Exploratory Visualization (CMV)*, pages 61–71, 2007.

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

[27] C. Stolte, D. Tang, and P. Hanrahan. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Trans. Visualization and Computer Graphics*, 8(1):52–65, 2002.

[28] E. Thomsen. *OLAP Solutions: Building Multidimensional Information Systems*. Wiley, second edition, 2002.

[29] C. Tominski, C. Forsell, and J. Johansson. Interaction support for visual comparison inspired by natural behavior. *IEEE Trans. Visualization and Computer Graphics*, 18(12):2719–2728, 2012.

[30] E. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 1983.

[31] C. Turkay, P. Filzmoser, and H. Hauser. Brushing dimensions – a dual visual analysis model for high-dimensional data. *IEEE Trans. Visualization and Computer Graphics*, 17(12):2591–2599, 2011.

[32] V. Verma and A. Pang. Comparative flow visualization. *IEEE Trans. Visualization and Computer Graphics*, 10(6):609–624, 2004.

[33] L. Wilkinson. *The Grammar of Graphics*. Springer, 2005.