# Flexible Visual Summaries of High-Dimensional Data using Hierarchical Aggregated Data-Subsets

??
*Supervised by: ??*

??
??

## Abstract

The number of installed sensors to acquire data, for example electricity meters in smart grids, is increasing rapidly. This huge amount of collected data needs to be analyzed and monitored by transmission system operators. This task is supported by visual analytics techniques, but traditional multi-dimensional data visualization techniques do not scale very well for high-dimensional data. The main contribution of this paper is a framework to efficiently inspect and compare such high-dimensional data. The key idea is to partition the data by the semantics of the underlying data dimensions into groups. Domain experts are familiar with the meta-information of the data and are able to structure these groups into a hierarchy. The proposed system visualizes the subsets of the data by appropriate means. These visual summaries can then be used to support the explorative overview tasks of the user.

**Keywords:** Visual Analytics, Hierarchical Aggregation, High-Dimension Data

## 1 Introduction

Creating an overview of the data is the first task a user desires of an information visualization [19]. However, traditional multi-dimensional data visualization techniques, like parallel coordinates or scatterplot matrices, do not scale very well for high-dimensional data [23, 7]. That results in the need of analyzing the unfiltered and raw high-dimensional data before the interesting information can be presented to the analyst [8].

Explorative overview tasks are relevant in various application domains. Examples include comparing outputs of multi-run simulations in the automotive sector, or monitoring multiple quality indicators of products in advanced manufacturing.

This work is motivated by High-Dimensional Overview (HDO) tasks in the energy sector — a domain where the amount of acquired data is increasing rapidly. Power generation, power consumption, and meteorological quantities are constantly measured by the providers, creating a vast number of time series. The number of sensors will grow even further with the advent of smart meters. Until the year 2020 EU member states are required to equip at least 80% of their consumers with smart meter devices [2]. The transmission system operators need to analyze and process this acquired time series in regular intervals, e.g., weekly or monthly. It is impossible for them to inspect every single acquired time series. Thus, they need to get an overview of the data first and interactively explore the data further to get an insight.

The primary contribution of this paper is a framework for analyzing and comparing high-dimensional data. The key idea is to partition the data by meta-information, and to visualize the resulting subsets by appropriate means. In the energy sector various data sensors share the same type, for example temperature sensors, or are placed at the same location. This meta-information of the data is familiar to the domain experts and allows them to analyze and compare the data in a more intuitive way. An example task would be the comparison of multiple time series of power consumptions of multiple locations, where only the locations are compared and the power consumptions of the different sensors within a location are combined. This scales better than comparing every single data dimension like at the Rank By Feature Framework (RBFF) [18] or comparing every data record like in parallel coordinates or scatterplot matrices. Still, the user is able to flexibly drill-down on demand in order to to explore the details of the different dimensions.

## 2 Related Work

A vast amount of scientific and application areas are confronted with high-dimensional datasets. Interactive visual analysis is an effective way to understand and process the data [7]. Approaches to visualize multi-dimensional datasets are an important topic of research and traditional visualization techniques like parallel coordinates or scatterplot matrices are well suited for targeting this problem for a small number of dimensions [15]. But when the number of dimensions increases, these techniques fail. This is because of the boundaries of our visual system, visual clutter and technical challenges [4].

There are multiple surveys that focus on high-dimensional data visualization. [11, 15] One way to main-

tain the scalability of the data is to reduce the information of the data by the number of data records or the number of data dimensions [16]. Principle Component Analysis (PCA), Multidimensional Scaling (MDS) and Self Organizing Maps (SOM) are common techniques to reduce dimensions in data visualization. [5, 6, 9, 13] The drawback of these methods is that they produce a subspace that has no intuitive meaning to the data analyst.

A technique that supports the generation of meaningful subspaces is called Visual Hierarchical Dimension Reduction (VHDR) [22]. It uses a similarity measure to hierarchically cluster the dimensions and allows the user of this framework to interactive explore and modify the created hierarchy. From this hierarchy clusters a subset is selected as meaningful and representative dimensions of these clusters are visualized as representations. The drawback of this method is that not all dimensions are used for the encoding of the visual representations.

The RBFF by Seo and Shneiderman [18] ranks small preview visualizations of one dimension or two dimensions by statistical properties which gives a good initial overview of all dimensions. This approach scales well for the number of data records, but has limitations regarding the number of dimensions. Especially for the comparison of dimension pairs (a scatterplot matrix) the number of simultaneously displayed pairs increases quadratically, but also in the case of one dimensional statistics the limit of displayed visual representations that can be handled reasonably is a few hundred [17].

Stole and Hanrahan developed the system "Tableau" (former "Polaris") [20]. It introduces a scalable pivoting algebra on meta-information on data records. This work applies this algebra on the partitioning of data dimensions into subsets using the categories of a data dimension as meta-information.

Elmqvist and Fekete presented a model for implementing hierarchical aggregated visualizations [3]. Their proposed guidelines are used in this work to create visual aggregates of the data subsets that are more scalable for the limited perceptual capabilities of a human viewer.

# 3 Tasks and Goals

This section characterizes tasks that are needed for an explorative overview of high-dimensional data in the energy sector (Section 3.1). These are used to derive the goals of the design process of the HDO framework (Section 3.2).

## 3.1 Task Analysis

Transmission system operators in the energy sector acquire time series data from different sensors on a regular basis. These are used for power control and risk management. The inspection and analysis of newly acquired data is hence a frequent, recurring and important activity.

The time used to look at the data can be optimized by identifying the recurring tasks a user needs to fulfill. In this paper I focus on the following tasks:

**T1 - Finding structures** A key task of a data analyst is to get insight into the data and to validate or discard an initial hypothesis. Hypothesis often refer to structures in the data. In contrast to the validation of expected structures, the discovery of new structures is also a user task. Important structures in time dependent data are listed below and clarified by an example:

**Trends** "Is the data increasing or decreasing?", "Does it have recurring peaks, troughs or plateaus?"

**Groups** "Does the data belong to the same group?"

**Modalities** "Is the distribution of the underlying data uni- or multimodal?"

**Outliers** "Is some of the data not fitting the general trend?"

**T2 - Rank by feature** Like described by the RBFF, a user is interested in the dimensions that match a specific feature. These features can be statistical properties like the median of the dimension. An example of this user task would be the exploration of the biggest electrical loads in the electric grid.

**T3 - Assessing the purity of groups** By merging the data dimensions by meta-information into groups, it is necessary to identify if the grouping is sufficient for the user. By characterizing the purity of these meta-information based groups the user is able to make further decisions. An example of this would be the question: "How much does the given grouping coincides with the similarity of the data?"

**T4 - Exploration and Tuning** After the user was able to get an initial overview, further questions regarding the data may arise, which can be solved by exploring and tuning the created groups and receiving more detailed information. An example of this would be the question: "Does the purity of a group gets better when I drill down the group with another partitioner?"

## 3.2 Design Goals

Based on the task analysis three design goals were established. These goals guided the design process of the HDO framework.

**G1 - Visual summaries of groups** To support the user to find structures in the data (T1), efficient visual summaries of groups of (large numbers of) dimensions need to be displayed. The requirement on the summaries is that they give a good reproduction of statistical position, variance and distribution and also the development of them over time or over categories (T3).

**G2 - Flexible drill down and roll-up** With respect to T4 the overview visualizations need to be explored in depth. The concept of drill down and roll-up with respect to "any known structure of the feature space" [21] enables this fast change of the viewing granularity. Also the ranking by a

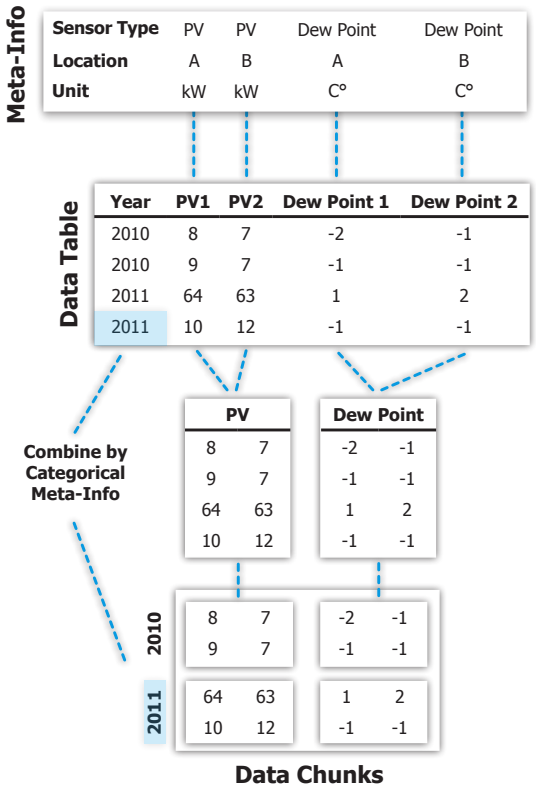


Figure 1: The subdivision of the data table to data chunks by utilizing the meta-information of data dimensions and data records is illustrated.

specific feature (T2) reduces the displayed visualizations to a set of interest.

**G3 - Scalability** Like previously mentioned the number of data dimensions and records is rising. The framework should not be limited by any inherent upper limit of dimensions or data records. This also concerns the visual complexity of the used visualization. The goal of this framework is to support simple monitoring and reporting tasks (T1, T2), but also to allow detailed exploration tasks (T3, T4). This implies a trade-off between the simplicity and cognitive ease of the visualizations and the preserving of the distributions, modalities and outliers of the underlying data.

# 4 Data Model

A raw dataset is most commonly available as a table, made up of columns and rows. In this work a column is referred to as a data dimension and a row is referred to as a data record. Fig. 1 shows a data table, which is used as a guiding example for the data model.

In the energy sector the data dimensions are on one hand numerical time series from various sensors. In Fig. 1 this time series are production values from PhotoVoltaic plants (*PV*) and temperature measurements from *Dew Point* sensors. On the other hand the data dimensions can contain categorical meta-information on the data records (For example the time categories like the year or if the day is a holiday or not). This is shown as the *Year*-column in Fig. 1. Additionally, meta-information can be assigned to the data dimensions themselves. This could be the information of the location of a sensor, the measured unit or the type of the sensor (see top table in Fig. 1).

The numerical data dimensions can be from a similar type with a common scale (e.g.: Multiple power consumption sensors in watt). This enables the comparison for similar distributions or sequences, or the detection of outliers (T1). In contrast to the common scale, the dimensions can also have different units (e.g.: weather time series with temperature, wind speed, wind direction,...) with no common scale. In Fig. 1 the *kW* values of the *PV* sensors are on a different scale than the $C^{\circ}$ values of the *Dew Point* time series.

The assigned meta-information can be present in a hierarchical manner. For data record based meta-information the categorical data dimensions can describe a level of detail (e.g. For time-categories the refinement from year over month to the day of the month). Also for data dimension based meta-information, a hierarchy can be derived. For example, if the location of a sensor is assigned as meta-information, the different levels of detail of the sensor hierarchy can be present as country $\rightarrow$ city $\rightarrow$ house $\rightarrow \ldots$

# 5 Description of the Hierarchical Data Overview

This section describes the visualization method for the HDO framework. Its design is engineered by the defined goals from Section 3.2 for an application to data described in Section 4.

By utilizing the meta-information, described in the previous section, the data table can be subdivided by data dimensions and by data records into smaller blocks of data. In this work these blocks are referred to as *data chunks*. Fig. 1 shows the combination of the data dimensions by the meta-information *Sensor Type* to the data chunks *PV* and *Dew Point*. Not shown but also possible subdivisions would be the combination of the dimensions by the *Location* or the *Unit* meta-information.

The data chunks at the bottom of Fig. 1 are created by the additional subdivision of the previous created data chunks by the combination of meta-information on data records (In this example by the *Year* of the time series). The key concept of this framework is to create a hierarchical relationship between these data chunks and visualize a combination of them.

The overview visualization is designed using a hierarchical tabular layout (see Fig. 3). The design decision of using a table-oriented display enables independent visual encodings of different aspects of the displayed data [10].

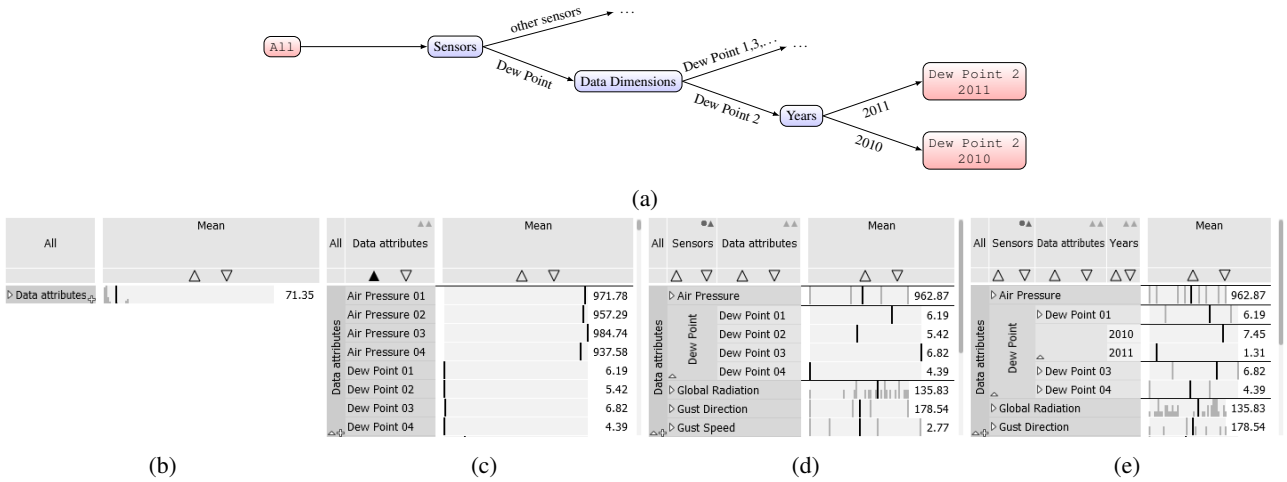


Figure 2: (a) shows the drill down of the hierarchy to the chunks with 3 hierarchy levels. (b-e) visualize the drill down in the framework with the means of the chunks. (b) The hierarchy is collapsed and only the all partitioner is shown. (c) Expanded by the data dimensions. (d) Refinement by the sensor type. (e) Refinement by the categorical attribute of the year. This is the same state as shown in (a)

Additional previous work showed that the users are familiar with this kind of layout [1].

The tabular layout consists of two orthogonal parts: Rows (Section 5.1) and columns (Section 5.2).

A *Row* defines a combination of data chunks and a *Column* is responsible for creating a visual summary of a descriptive quantitative feature of the data chunks. (G1).

## 5.1 Table Rows

The hierarchical structure is a key concept to ensure the scalability (G3) of the HDO framework [3]. The system enables the user the interactive hierarchical definition of the rows of the tabular layout.

Visually the first part of the tabular layout of the framework corresponds to the hierarchy of the data chunks. One row of this table is a node or a leaf of this hierarchy. In Fig. 3b one row shows a single data dimension "*Dew Point 04*" and in Fig. 3c one row shows all data dimensions with the same meta-information "*Dew Point*". The table headers (Fig. 3c) of the left hand part show the different hierarchy levels. The first column, marked with "All" in the example, is the root node of the hierarchy, the second column the sensor type and the third column the data dimensions.

The Fig. 2 visualizes the drill down of the data table into a hierarchy of data chunks. The displayed column "*Mean*" plots the mean values of the underlying data chunks (see Section 5.2). Initially the hierarchy consists only of one node, which contains all assigned data dimensions (Fig. 2b).

Additional hierarchy levels refine the data table into data chunks (G2). A usual refinement is the partitioning of the data table by data dimensions. Fig. 2c shows the table with all assigned data dimensions (attributes) which is a similar layout to the RBFF. However, too many dimensions are assigned to the visualization to display all of them on the limited screen area.

By combining data chunks, the number of rows can be reduced. In Fig. 2d the data dimensions are combined according to their sensor type. One can see that the number of displayed data chunks increases. For example, the *Air Pressure* sensor type contains four dimensions which are plotted in the mean column (see Section 5.2). In Fig. 2e the level *Year* is assigned to the hierarchy. As previously described, this level partitions the data records of the data chunks by the categorical meta-information (see Fig. 1). The number of chunks has increased, as the visualization of the dimension *Dew Point 01* shows.

The user is able to control the order and the level-of-detail of the levels (see Section 6). Additionally, the visible set of rows can be defined by collapsing and expanding the hierarchy nodes individually. Fig. 2e shows the expanded node "*Dew Point*" of the sensor type hierarchy level with an additional refinement for the dimension *Dew Point 02*. The Fig. 2a visualizes this drill down by a graph representation.

The interactive refinement supports the identified G3 by enabling a visual scalability for a high number of dimensions and still allows the user to explore and tune (T4) the data.

## 5.2 Table Columns

The orthogonal part of the tabular layout is the visual encoding of different aspects of the data chunks as columns of the table. For example in Fig. 2 the mean of a data chunk was visualized as lines inside the cells of the column.

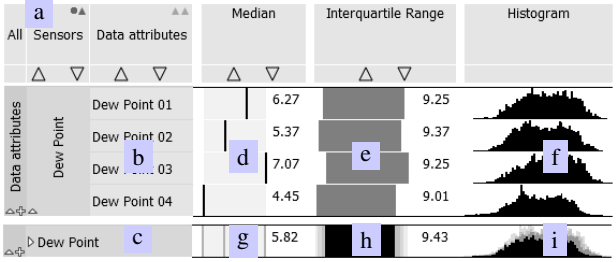


Figure 3: The tabular layout of the framework.
(d,g) Central tendencies are shown as lines.
(e,h) Dispersions are shown as areas.
(f,i) Frequency distributions are shown as histograms.

#### 5.2.1 Visual Aggregation Aspect

One column is responsible for calculating and visualizing a specific descriptive statistic for every row. These statistics are used to quantitatively describe and summarize different features of the defined data subsets (G1) [12]. The system differs three different classes of descriptive statistics: Central Tendency, Dispersion and Frequency Distribution. Fig. 3 shows the visual encodings of these three classes.

A simple way to visualize a statistic feature is by a textual representation. However, this representation does not give a good reproduction of the position, scattering and distribution (G1) to the other displayed values. By plotting the features of the data chunks onto an axis a more descriptive representation can be achieved, because one visual aggregate can be visually compared with another in an intuitive way.

**Central Tendency** A univariate statistic can describe a point inside an axis. This includes the central tendency Examples of this class of statistics are the average, the extreme values (minimum, maximum) or the percentile values (Median,...). Like other univariate statistics they can be used for ranking (T2) but also for other tasks like finding outliers (T1). A textual representation of the aggregation, as shown in Fig. 3d on the right hand side of the cell, enables the user to receive the actual value. To visualize the value of the aggregation, a line is positioned between the extents of the axis. This enables the user to compare the value to one data chunk with another (T1). When multiple data chunks have to be visualized within one cell, the same visual aggregate can be used. The location of the underlying features is plotted as gray line, and the combined statistic is shown as black line (see Fig. 3g). If more than one line is drawn at the same location, the visualization turns to a bar chart as shown in the *mean* cell of the *Global Radiation* row of Fig. 2e.

**Dispersion** The second class of univariate statistics is the dispersion. It describes a positive range inside the axis. Examples for this class of descriptive statistics are the standard deviation or the InterQuartile Range (IQR). Similar to the previous encoding, an area is positioned around a dependent first moment inside the axis of the aggregation (e.g.: Area in the size of IQR around the median, Fig. 3e). When the dispersion of multiple data chunks have to be visualized within one cell, the range of the underlying ranges can be visualized in an aggregated cell by over-plotting the areas (see Fig. 3h).

**Frequency Distribution** An example of a frequency distribution is the histogram. A textual representation is no longer suitable, because multiple values are visually encoded. To support details on demand (T4), detailed information of a bin value of a histogram can displayed with tool-tip information. In contrast to the previous classes, it is not possible to rank the data chunks by a frequency distribution.

The binning of the histogram depends on all data chunks inside the axis (see Fig. 3f). When the frequency distribution of multiple data chunks have to be visualized within one cell the histograms of the underlying data chunks are plotted over each other. The darker a part of a bin (as shown in Fig. 3i) the more histograms overlap at this position.

#### 5.2.2 Partitioning Aspect

Another important information is the development of a descriptive statistic over time (T1). The system supports the partitioning of a column into sub-columns. Then the descriptive statistics for the data chunks are calculated for every partition of the column. This provides a global overview over local relationships of the statistic features (G1) [14].

Fig. 4 shows a partitioning of the column into time intervals (in this example thirds of months). For every row and every partition, the values from the three classes that were proposed in the previous section are computed and visualized:

**Central Tendency** The calculated descriptive statistic of every partition is connected with a line, resulting in a line chart for every data chunk. Multiple lines are drawn inside one cell if multiple data chunks contain to one row (see Fig. 4b). This representation can be used to analyze the trend of the data or to find outliers (T1).

**Dispersion** Similar to the simple visual aggregate, an area is drawn which is centered at every partition at a central tendency. Its upper and lower boundaries are defined by the value of the dispersion. The upper and lower boundaries for all partitions are connected, and the area between them is filled (see Fig. 4c). This representation can be used to characterize the purity of the underlying data (T3).

**Frequency Distribution** The frequency distribution of all data chunks inside a sub-column is calculated. A heat map visualization is used to encode the distribution inside one sub-column (see Fig. 4d). This representation can be used to analyze the modalities of the underlying data chunks (T1).

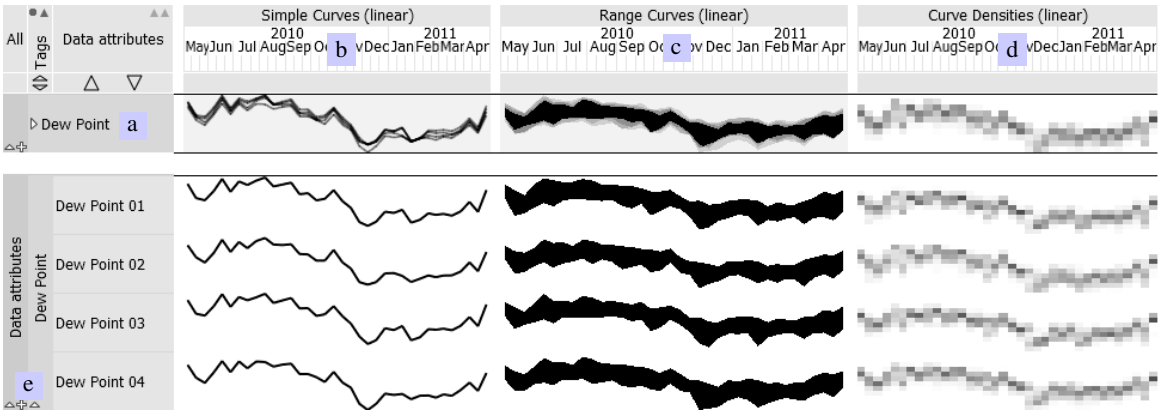


Figure 4: The HDO framework is visualized by a tabular layout. The left hand side defines the rows of the visualization, which are defined by the hierarchy of the data model. (b-d) The columns define which visualizations are displayed in the cells of the rows. A cell can visualize one leaf (e) of the hierarchy, but is also able to visualize a node (a) by combining the underlying leafs.

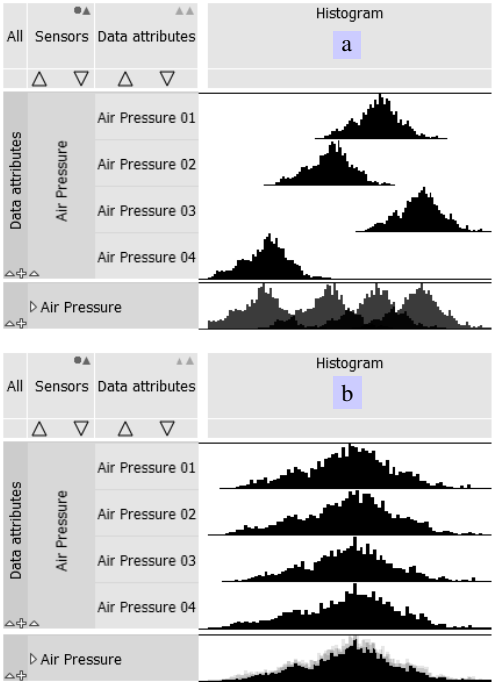


Figure 5: (a) shows data chunks on a common scale (▲▲) which makes it possible to compare their position.
(b) If the level is set to ▲▲, the comparison of the shape of the data chunks is possible.

#### 5.2.3 Combination Aspect

All shown examples have used the same axis for creating the visual aggregate. A prerequisite is, however, that not all data chunks share a common scale or are combinable.

**Combination** The created hierarchy levels, as described in Section 5.1, can have the property that they separate the data into distinctive nodes, in which the combination of the data chunks have no useful meaning. In the visualization this is indicated by displaying the icon ●▲ in the header of the hierarchy level (see Fig. 3a). An example of not combinable nodes is the partitioning of the data by the sensor type of the underlying dimension, in which it makes no sense to plot a temperature and a voltage value in the same coordinate frame.

However, a level can also separate the data into chunks that are comparable with each other (for example the partitioning of power consumption time series by their location). This is indicated with the icon ▲▲ in the header.

**Scaling** If two nodes of a hierarchy are set to not comparable (●▲), they can not share a common axis. The common idiom "*comparing apples and oranges*" states that a not suitable comparison would indicate a false analogy.

In contrast to this idiom, if two nodes are visualized that are set to be comparable, the scaling of the visualizations needs to be defined. One possibility is that the two nodes can share a common scale, which is indicated with ▲▲. Another option is that every node may use its own scale and the visualizations are only displayed overlayed or in juxtaposition (▵▵).

Fig. 5 compares these two scaling aspects. If the hierarchy level *data attributes* is set to a common scale (▲▲), it is possible to compare the position of the underlying data chunks (see Fig. 5a). As opposed to this, it is not as easy to compare the shape of the frequency distributions of the different data chunks (T1). To be able to compare the shape the level can be set to no common scale (▵▵). Fig. 5b shows that the shapes of the frequency distributions of the different data chunks are similar.

## 6 Exploring the Hierarchy

This section describes how users are able to interact with the framework to be able to address their explorative overview tasks. As prerequisite it is assumed that the domain expert knows what data types, sensors and meta information one can expect from the data. Typically a user

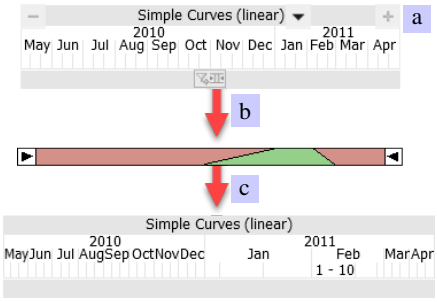


Figure 6: Controls to configure a column that are displayed on demand. (a) adding new columns. (b) opens controls to change the mapping of the displayed interval. (c) shows the changed mapping.

does not start his exploration task with no initial hierarchy, like shown in Fig. 2b. A pre-defined set of hierarchy levels that are relevant in his domain is assigned. To stay with the guided example of sensor data, the initial hierarchy could be the sensor type, and the underlying data dimensions, which are collapsed so that the user is able to get an overview of the data (see Fig. 2d).

The exploration process supports the extension of the complexity of the displayed visual representations in multiple ways: First, it is possible to increase the visual aggregation aspect of the data. For example, if initially only the central tendency of the data chunks is considered, the apposition of a column that visualizes the dispersion or the frequency distribution increases the visual complexity, but also enables the user to find structures (T1). To add a new column, a user may hover a table header with his mouse. Pop-up controls are displayed on demand, and by clicking on the plus sign (Fig. 6a) a new column can be selected.

The second complexity increase is the partitioning of the columns into sub-columns like described in Section 5.2.2. This enables the user to analyze the trend of the data chunks.

While exploring the visualizations, a user may observe that the variance in some displayed nodes is very high, and one wants to drill down the node to see if the purity of the underlying nodes in the next hierarchy level increases (T3). This is done by clicking on the small arrows of the displayed hierarchy nodes (Fig. 4e). Fig. 2e shows this drill down of multiple hierarchy nodes to a detailed representation.

Every column can be configured individually. For example if the user is only interested in the time series of January to February of the year 2011, one can zoom into this interval with controls for every column (T4). These controls are displayed when hovering the table header. These controls enable the user to change the mapping of the displayed interval interactively like shown in Fig. 6.

Additionally the user is able to refine the hierarchy further by adding more hierarchy levels by clicking on the plus sign in the lower left corner (Fig. 4e) or changing the ordering of the hierarchy levels by dragging the headers to another position.

## 7 Implementation Aspects

This framework has been implemented in C++ and uses OpenGL for rendering. The visual feedback of the visualization should be in real-time to help the user to explore the data faster and thereby support him to make decisions faster. However, the computational cost of measurements on high-dimensional data is expensive. Several possibilities to achieve real-time capability can be utilized: The hierarchy is used to compute the results of the measurements on higher levels of the hierarchy by a bottom up approach. This is for example possible for the statistic mean of values. Unfortunately, not all statistics can be calculated by reusing intermediate results of lower levels. For example, the quantiles of multiple chunks in an intermediate node of the hierarchy need to be recalculated using the data of the combined chunks.

To support these expensive tasks, the joint computation of results for multiple dimensions is considered. For example, a data dimension is sorted only once for all measures that need a sorting of subsets of the dimension.

## 8 Conclusion

I described a framework to efficiently inspect and compare high-dimensional data. Motivated by the tasks of domain experts in the energy domain, I defined three design goals for this framework: Visual summaries, flexible interaction and the scalability for high dimensional data.

Based on these goals I described the HDO visualization, which utilizes meta-information of the assigned data dimensions to partition the dimensions into data chunks. In a tabular layout multiple descriptive statistics of these chunks can be visualized. I distinguished between two aspects of scaling, to combine the visual representations of the descriptive statistics: If they have a common scale ▲▲ or no common scale ▲▲.

To support the interactively exploration tasks of domain experts, I described the high configurability of the framework. By the interactive refinement of the displayed rows and the flexible configuration of the columns of the tabular layout I showed that the framework is able to address the posed goals.

## Acknowledgments

## References

[1] Clemens Arbesser, Florian Spechtenhauser, Thomas Mühlbacher, and Harald Piringer. Visplause: Visual data quality assessment of many time series using

plausibility checks. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):641–650, 2017.

[2] Council of European Union. Council regulation (EU) no 189/2014, 2014. http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=SWD:2014:189:FIN.

[3] Niklas Elmqvist and Jean-Daniel Fekete. Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *IEEE Transactions on Visualization and Computer Graphics*, 16(3):439–454, 2010.

[4] J-D Fekete and Catherine Plaisant. Interactive information visualization of a million items. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, pages 117–124. IEEE, 2002.

[5] Arthur Flexer. On the use of self-organizing maps for clustering and visualization. *Intelligent Data Analysis*, 5(5):373–384, 2001.

[6] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[7] Daniel A Keim, Jörn Kohlhammer, Geoffrey Ellis, and Florian Mansmann. *Mastering the information age-solving problems with visual analytics*. Florian Mansmann, 2010.

[8] Daniel A Keim, Florian Mansmann, Jörn Schneidewind, and Hartmut Ziegler. Challenges in visual data analysis. In *Information Visualization, 2006. IV 2006. Tenth International Conference on*, pages 9–16. IEEE, 2006.

[9] Teuvo Kohonen. Self-organizing maps, volume 30 of springer series in information sciences, 1995.

[10] Alexander Lex, Nils Gehlenborg, Hendrik Strobelt, Romain Vuillemot, and Hanspeter Pfister. Upset: visualization of intersecting sets. *IEEE transactions on visualization and computer graphics*, 20(12):1983–1992, 2014.

[11] Shusen Liu, Dan Maljovec, Bei Wang, Peer-Timo Bremer, and Valerio Pascucci. Visualizing high-dimensional data: Advances in the past decade. *IEEE Transactions on Visualization and Computer Graphics*, 2016.

[12] Prem S Mann. *Introductory statistics*. John Wiley & Sons, 2007.

[13] A Mead. Review of the development of multidimensional scaling methods. *The Statistician*, pages 27–39, 1992.

[14] Thomas Mühlbacher and Harald Piringer. A partition-based framework for building and validating regression models. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):1962–1971, 2013.

[15] Tamara Munzner. *Visualization Analysis and Design*. CRC Press, 2014.

[16] Harald Piringer. *Large data scalability in interactive visual analysis*. PhD thesis, Institute of Computer Graphics and Algorithms, Vienna University of Technology, 2011.

[17] Harald Piringer, Wolfgang Berger, and Helwig Hauser. Quantifying and comparing features in high-dimensional datasets. In *2008 12th International Conference Information Visualisation*, pages 240–245. IEEE, 2008.

[18] Jinwook Seo and Ben Shneiderman. A rank-by-feature framework for interactive exploration of multidimensional data. *Information Visualization*, 4(2):96–113, 2005.

[19] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343. IEEE, 1996.

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

[21] Cagatay Turkay, Arvid Lundervold, Astri Johansen Lundervold, and Helwig Hauser. Representative factor generation for the interactive visual analysis of high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2621–2630, 2012.

[22] Jing Yang, Wei Peng, Matthew O Ward, and Elke A Rundensteiner. Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets. In *Information Visualization, 2003. INFOVIS 2003. IEEE Symposium on*, pages 105–112. IEEE, 2003.

[23] Jing Yang, Matthew O Ward, and Elke A Rundensteiner. Visual hierarchical dimension reduction for exploration of high dimensional datasets. 2002.