FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Visualisierung hochdimensionaler Daten mit hierarchischer Gruppierung von Teilmengen

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Masterstudium Visual Computing

eingereicht von

David Pfahler
Matrikelnummer 1126287

an der Fakultät für Informatik
der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Univ.-Doz. Dipl.-Ing. Dr.techn. Eduard Gröller
Mitwirkung: Dipl.-Ing. Dr.techn. Harald Piringer
Dipl.-Ing. Dr.techn. Thomas Mühlbacher

Wien, 1. Oktober 2019

David Pfahler

Eduard Gröller

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Visualizing High-Dimensional Data with Hierarchically Aggregated Subsets

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Master Visual Computing

by

David Pfahler
Registration Number 1126287

to the Faculty of Informatics
at the TU Wien

Advisor: Ao.Univ.Prof. Univ.-Doz. Dipl.-Ing. Dr.techn. Eduard Gröller
Assistance: Dipl.-Ing. Dr.techn. Harald Piringer
Dipl.-Ing. Dr.techn. Thomas Mühlbacher

Vienna, 1st October, 2019

David Pfahler

Eduard Gröller

Erklärung zur Verfassung der Arbeit

David Pfahler
Breyerstraße 7/1/5 2500 Baden

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 1. Oktober 2019

David Pfahler

Acknowledgements

I would first like to thank my supervisor Dr. Harald Piringer of the VRVis Research Center[1] (VRVis) in Vienna. He provided scientific guidance and always suggested helpful improvements. Second, I wish to thank my official supervisor Meister Edi Gröller from TU Wien for his encouraging feedback and great support. Furthermore, I would like to acknowledge Dr. Thomas Mühlbacher of VRVis for reviewing and supporting me during the technical development of this thesis.

On a personal level, I must express my very profound gratitude to my parents for providing unfailing support and continuous encouragement throughout my years of study and through the process of research and writing this thesis.

[1] VRVis Zentrum für Virtual Reality und Visualisierung Forschungs-GmbH, Vienna, `www.vrvis.at`

Kurzfassung

Die Anzahl der installierten Sensoren zur Erfassung von Daten, z. B. Stromzähler in Smart Grids, nimmt rasant zu. Diese riesige Menge an gesammelten Daten muss von den Übertragungsnetzbetreibern analysiert und überwacht werden. Diese Aufgabe wird durch Visual Analytics Techniken unterstützt, aber traditionelle multidimensionale Datenvisualisierungstechniken skalieren nicht sehr gut für hochdimensionale Daten. Der Hauptbeitrag dieser Arbeit ist ein Rahmenwerk, um solche hochdimensionalen Daten effizient zu inspizieren und zu vergleichen. Die zentrale Idee ist es, die Daten durch die Semantik der zugrundeliegenden Datendimensionen in Gruppen zu zerteilen. Fach-Experten kennen die Metainformationen der Daten und können diese Gruppen in eine Hierarchie strukturieren. Das System berechnet aus den Gruppen statistische Eigenschaften, welche dann visualisiert werden. Diese visuellen Repräsentationen können verwendet werden, um die analytischen Aufgaben des Benutzers zu unterstützen.

Abstract

The number of installed sensors to acquire data, for example electricity meters in smart grids, is increasing rapidly. The 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 thesis is a framework to efficiently examine and compare such high-dimensional data. The key idea is to divide the data by the semantics of the underlying dimensions into groups. Domain experts are familiar with the meta-information of the data and are able to structure these groups into a hierarchy. Various statistical properties are calculated from the subdivided data. These are then visualized by the proposed system using appropriate means. The hierarchy and the visualizations of the calculated statistical values are displayed in a tabular layout. The rows contain the subdivided data and the columns visualize their statistics. Flexible interaction possibilities with the visual representation help the experts to fulfill their analysis tasks. The tasks include searching for structures, sorting by statistical properties, identifying correlations of the subdivided data, and interactively subdivide or combine the data. A usage scenario evaluates the design of the framework with a data set of the target domain in the energy sector.

Contents

CHAPTER 1

Introduction

In the year 1971 the first commercial microprocessor chip, the Intel 4004, was launched. Thanks to the silicon technology it contained 2,300 transistors [FK70]. The improvement of this technology led to an exploding growth of computing power and storage capacities (Moore's law [Moo65]). With these improvements the speed of data generation has increased every year. This increment of raw and unfiltered data concerns multiple application areas. For example, the world's largest particle accelerator, the Large Hadron Collider (LHC), generates a petabyte (PB) (10^{15} bytes) of unfiltered data per second [Bru11].

In addition to the increase of data generation per device, the number of devices is rapidly increasing. For example, in the energy sector smart metering devices are installed to measure energy consumption. Table 1.1 shows the fast growth of data in this area. If the energy provider collects data from one million smart metering devices, every 15 minutes, for a year and a single data record is assumed to have 5 kilobyte (kB), the generated data table would have a volume of 2.9 PB [ZFY16].

Traditional data processing applications are not designed to deal with this amount of data. Problems may be the computational complexity or the physical memory of the hardware. Even if the application or hardware would be able to handle this, the user who is analyzing the data, is no longer able to look at and analyze the whole input data. This problem is called information overload and leads to a reduced quality

Table 1.1: The potential amount of generated data from 1 million of smart meter devices in a year. A data record is assumed to have 5 kB. The aggregated data volume is given in terabyte (TB) [ZFY16]

Collection Fequency	1/day	1/hour	1/30 min	1/15 min
Data record (billion)	0.37	8.75	17.52	35.04
Volume of data (TB)	30.42	730	1460	2920

of the decision-making process for tasks of data analysts. Typically, this is caused by parts of the data being irrelevant for the particular task or inappropriately processed or presented [KAF+08]. One approach to overcome the information overload is information visualization. The data is mapped to a visual form, which then can be interpreted by a human in a fast and intuitive way [Kei02]. An orthogonal approach is to utilize automated data analysis methods and tools. These tools outperform the human analysts if the task and problem is well-defined and well-understood.

Defining a problem can become very cumbersome and difficult for complex data and tasks. While trying to understand it in more detail, the problem definition may change. Hence, a combination of automated data analysis methods and information visualization is considered, which is called visual analytics. Whereby the strengths of both, the human processing of visualizations and the electronic data processing are utilized to overcome the information overload.

1.1 Guiding Data Example: Photovoltaic Production

In the energy sector very diverse data are collected. These can include customer, electricity generation, or electricity consumer data. The generation and consumer data are numerical time series from different sensors.

This thesis uses a data set from this domain that is employed in all examples. It contains time series from various meteorological sensors and power Production Values from 95 photovoltaic power plants (PV01-PV95). The meteorological data consists of hourly measurements from 20 weather stations of global radiation and temperature, as well as humidity, wind speed, gust speed, wind direction, air pressure, and dew point for four weather stations.

Fig. 1.1 shows a single time series of a photovoltaic power plant. The PV of the first plant are shown for a whole year. In Fig. 1.1b one can see the values for a single day (02.10.2010).

The time series are from real, but anonymized, data measurements in the period May 2010 to April 2011. They are chosen for this thesis, because they contain multiple time series from diverse sources with meta-informations. The meta-information contains the sensors of the data dimensions (see above), the location of the photovoltaic power plants and the regions of the meteorological sensors. By grouping the data into these meta-informations a hierarchy can be created. For the example data set this may be grouping the time series by their sensors, by their location or by both.

1.2 Motivation

In various application domains often recurring and thereby relevant user tasks are exploration overview tasks. Examples include comparing outputs of multi-run simulations in the automotive sector or monitoring multiple quality indicators of products in advanced manufacturing.

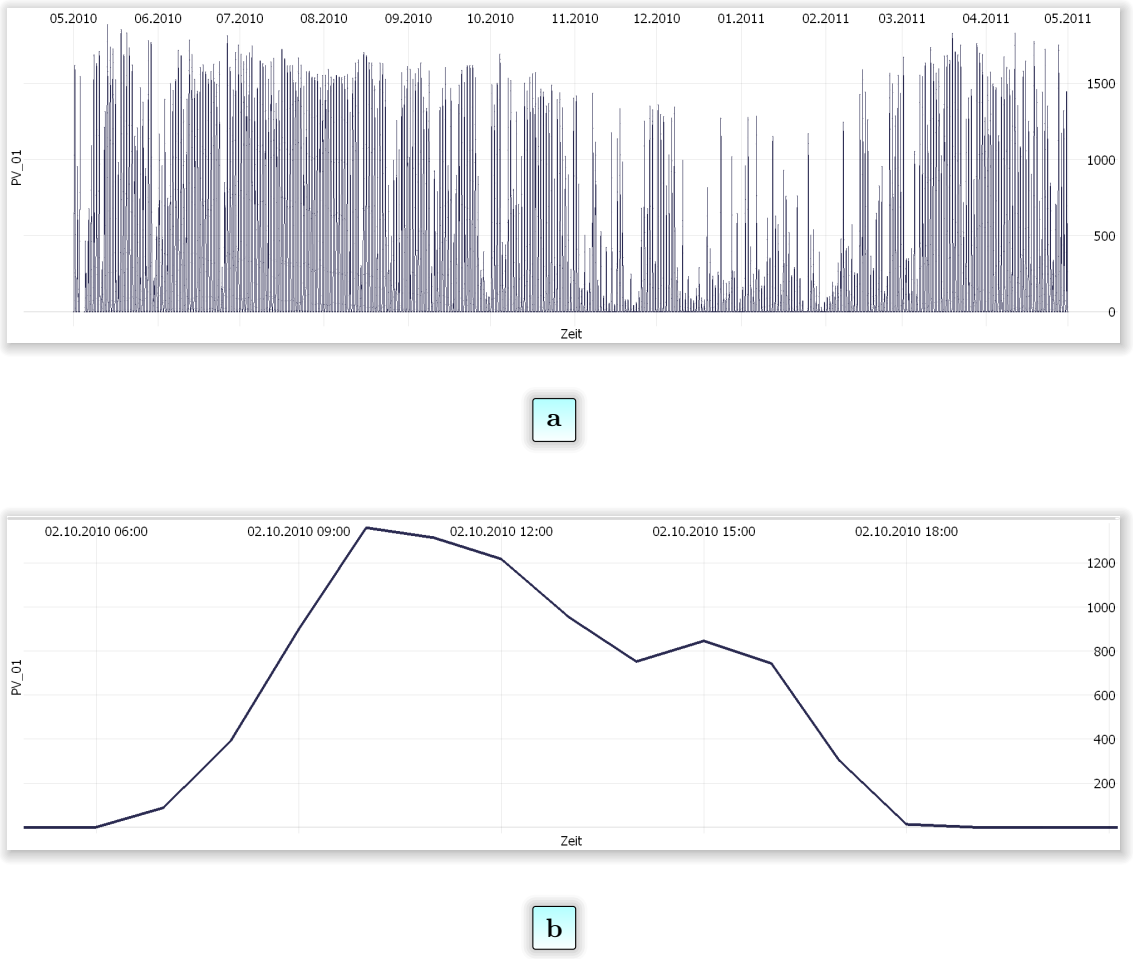

Figure 1.1: Photovoltaic power Production Values (PV). (a) shows the PV_01 for the whole available time interval. (b) shows a single day.

This thesis is motivated by Hierarchical Data 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 guiding data example shows such data (Section 1.1).

The number of sensors will grow even further with the advent of smart meters. EU member states are required to equip at least 80% of their consumers with smart meter devices until the year 2020 [oEU14]. The transmission-system operators need to analyze and process this acquired time series. It is inefficient or even impossible for them to inspect every single acquired time series.

There may be very different questions the system operators of the energy sector want to answer with the data. Different methods may help answer these questions:

- Finding trends, groups, modalities and outliers, helps the operator understand the structure of the data.
- Statistics are used to score interesting features of the measured sensors. Ranking them by these features helps finding the interesting ones.
- In order to gain control of the amount of data, it is useful to group the time series into contiguous groups and structure them hierarchically.
- Through these analyses of the users, they could have derived some conclusions from the data. Additional analysis is needed to confirm them. By interactively exploring the data further and tuning the parameterization, more detailed information may be obtained.

By creating a framework that supports these methods the efficiency of decision making for transmission-system operators can be improved.

1.3 Contributions

The primary contribution of this thesis is the validated design of a framework for analyzing and comparing high-dimensional data. The key idea is to partition the data by meta-information. For each resulting subset statistics are computed. The statistics are differentiated into types, which are central tendencies, dispersions and frequency distributions. Depending on the type of the statistic different visualization techniques for the computed values are used. These are then shown in a tabular layout.

In the energy sector, various data sensors are placed in the same location or share the same type, for example, temperature sensors. This meta-information of the data is familiar to 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 consumption at multiple locations, where only locations are compared and power consumption of the different sensors within a location are combined. This approach scales better than comparing every single data dimension like in the Rank By Feature Framework (RBFF) [SS05] or comparing every data record like with parallel coordinates or scatterplot matrices. Still, the user is able to flexibly drill-down on demand in order to explore the details of the different dimensions.

1.4 Structure of the Thesis

The design process of the HDO visualization can be structured into development phases. The design-study methodology by Sedlmair, Meyer and Munzner is used as guideline, to identify the phases [SMM12]. Fig. 1.2 shows the nine involved phases and classifies them into three top-level categories. These phases are also used to structure the thesis.

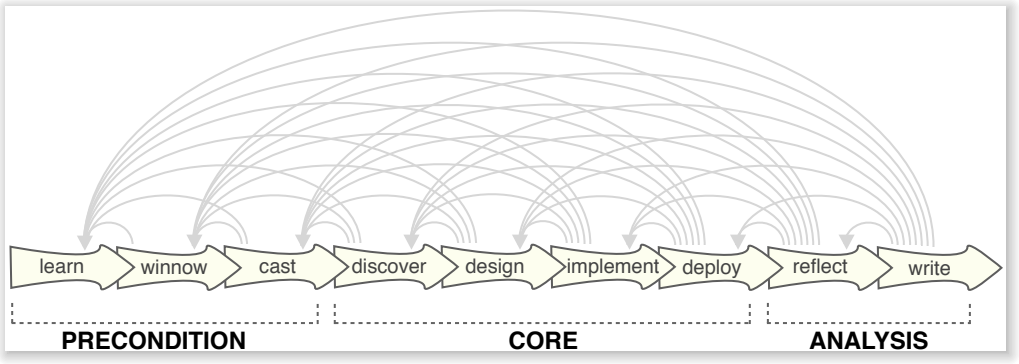

Figure 1.2: The nine-stage design-study methodology-framework classifies the design process into three top-level categories. Whereby the process is not a linear approach, but an iterative and dynamic process [SMM12].

1.4.1 Precondition Phase

In the precondition phase the related work of HDO tasks was studied to learn about existing approaches (see Chapter 2). The user group of transmission-system operators was identified as promising collaborators.

1.4.2 Core Phase

In the core phase this user group was consulted to identify their tasks and based on these tasks the design goals were established (see Chapter 3). Additionally, the high-dimensional data of the application area (the energy sector) were analyzed and abstracted into a data model (see Chapter 4).

Based on these goals and the data the HDO framework was designed. This design includes on the one hand the visual representations (see Chapter 5) and on the other hand the interaction techniques (see Chapter 6). In the end of the core phase the framework was implemented and integrated into the existing software Visplore (see Chapter 7).

1.4.3 Analysis Phase

The last phase focusses on the reflection of the work done. In order to present the results of the design process, a usage scenario was written, which describes the analysis of a smart-meter data set on the basis of the implemented visualization (see Chapter 8). Additionally, the lessons learned are discussed and possible extensions of the framework are presented (see Chapter 9). To conclude the analysis phase this thesis was written as a documentation of the design process.

CHAPTER 2

Related Work

A vast number of scientific and application areas are confronted with high-dimensional data sets. Visualization is a method to analyze this data.

This chapter provides an introduction to visualization (Section 2.1), with the focus on information visualization of multi-dimensional data, like parallel coordinates (Section 2.1.4) or scatterplot matrices (Section 2.1.4). These techniques are well suited for exploring a few data dimensions [Mun14]. If the number of data dimensions increases, these techniques fail, because of the limitations of our visual system, visual clutter, and technical challenges [FP02].

One possibility to overcome these problems is to integrate automatic data analysis methods into the visual data exploration process. Section 2.2 introduces visual analytics, which is an effective way to understand and process high dimensional data [KKE10].

To maintain the scalability of visualization techniques, the reduction of displayed information is needed [Pir11]. Section 2.3 presents techniques of reducing information on a data record and on a data dimension level and introduces different previous visual analytics frameworks for overview visualizations.

The common task of relating one data record to anotherone is essential for exploring data in an information visualization (see Section 2.1.1). Section 2.4 presents different techniques of comparing visualizations.

2.1 Introduction to Visualization

The graphic representation of information has been used to communicate messages since the early evolution of mankind (e.g., cave paintings). Early data visualizations occurred in the form of maps and diagrams to aid in navigation and exploration [Fri08]. In the field of computer science the term visualization refers to a technique for creating images, diagrams or animations of not immediately visible data to communicate an idea or a message [KDHL08].

The created, computer supported, interactive, visual representations are used to amplify human cognition. The human visual system and brain capabilities are then utilized for hypothesis building and reasoning [CMS99]. Coupling the strength of the human visual system with interaction techniques and the visual representations of data, supports the understanding and decision-making process.

Visualization is subdivided into three fields, scientific visualization, information visualization and visual analytics. The separation into the three fields is hard, because they share common goals and techniques. Possible distinctions are the used data and mappings.

Scientific visualization: The underlying data of the graphical representations in scientific visualization involves information with an inherent physical component [TM04]. Hence, the object space of the data and its mapped visual representations is 1D, 2D, 3D, or 4D. Application areas of scientific visualization are flow visualization, volume visualization and others.

Information visualization: In contrast to the spatial environment of scientific visualization, information visualization depicts abstract data with multiple dimensions. Examples for abstract data are business data, social networks, and process data.

Hence there exists no spatial mapping from real-world data to the virtual world, additional steps have to be performed to create a visualization. To be able to map the multi-dimensional data to the 2D or 3D screen space a visual mapping to a visual metaphor has to be created [GP01].

Additionally, standard diagrams, such as x-y plots and bar charts, are not flexible enough for multi-dimensional data. To overcome these problems new diagrams were developed (see Section 2.1.4).

Visual analytics: A definition of visual analytics is "the science of analytical reasoning supported by the interactive visual interface" [Tho05]. This field of research differs from information visualization in the preceding data analysis methods, such as statistical calculations or data mining. Furthermore, the focus is on the interaction between man and computer. The scope of this thesis focuses on the design of an visual analytics technique and thus this field of research is discussed in more detail in Section 2.2.

An approach to design a novel technique is to start with a problem characterization and abstraction [SMM12]. First the domain problem is characterized by an abstraction of the tasks and second the goals of the approach for dealing with the problem are identified. In the subsequent section the tasks and goals of visualizations in general are introduced. This provides visual-design guidelines for creating an information-visualization application. In Chapter 3 a domain specific problem is characterized through its tasks and goals.

2.1.1 Tasks

The visual information seeking mantra of Shneiderman ("Overview first, zoom and filter, then details-on-demand" [Shn96]) provides an abstract task analysis for designing an information visualization technique called the Task by Data Type Taxonomy (TDTT). Whereby the tasks of the visual information seeking mantra are enhanced by the tasks relate, history, and extract. These tasks apply to multiple defined data types, but have to be adjusted for the specific properties of seven identified data types: 1D-, 2D-, 3D-, multi-dimensional, temporal, tree, and network data [Shn96]. This thesis focuses on multi-dimensional data, thus examples for this type are given in the description of the identified abstract tasks of the TDTT.

Overview: To get an initial understanding of the data set the whole context of the data needs to be represented on the screen space. This helps to see global patterns and structures in the data, like clusters or outliers [CC05].

Zoom and Filter: Since the screen space is limited, it is necessary to exclude unimportant information in the overview representation. Zooming into the representation or filter away unimportant information, helps to overcome this limitation. Examples for zooming techniques are geometric, fisheye, and semantic zooming.

Details-on-demand: Often it is impractical to change the visual representation (e.g., by zooming or filtering), but it is still necessary for the user to be able to access every detail, even if it is not visualized. The interaction technique details-on-demand provides this detailed information only when the user requests it. A well-known example of details on demand are tooltips, which are small pop-up windows that appear when the mouse pointer is moved over a specific area.

Relate: Visualizing the relation of a data record to another helps the user to find similarities. A technique that emphasize the relation is called brushing and linking. Whereby the change of a filter in one representation reflects to all representations that encode the same affected data records.

History: To compare the current state of the analysis of a user with a previous state, it is necessary to step back and forth in the history of changes. Additionally, it enables the recovering of errors made by the user.

Extract: After the data was analyzed and a set of interesting data records was obtained, the user wants to extract and share these insights. For example by saving the set as a file.

This thesis focuses on the *Overview* task of the visual information seeking mantra. However, as Shneiderman describes, all other tasks are also relevant for designing an effective information visualization technique and are considered in this thesis as well.

2.1.2 Goals

Keim identified three main goals when visually exploring data sets [Kei97].

Exploratory analysis: Users explore the data and information to get new (unexpected, profound) insights. The results of this search is used to generate an initial hypothesis. This includes tasks like finding structures, for example, trends, groups, modalities, or outliers [Yu77].

Confirmatory analysis: If the user found a hypothesis, she wants to verify or reject it. To support this process, the visualization design guides the exploration of the user.

Presentation: To communicate the found insights, the user wants to illustrate them. This involves the selection and export of an appropriate visualization technique, which is intuitive and self explanatory for the target audience.

The scope of this thesis is the exploration of the data, whereby the exploratory and the confirmatory analysis are desired goals in the proposed visualization design.

2.1.3 Visualization Pipeline

Fig. 2.1 shows a reference model for visualization. The model is often referred to as the visualization pipeline, whereby it is only a simplification of an information visualization system. The individual stages can be used to simplify the discussion of such systems and to compare them. It describes the steps to generate an image from the memory-stored data and where the user may interact with these steps [CMS99]:

Data transformations map the *raw data* to *data tables*. This includes reducing the data and including meta-information.

Visual mappings transform *data tables* into *visual structures*. Standard visualizations define a visual mapping for every data dimension, which maps every data record of this dimension to its visual representation. This data record is called a visual data record. For example, if a data dimension is assigned to the x-axis of a scatterplot, every data record of this dimension is mapped to a specific x-position.

View transformations create *views* of the *visual structures*. Since interactive visualizations are not static, it is possible to modify and augment the composition of the visual structures over time. Common view-transformation techniques include location probes (e.g., tooltips), viewpoint controls (e.g., zoom, pan, and clip) and distortion (e.g., Fish-eye views [Fur86])

The information visualization system described in this thesis may also be simplified to this reference model.

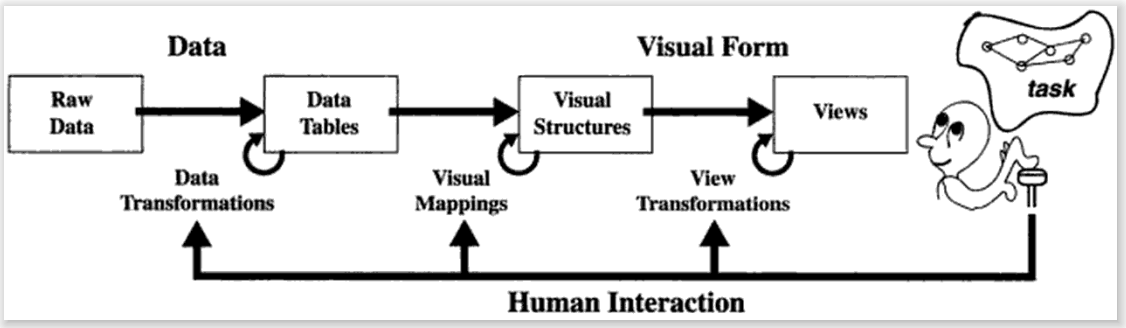

Figure 2.1: The reference model for visualization describes mappings from the raw data to a visual form to be analyzed by the human observer, who is able to interact with the mappings [CMS99].

2.1.4 Multi-Dimensional Visualization Techniques

The TDTT [Shn96] already mentioned that it is necessary to adjust the visualization technique to the used data type, to be able to support the tasks of the user. As this thesis focuses on multi-dimensional data, this section introduces two traditional and commonly used multi-dimensional visualization techniques: The scatterplot matrix and the parallel coordinates.

There are far more visualization techniques for multi-dimensional data, that can be used for different applications [Mun14]. These include Glyphs [BKC+13], Star Plots [CCK+83], or Chernoff faces [Che73].

Scatterplot Matrix

A scatterplot matrix [C+85], lays out scatterplots for all pairs of data dimensions as a matrix. The plots of one column share a common x-axis and the plots of one row share a common y-axis. One task that this technique is able to support is the determination if multiple variables have correlations between each other. Fig. 2.2 shows a scatterplot matrix with five data dimensions. As the linear aligned visual data records between the dimensions indicate, the statistical correlation between *Gust Speed 01* and *Wind Speed 01* is significantly high (0.962 Pearson Correlation [Pea95]).

When comparing p variables the matrix shows exactly $p(p-1)/2$ projections of the data. The quadratic dependency suggests that this visualization does not scale well for many variables, as for every new variable significantly more visual space is needed.

Parallel Coordinates

Another technique to visualize and analyze multi-dimensional data are parallel coordinates [ID90]. The axes of the data dimension are drawn in parallel and a data record is

represented as a polyline intersecting the parallel axes at the position of the coordinate of the corresponding dimension.

Parallel polylines between two parallel axes indicate a positive correlation between these two dimensions. If the polylines cross randomly, it can be assumed that no correlation exists. If the polylines cross intersect, the correlation is negative [Ins97]. Fig. 2.3 shows parallel coordinates with the same dimensions as Fig. 2.2. One can observe the parallel lines between *Wind Speed 01* and *Gust Speed 01*, whereby a positive correlation can be assumed.

When adding another variable, only one more axis is added to the visualization. Therefore, in contrast to the scatterplot matrix, the used visual space scales on a linear basis to the number of observed variables.

A visualization method that displays individual data values cannot scale for large amounts of data [FWR99]. To resolve this issue, hierarchical aggregation methods were developed for parallel coordinates. One hierarchical aggregation method is called hierarchical clustering. Only aggregated information is displayed in the polylines of the parallel coordinates [FWR99]. This reduces the number of visual elements. Through manual interaction, a user can then split clusters and further explore the aggregated data. This technique can be generalized into a framework and then used by multiple visualization types [YWR03].

Comparing multiple connected small visualizations is a common technique to observe several dimensions and gain insight into relations between them. This is described in detail in Section 2.4.

2.2 Visual Analytics

The TDTT described the overview of the data as the first task a user requires of an information visualization [Shn96]. However, traditional multi-dimensional data visualization techniques, like parallel coordinates (see Fig. 2.3) or scatterplot matrices (see Fig. 2.2), do not scale very well for high-dimensional data [KKE10, YWR02]. This leads to the need of analyzing the unfiltered and raw high-dimensional data before the interesting information can be presented to the analyst [KMSZ06]:

"Visual analytics combines automated analysis techniques with interactive visualizations for an effective understanding, reasoning and decision making on the basis of very large and complex data sets" [KKE10]. As Keim described, visual analytics is a multidisciplinary field, which combines multiple focus areas.

2.2.1 Focus Areas of the Visual Analytics Process

The **visual representations** and the **interaction** with them, exploit the advantages of the **human perception and cognition** to see, process and understand displayed information, as described in Section 2.1. The human perception, the monitor resolution, or the used visual metaphors meet their limits when analyzing large data sets. For example when comparing 100 data dimensions in a scatterplot matrix, 4950 scatterplots

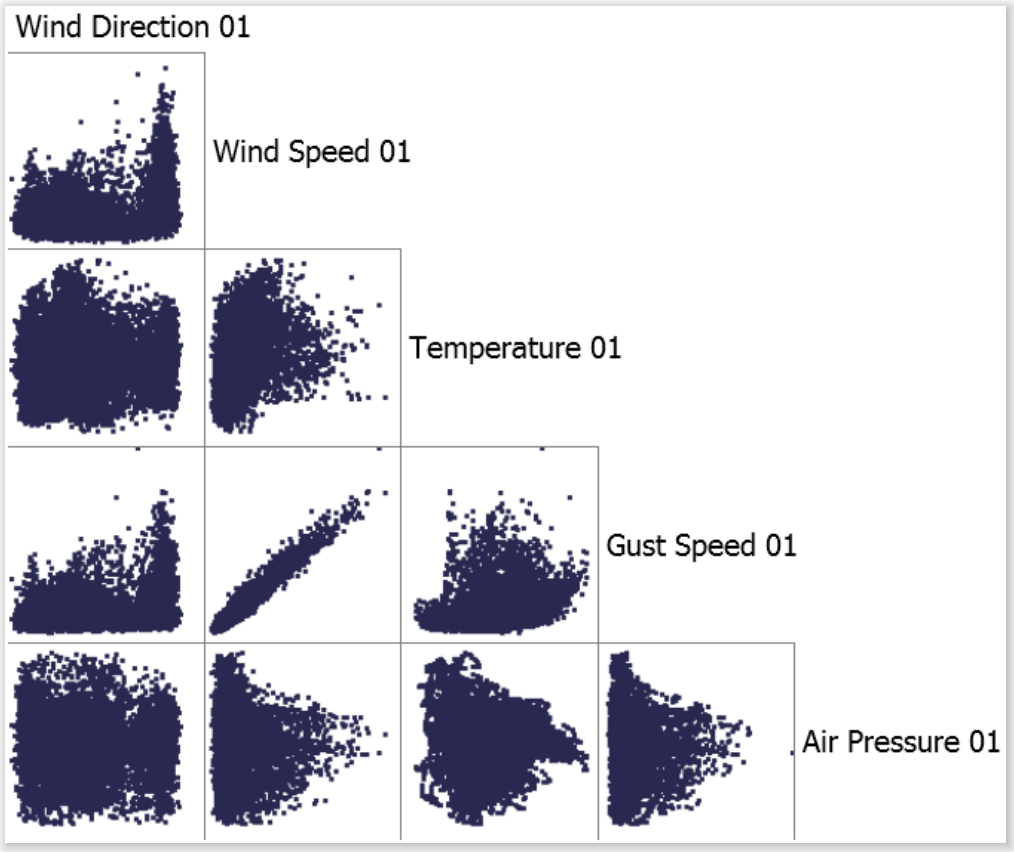

Figure 2.2: The scatterplot matrix is a multi dimension visualization technique to lay out pairs of data dimensions. The visualized data is from an example data set used in this thesis (see Section 1.1). The images are created with VISPLORE.

need to be observed, which is probably hard to display on a screen and hard for a human to analyze.

The **automated analysis techniques** enable the user to transform the data into a usable and reduced form. The computational powers of current hardware make these techniques applicable on huge data sets, but it is hard for humans to understand the process and the found solutions. For example, it is possible that data mining techniques only find a local optimum in a set of candidate solutions.

Fig. 2.4 emphasizes that the user is not a passive element in the decision-making and -exploration process of visual analytics, but rather the connection of the multiple fields to get deeper insights into huge data sets. Interactive visual analytics is an effective way to understand and process the data, by combining the strengths of these fields [TC06, Tho05, KKE10]. After the visual analytics process it may be necessary to support the user with techniques to **produce**, **present** and **disseminate** the gained analytical results.

The design study done in this work shows this interdisciplinarity, where the challenges

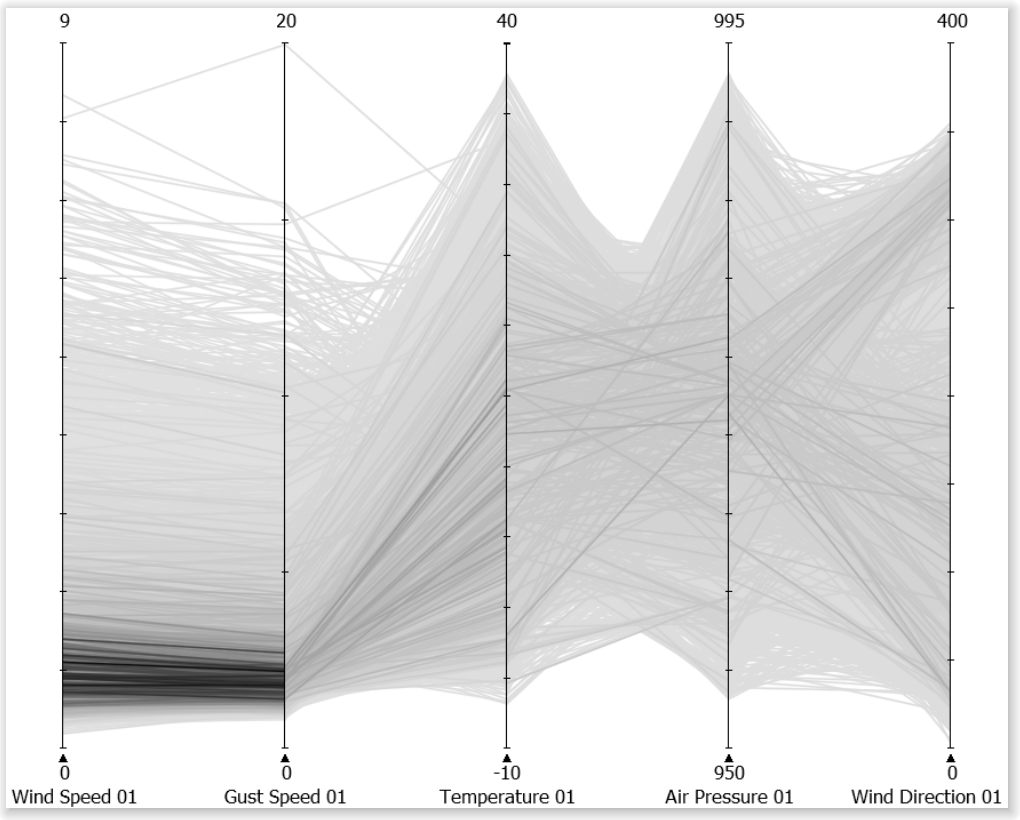

Figure 2.3: The parallel coordinates is a multi dimension visualization technique. The visualized data is from an example data set used in this thesis (see Section 1.1). The parallel lines between the data dimensions wind speed and gust speed indicate a high positive correlation. The images are created with Visplore.

of high-dimensional data, the tasks and goals of the users, and the design of the system are addressed. To analyze and understand these tasks and goals, the tasks and goals of visual analytics are introduces. These are based on the tasks and goals of information visualization Sections 2.1.1 and 2.1.2.

2.2.2 Tasks and Goals of Visual Analytics

> *Analyze first, show important, zoom, filter and analyze further, details on demand* [Kei05]

The visual information seeking mantra suggests visualizing an overview first, but the unfiltered and unprocessed data in high dimensional data sets make it hard or impossible to create an overview at the start. Keim adapted this mantra and formulates the visual analytics mantra. The main difference is the interaction with automatic analysis methods

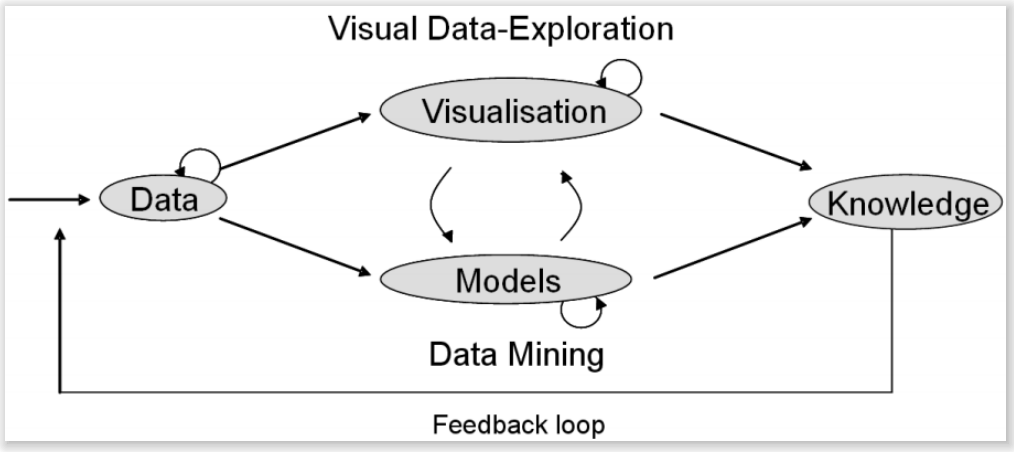

Figure 2.4: The feedback loop of visual data-exploration in visual analytics shows that the integration of automatic data analysis methods to the visual data exploration process supports the interactive decision making [KAF+08].

while visualizing the data. Uninteresting or falsified results can be detected early in order to obtain a better and more trustworthy end result.

To understand the problems of the domain and to better address them, the tasks and goals of visual analytics have been defined. This procedure is similar to the one already used in the previous chapter for information visualization (Section 2.1).

Tasks

Thomas [TC06] has realized that the decisions of the users in the visual analytics reasoning process are often made under extreme time pressure . He identified tasks where the user needs to make decisions and needs support [Tho05]:

- Understanding situations: Quickly understand past and present situations, as well as the trends and events that have led to the current conditions.
- Future scenarios: Identification of possible alternative futures and their warning signs.
- Monitoring: Monitoring of current events for the occurrence of warning signs and unexpected events.
- Indicators: Identifying indicators of the intent of an action or an individual.

In a specific scenario or a domain-specific application, these tasks can be formulated more precisely. For this work the tasks were identified for a domain-specific problem from the energy domain (see Chapter 3).

Goals

Based on the identified tasks, Keim formulated the goals of visual analytics. One of the most often cited goals of visual analytics is the creation of tools and techniques to enable people to "Detect the expected and discover the unexpected" [KKE10]. To be able to detect something, the user has to ask a question. An example for a simple question from the energy domain would be: *What is the current demand for energy in Vienna?* Different features in the underlying data may be detected or discovered:

- Common themes or patterns
- Trends
- Anomalies
- Unexpected relationships

Commonly, the asked questions are not as simple, as previously identified by the user tasks. The question may not only be fact based, but also needs the judgment of the user. One example of such question from the energy domain may be: *What is the gas consumption five years from now? How much should I invest in renewable energy sources?* These questions are often called "Wicked"-Problems [RW73] as they do not have an optimal solution. Their answers range from good to bad, instead being only true or false. Additionally, problems are unique, which makes it hard to learn from previous answers. This makes it necessary to include all existing data sources in the decision-making process, which makes the data massive, dynamic, ambiguous, and often conflicting [KKE10].

After the initial hypotheses are tested using the available data, it may be necessary to develop alternative hypotheses or explanations. These hypotheses should lead to a timely, defensible, and understandable assessment, which then needs to be communicated for action.

In this work the goals are identified based on determined tasks to guide the design process of the application. They are motivated by the goals of visual analytics.

2.3 Information Reduction

The scalability of a visualization technique can be measured by certain terms. Eick and Krall [EK02] formulated six factors of visual scalability: Human perception, monitor resolution, visual metaphors, interactivity, data structures, algorithms and computational infrastructure. Human perception is a factor that is hard to change, but it is possible to learn from its powers and weaknesses, to use them wisely in the visualization. Technical factors, as the monitor resolution and the computational infrastructure, are also hard to change in an information visualization design process. Alternative methods like a single device, a personal computer, a smartphone, or a tablet, can be used. Concepts for overcoming limited monitor resolutions exist, which can be considered when designing an application. Examples are the CAVE, a surround-screen projection-based virtual

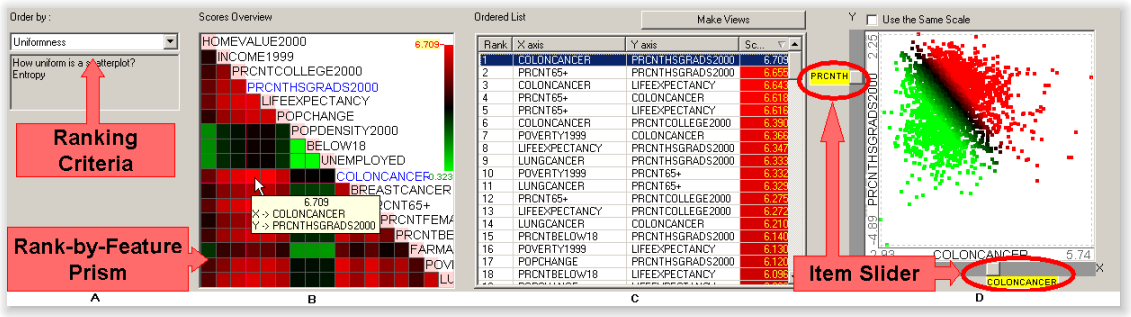

Figure 2.5: Screenshot of the Rank By Feature Framework by Seo and Shneiderman [SS05].

reality room [CNSD93], or head mounted virtual reality devices [Sut68]. To overcome the limited computational infrastructure, the utilization of multiple devices as multi node rendering for computationally highly expensive visualization techniques [GEM+13] or cluster, grid, and cloud, computation may be considered [AFG+09].

The limiting factors concerning the scalability of algorithms and data structures are mostly the number of data dimension and the number of data records. These two factors also limit the possible visual metaphors. To ensure the scalability of a visualization technique with high dimensional data, it is necessary to reduce information [Pir11].

2.3.1 Dimension Reduction

Principle Component Analysis (PCA), Multidimensional Scaling (MDS), and Self Organizing Maps (SOM) are common techniques used to reduce dimensions in data visualizations. They transform the initial data space into a smaller subspace, whereby unimportant information is omitted.

PCA: The initial data dimensions are linearly transformed to new variables, called principal components. All principal components are uncorrelated amongst themselves and the variance of the transformed data records is maximized [Jol02]. The number of formed principal components corresponds to the number of the initial data dimensions. By ranking them by their variance and selecting only the first components, the number of data dimensions can be reduced, whereby the original information of the data is best preserved.

MDS: The goal of MDS is the spatial arrangement of the input data dimensions in a low-dimensional space in a way that the distances after the projection are maintained as closely as possible to the inconsistencies or similarities in the initial data dimensions [Mea92]. To be able to interpret the resulting low-dimensional space, the chosen space is mostly two or three dimensional.

SOM: This is a vector quantization technique often used to visualize a two-dimensional representation of high dimensional data. The mapping from the initial space to

the reduced two dimensional space is done by arranging prototype vectors (with the same number of dimensions as the input space) on a two-dimensional grid, whereby a weight is assigned to each prototype vector. With a competitive learning algorithm the prototype vectors are placed with minimal distance to the initial data dimensions [Koh90, Pen05].

The drawback of these methods is that they produce a subspace that has no intuitive meaning to the data analyst [Fle01]. A data analyst knows in which space a single measurement should be. For example the Production Values of a photovoltaic plant are positive for a specific time period. The analyst has a concrete mental model how the time series of the PV should run. If this space has been transformed, the mental model of the data analyst will no longer fit to the data and the expert knowledge of the analyst will no longer be applicable.

Nonetheless, the number of dimensions needs to be reduced. A reduction method that does not transform the original data dimensions is a sampling of the dimensions by certain criteria to create a meaningful subspace.

A technique that supports the generation of meaningful subspaces is called Visual Hierarchical Dimension Reduction (VHDR) [YPWR03]. It uses a similarity measure to hierarchically cluster the dimensions and allows the user of this framework to interactively explore and modify the created hierarchy. From this hierarchy clusters, a meaningful subset is selected. Representative dimensions of these selected clusters are then visualized as visual representations. The drawback of this method is that not all dimensions are used for the encoding of the visual representations. User interaction is required to preselect important displayed dimensions. This may become a difficult task, especially for high dimensional data and without any a-priori knowledge or dedicated support.

To guide the user to select a meaningful subset, the user may rank the dimensions by a certain feature. The Rank By Feature Framework by Seo and Shneiderman [SS05] ranks small preview visualizations of one dimension or two dimensions by statistical properties, which give a good initial overview of all dimensions. The dimensions that match best for the chosen criteria are displayed first. This approach scales well with the number of data records, but has limitations regarding the number of dimensions. Especially for the comparison of dimension pairs (e.g., through a scatterplot matrix), the number of simultaneously displayed pairs has a quadratic growth. But also in the case of one-dimensional statistics, the upper limit of displayed visual representations that can be handled reasonably is a few hundred [PBH08]. Fig. 2.5 shows a screenshot of the RBFF for a two-dimensional comparison.

2.3.2 Record Reduction

An orthogonal approach to maintain the scalability of an application is the reduction of data records. Commonly used techniques are the removal and the aggregation of entries.

To reduce the number of displayed visual entries it is one option to just leave some out. There are two different options to choose the data records to leave out. One option is to randomly choose the preserved records, which is called sampling. Sampling reduces

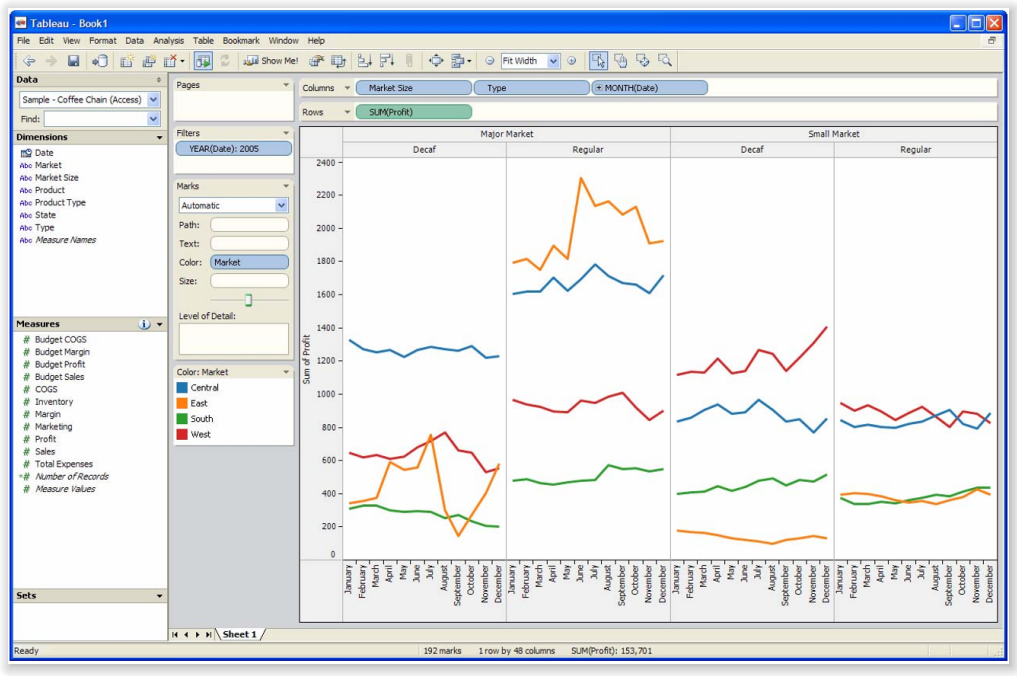

Figure 2.6: Screenshot of the visual analytics software Tableau. The *Profit* is aggregated by four categories: market size, product type, month and market region [MHS07].

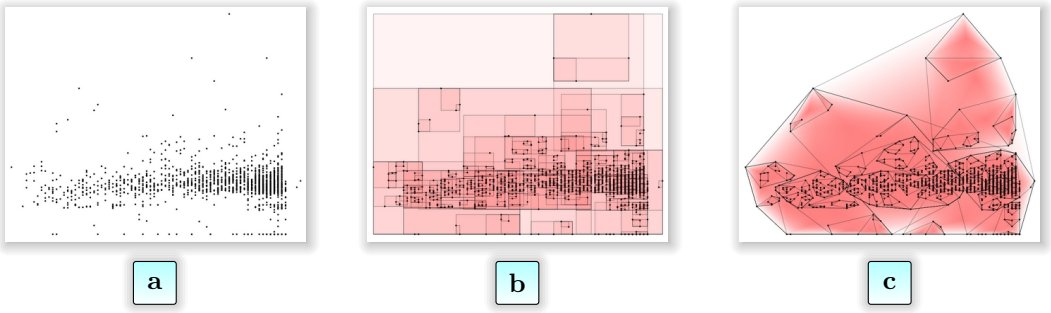

Figure 2.7: Hierarchical visual aggregation shown for a scatterplot (a). Hierarchical clusters may be visualized with their bounding boxes (b) or their convex hulls (c) [FP02].

clutter and improves the visual scalability and the scalability of data structures and algorithms at the same time, while preserving trends and correlations [DE02, ED07]. In contrast, it is not guaranteed that the remaining data records maintain all information the user needs (e.g., outliers). The other option is to reduce data records according to specific conditions, which is called filtering. Dynamic filtering is a frequent step in the visual information seeking mantra and found its way into different visualization designs [TSDS96, WK06, STH02, LSS09]

The aggregation approach combines data records to form a reduced number of visual

entries. In contrast to data removal, all data records are contributing to the final result. Three main approaches of aggregation are used in information visualizations:

- *Pivotization* combines the quantitative data of one column of a data table by the categories of other columns.
- In *binning*, the target quantities of the data dimensions are divided in ascending order into intervals — so-called bins. A commonly used technique to visualize the created frequency distribution is the histogram for a single data dimension.
- All other possibilities to combine data, fall into the class of *data abstraction*. Examples include cluster analysis [TB66], graphical statistical summaries such as box plots [Yu77], and visualizations of models such as regression lines and curves [DS14].

For the creation of a pivot table the categories of a categorical data dimension are used to reorganize and summarize the quantitative data dimensions of the data table. The output data may be represented in a condensed, summarized form due to the aggregation used in the data fields. This concept has been formalized and found its way into many business applications. They create a hierarchy by splitting the created aggregates from an categorical data dimension by another dimension. This drill down from one overview visualization was used in On-Line Analytical Processing (OLAP) [CCS93]. Stole and Hanrahan developed the system Tableau [MHS07] (former Polaris [STH02]). This thesis applies an algebra [Han06] for the partitioning of data dimensions into subsets to create the pivot table. Additionally, specifying the arrangement of the partitioning and creating visual representations of the aggregations via drag and drop, led to commercial success. Fig. 2.6 shows the interface of Tableau. The quantitative data dimension *Profit* is aggregated into a pivot table by building the sum of the categorical combinations of the data dimensions *Market Size*, *Type*, *Month*, and *Market*. The first three data dimensions partition the data on the y-axis and the *Market* is shown as a colored lined. This concept of defining the pivot table and partitioning and aggregating the quantitative data is used in this thesis to maintain scalability (see Chapter 4).

Elmqvist and Fekete presented a model for implementing hierarchically aggregated visualizations [EF10]. They propose that for many overlapping visual elements, as seen in a scatterplot visualization in Fig. 2.7a, it may be more scalable to aggregate them. Axis-aligned bounding-boxes (Fig. 2.7b) can be used to combine nearby points into clusters. These boxes give a good abstraction of the minimum and maximum values of a cluster. Fig. 2.7c shows the use of a more accurate visual representation, i.e., the convex hull, for displaying the extents of a set of points. In addition to the extents, the central tendency of a cluster may be encoded with opacity. This is also shown in Fig. 2.7c where the center point is displayed fully opaque and the border of the convex hull is shown transparently. This concept of abstract by including statistics of data into a visualization is similar to box plots.

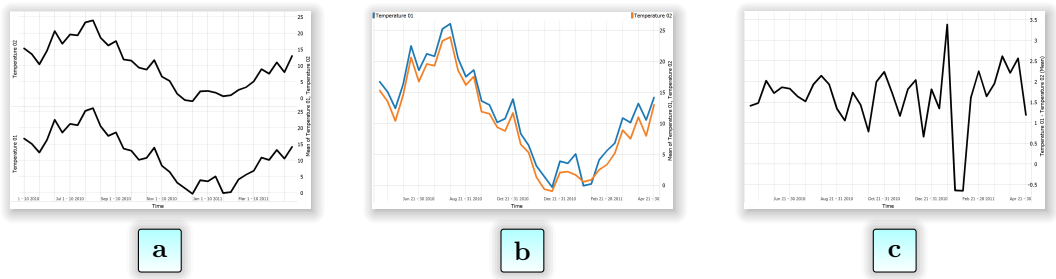

Figure 2.8: Design strategies for comparing visualizations can be subdivided into three categories: *Juxtaposition* (a), *Superposition* (b) and *Explicit Encoding* (c) [GAW+11]. The two compared *Temperature 01* and *Temperature 02* time series are from the used example data set (see Section 1.1) and are visualized with VISPLORE.

Elmqvist and Fekete not only implemented this technique for visualization techniques, but proposed general guidelines for creating hierarchically aggregated visualizations [EF10]. There should be an upper limit of rendered visual entities of a visualization. The visual appearance of aggregates should be simple and they should summarize the information of the underlying data. Aggregates are prone to interpretation errors, hence they should be distinguishable from data records and convey a meaning to the viewer. The design of the visual aggregates of this theses follows these guidelines. For example, the visual encoding of the different statistical properties use simple representations that helps the user to interpret the statistical type (dispersion, the central tendency and in addition to the work of Elmqvist and Fekete the frequency distribution).

There are multiple surveys that focus on high-dimensional data visualization [LMW+16, Mun14, AMST11]. These surveys extend the list of already mentioned techniques and design guidelines to handle high-dimensional data.

2.4 Comparative Visualization

The strength of the human in the visual analytics process is his ability to effectively compare objects through the visual system. This includes finding differences and similarities in visualizations.

Techniques for visual comparison for information visualization can be subdivided into three categories *Juxtaposition*, *Superposition* and, *Explicit Encoding* [GAW+11]. Fig. 2.8 illustrates these three visual designs on a simple time-series visualization of a used example dataset.

Juxtaposition: The visual objects in this design strategy are placed next to each other. This enables the visual objects to be placed within their own space. (See Fig. 2.8a) The number of displayed objects next to each other is limited, because the design relies on the viewers memory and eye-span [Tuf06]. Fig. 2.9a shows only 10 time series next to each other, where it is already hard for a human to spot differences between the visualizations.

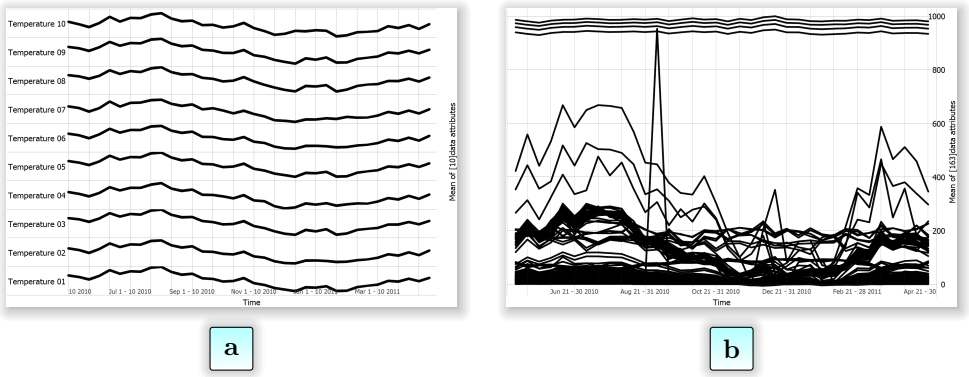

Figure 2.9: The limits of comparing high-dimensional data sets are the human memory for *Juxtaposition* (a) and the visual system for *Superposition* (b) [Fra13]. The compared time series are from the used example data set (see Section 1.1) and are visualized with Visplore.

Superposition: In this design strategy the visual objects are placed (overlaid) in the same coordinate space. Fig. 2.8b shows the same time series as in Fig. 2.8a overlaid in the same space. The visual system is used to compare the objects in this design. The number of displayed objects is limited by the mechanisms of perception [Fra13] and the screen space. Fig. 2.9b shows 163 time series in one coordinate space. Visual clutter makes it difficult to compare the time series.

Explicit Encoding: The last category is the visualization of computed relationships between the objects. Fig. 2.8c depicts the difference of the two previously shown time series *Temperature 1 — Temperature 2*. A drawback of this method is that the contexts of the original objects are not visualized and only the relationship is displayed.

In practice many visualization algorithms use a combination of these design strategies to enable a comparison [GAW+11]. This thesis focuses on a combination of the categories *Juxtaposition* and *Superposition* to support the user in comparing data.

CHAPTER 3

Tasks and Goals

This chapter introduces the used methodology (Section 3.1) to characterize tasks that are needed to overview high-dimensional data from the energy sector (Section 3.2). These tasks are used to derive the goals of the Hierarchical Data Overview (HDO) framework (Section 3.3).

3.1 Methodology

The nine-stage design study methodology framework [SMM12] is used to create a validated design of the HDO framework, as described in Section 1.4. In the first step of the core, phase the tasks of the user group were identified, as previously introduced for the tasks and goals for information visualization (Sections 2.1.1 and 2.1.2) and for visual analytics (Section 2.2.2). In the next step, based on these tasks, the goals of the framework were established.

To be able to characterize these tasks and on their basis the goals for an overview visualization of high-dimensional data, discussions with domain experts from the energy sector and visualization researchers from the VRVis were conducted.

3.2 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 spent looking at the data can be shortened by identifying the recurring tasks a user needs to fulfill. This thesis aims to focus on the several tasks described in the following. To reference them later in the work they are labeled with T1–4: T1 - finding

structures, T2 - rank by feature, T3 - assessing the purity of groups, T4 - exploration and tuning.

3.2.1 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. Hypotheses 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 illustrated by an example:

- Trends: Finding trends is a high-level task, which is not only relevant in the energy sector. Patterns that can be relevant for the analyst of a data set include the increase or decrease of the statistical properties of the data over time or recurring peaks, troughs, or plateaus of data records [Bre16].

  In the energy sector a trend is mostly associated with the behavior of time series. A transmission-system operator may be interested in the change of the power consumption over time of a certain area to estimate the needed power generation.

- Groups: Identifying structures like groups or clusters of data records enables the analyst to combine the underlying information, reduce the displayed visual elements and maintain the communicated information [Mun14]. Clusters are a set of data records that are more similar to each other than to the ones from other clusters. Finding groups, or clusters is relevant in the energy sector, when considering the similarity between time series. For example grouping the end-consumers by their energy consumption over time, helps the operator to summarize information and getting an overview.

- Modalities: One feature that can be identified, by looking at all data records of a data dimension, is the modality of the distribution of the data. The mode, also called modal value, is a location parameter in descriptive statistics. It is defined as the most common value in the data. If the distribution has more than one mode it is called multimodal. Energy operators may be of special interest to energy production or consumption time series with multimodal distributions. An example would be the identification of energy consumption peak loads in the morning and the evening of a work day, where the power consumption is significantly higher, than on the average supply level.

- Outliers: In every dataset outliers like anomalies, novelties, deviants, and surprises exist, which do not match the general trend [Mun14]. Finding them helps the analyst to direct the exploration of the data to find the cause of the outlier. Power grids need to be designed according to the highest peaks of power consumption. These peaks may be outliers to the general trend. Understanding the origin of peaks and preventing them can help the operators safe money.

3.2.2 T2 - Rank by Feature

The modality is not the only feature that can be computed from a data dimension. Other statistical properties like the median, the sum, or the number of outliers can be from interest to the user. As described by the RBFF, ranking all data dimensions by one feature, helps the user explore the most relevant data dimensions. An example from the energy sector would be the exploration of the biggest electrical loads in the electric grid. Moreover, transmission-system operators developed more advanced statistics to assess features of the measured sensors. Ranking gives them an overview of the features of the data and finding the interesting time series.

3.2.3 T3 - Assessing the Purity of Groups

By merging the data dimensions with the help of meta-information into groups, it is necessary to identify whether the grouping is sufficient for the user. If the similarity of the data dimensions inside a cluster or group is high, it can be assumed that the purity of this group is high. By characterizing this purity of these meta-information based groups, the user is able to make further decisions. One use case for assessing the purity of groups is the validation of the mental model of the meta-information based grouping to the actual data. For example photovoltaic panels may be grouped into plants. By assessing the purity of these groups, panels may be identified that do not match the others and need to be analyzed in more detail.

3.2.4 T4 - Exploration and Tuning

After the user was able to get an initial overview, further questions concerning the data may arise. These can be answered by exploring and tuning the created groups and receiving more detailed information. Two important exploration and tuning tasks are identified and are listed below and clarified by an example:

- Drill-Down: An analyst may be interested to locate and compare the identified structures found in task T1 in smaller groups of the data. In the energy sector this is especially relevant for a temporal refinement in time-dependent data (e.g., year, month, day) and local refinement in spatial data (e.g., country, city, power plant).

- Roll-Up: If a structure of interest is found (task T1) inside a group, it may be interesting, which impact this structure has on the parent group and how the structure relates to other groups. The roll-up is the counter-operation to a drill-down. Groups are summarized to a higher hierarchy level. For example, a local structure in the energy sector would be the power production of one power plant at a specific time interval. Summarizing these local structures from monthly to yearly intervals or from single plants to regional groups may help the system operators to get an overview.

3.2.5 Task sequences

The presented tasks are not performed in isolation. As the visual information seeking mantra (see Section 2.1.1) indicates, it is necessary to repeat the information visualization tasks until the goal of the user is reached.

In case of the energy sector, the data analyst starts with an overview (T1), filters for the relevant information (task T2), and explores and tunes further (task T3 and T4). Whereby not every task sequence occurs equally often [Bre16]. It depends on the previous knowledge of the user, which tasks require most of the time. If the question of the user is well formulated the time spent with task T1 will be relatively short in contrast to operators who analyze their data not very often and need to get an overview first.

3.3 Design Goals

Based on the task analysis three design goals have been established. These goals guided the design process of the HDO framework. Similar to the tasks they are labeled with G1–3: G1 - visual summaries of groups, G2 - flexible drill-down and roll-up, G3 - scalability

3.3.1 G1 - Visual Summaries of Groups

To support the user to find structures in the data (task 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 trend of data dimensions over time or over categories (task T3).

3.3.2 G2 - Flexible Drill-Down and Roll-Up

With respect to task T4 the overview visualizations need to be explored in depth. The large-scale visual overview summaries of the data can be seen as a starting point for a drill-down exploration into interesting parts of the data. The concept of drill-down and roll-up with respect to “any known structure of the feature space” [TLLH12] enables this fast change of the viewing granularity. A goal of the framework is to make it possible for the user to define structures in a way that makes it easier for the user to flexibly change the granularity of the data to be analyzed, according to which it can be subdivided or summarized.

3.3.3 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 the framework is to support simple monitoring and reporting tasks (T1, T2), but also to allow users to perform detailed exploration tasks (T3, T4). This implies a trade-off between

the simplicity and cognitive ease of the visualizations and preserving the distributions, modalities, and outliers of the underlying data.

CHAPTER 4

Data Model

This chapter describes the underlying data model of the Hierarchical Data Overview framework. The data must have a special structure so that the formulated goals can be achieved in the energy sector. A single table with the raw data is required Section 4.1. This consists of columns and rows, which can contain meta-information (Section 4.2).

In this thesis, a column is referred to as a data dimension and a row is referred to as a data record. Fig. 4.1a shows a data table, which is used as a guiding example for the data model. The guiding data example from Section 1.1 is used throughout the data model to create a context to the main application area.

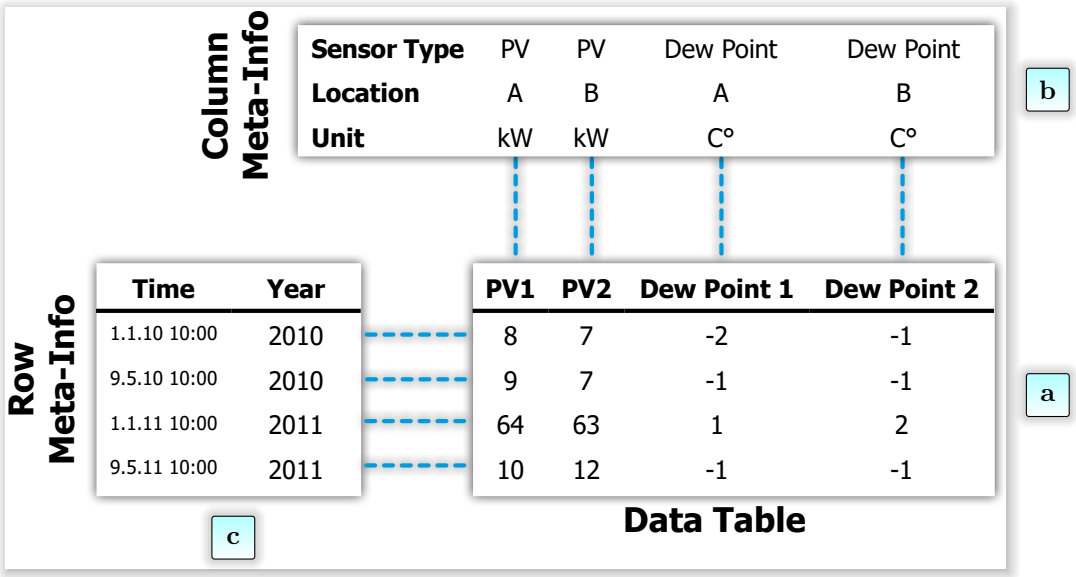

Figure 4.1: Data tables in the energy sector (a), contain meta-information on the columns (b) and on the rows (c)

4.1 Raw Data Table

The data table in Fig. 4.1a is a simplified example of how the raw data can be present in the data model. It contains values from the guiding data example. A column forms a time series and thus a data dimension. In this example the data dimensions are PV and temperature measurements from *Dew Point* sensors.

A row contains the measurements from a specific point in time. This can also be an aggregation over a time interval. A row forms a single data record. If the original data are not available in the form of this table, they must be transformed into this form by a data transformation.

4.2 Meta-Information of Raw Data

In addition to the raw data, there is often meta-information about the data available. Since the use of meta-information is a key element of this thesis, a classification of it is presented first. Then we will show how meta-information can be represented in the data model and how the meta-information depend on each other.

4.2.1 Classification of Meta-Information

Meta-information is the data about data. Meta-information can be organized into four main categories [FR14]. *Descriptive meta-information* is used for discovering and understanding the underlying resources (e.g., the sensors). *Administrative meta-information* is used for the processing of files. It can further be subdivided into technical, preservation, and rights meta-information (e.g., the acquisition time of the sensor data). *Structural meta-information* creates relationships between parts of the resources (e.g., the position of the sensor). *Markup languages* mix the previous types of meta-information within the content (e.g., SensorML by the Open Geospatial Consortium [BR07]).

Table 4.1 shows the four categories of meta-information with examples of the properties of the type and the primary task. This thesis focuses on descriptive and structural meta-information. Administrative meta-information and markup languages may be used to process the data to transform it into the needed data model of the HDO framework.

One important property of meta-information is that it is collected or extracted to fulfill a specific purpose and sorted into defined categories [Ril04]. Domain experts are familiar with these categories and know how to use them.

4.2.2 Meta-Information on Data Records

The data set can contain categorical or numerical meta-information on every data record. This meta-information may be present as a categorical data dimension in the data table.

One common categorical meta-information are time intervals of the data record (e.g., year, month, day, ...). They are created by partitioning the time value into time intervals. In Fig. 4.1c a categorical meta-information on data records is shown as the *Year*-column. But also other categories may be relevant meta-information in the energy sector. For

Table 4.1: Meta-information can be categorized by type [Ril04].

Type	**Example Properties**	**Primary Uses**
Descriptive meta-information	• Location • Name • Sensors • Keywords	• Discovery • Display • Interoperability
Administrative meta-information	• Technical meta-information: file type, file size, creation time • Preservation meta-information: checksum, preservation event • Rights meta-information: copyright status, license terms, rights holder	• Interoperability • Digital object management • Preservation
Structural meta-information	• Sequence • Place in hierarchy	• Navigation
Markup languages	• Paragraph • Heading • List	• Navigation • Interoperability

example a flag that indicates the status of the sensor or the current billing information, may be represented in the data and important information for analyzing the data further.

The data model of the HDO framework requires a time identification for every row of the data column. This may be a single time value or a time interval. The value is represented as a data record meta-information, which is shown in Fig. 4.1c as the *Time* column next to the *Year* column.

4.2.3 Meta-Information on Data Dimensions

Additionally to data records, meta-information can be assigned to the data dimensions themselves. This information cannot be stored inside the data table itself, but has to be stored in an external resource, because it may have a different data type than the data dimension. This meta-information is valid for all data record of the corresponding data dimension. This could be the information of the location, the measuring unit, or the type of the sensor creating the data records of the dimension.

Fig. 4.1b shows the meta-information for every column of the data table as an additional table. One rows in this table contains one meta-information of the same type for all data dimensions. This relationship is later used by the HDO framework to group the data dimensions.

4.2.4 Combination of Meta-Information

The numerical data dimensions can originate from a similar type (e.g., multiple power consumption sensors in kiloWatt (kW)). It is possible to plot the data on a common

scale. This enables the comparison of similar distributions or sequences, or the detection of outliers (task 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, ...). Typically it doesn't make sense for this data to share a common scale. In Fig. 4.1 the kW values of the PV sensors are on a different scale than the C° values of the *Dew Point* time series. Section 5.5.3 describes the combination aspect in more detail.

4.2.5 Dependencies of Meta-Information

Meta-information can be hierarchically structured. An example of a large hierarchy of descriptive meta-information is the Resource Description Framework (RDS) vocabulary [Sch]. It is used to mark up semantics within web pages and it is widespread on the web. An example for a schema hierarchy from the entity `LocalBusiness` is:

```
Thing > Organization > LocalBusiness
Thing > Place > LocalBusiness
```

In the energy sector, the meta-information of the data dimensions can have hierarchical dependencies. One common example is the location of a sensor. If it is assigned as meta-information, it may contain different levels of detail, which form the hierarchy. The highest level could be the state (e.g., Austria). A more detailed second level could be the city (e.g., Vienna). The lowest level could then be the actual address of the sensor.

Categorical meta-information on data records can also describe a level of detail. For example the measurement of *PVs* could contain status-codes as a categorical meta-information. A status code is assigned for every observed value. To be able to overview the status of the *PVs*, different levels of detail can be derived. These cover everything from from very basic information (Ok and Not-Ok) to detailed information (the concrete error code of the sensor).

CHAPTER 5

Visualization Design

This chapter describes the visualization method for the HDO framework. Its design is engineered by the defined goals from Section 3.3 for an application to data described in Chapter 4.

To fulfill the goals, the framework performs multiple steps. First, the data table is split into smaller parts. These parts are placed in a hierarchical relationship. Then statistical properties of every part are computed and in the end visualized in a tabular layout.

5.1 Subdividing Raw Data Tables into Data Chunks

The first of the framework is to split the data table into smaller parts. By utilizing the meta-information, described in the previous chapter, the data table can be subdivided by data dimensions and by data records into smaller blocks of data. In this thesis, these blocks are referred to as *data chunks*. Fig. 4.1, described in the previous chapter, is extended, to describe the subdivision into these data chunks. Fig. 5.1 illustrates the splitting from the not yet subdivided raw data table (see Fig. 5.1a) to smaller data chunks (see Fig. 5.1f). The data table is separated both at the rows and at the columns. These are orthogonal concepts and utilize different meta-information. For the subdivision of columns the meta-information on data dimensions is used (see Section 4.2.3). The meta-information on data records is used for splitting rows (see Section 4.2.2).

Fig. 5.1d shows the subdivision of the data table by the meta-information *Sensor* to the data chunks *PV* and *Dew Point*. Not shown but also possible subdivisions would be the combination of the data dimensions by the *Location* or the *Unit* meta-information, resulting in different data chunks.

A further possibility for the decomposition of the data table is a separation at data record level. Here all rows are combined to a data chunk which shares the desired meta-information. Fig. 5.1e shows the subdivision of the table according to the *Year* in which the time stamp (i.e. the row) was created.

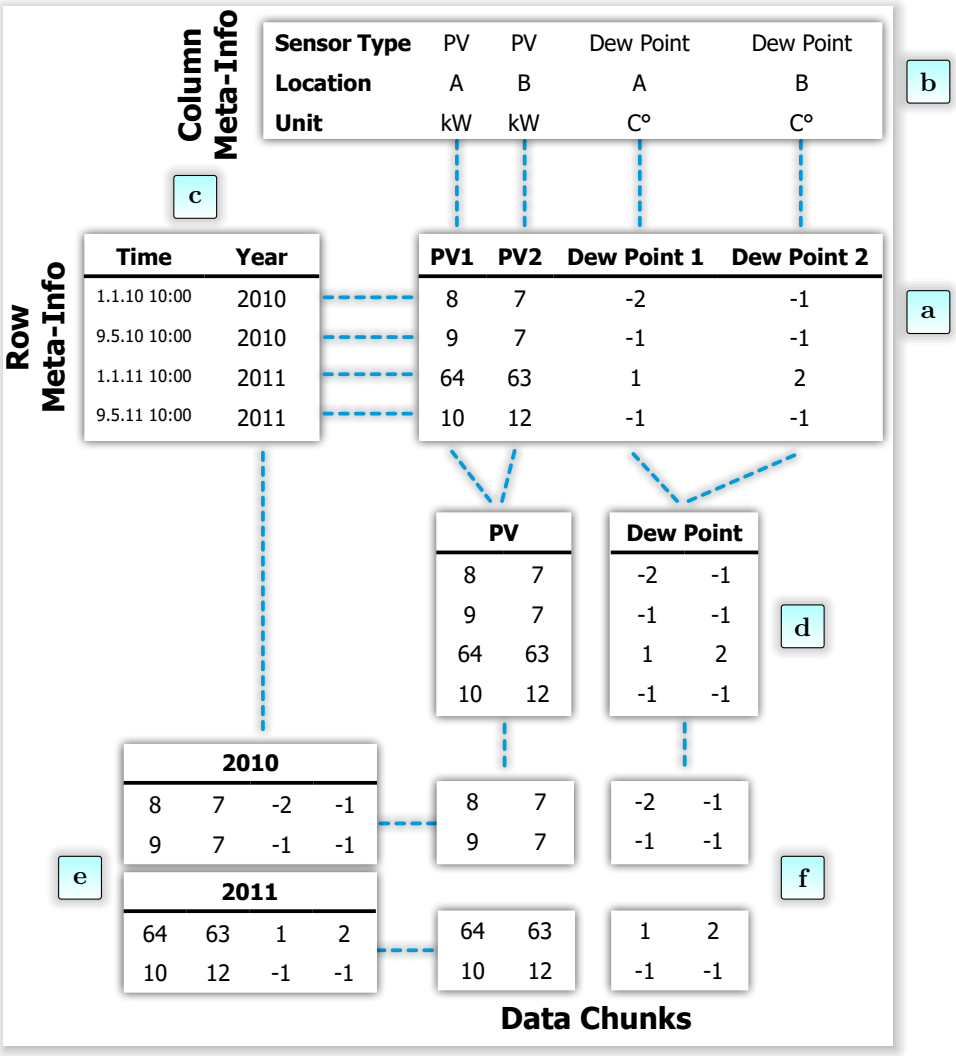

Figure 5.1: The subdivision of the data table (a) by utilizing the meta-information of data dimensions (b) and data records (c) to data chunks (d — f)

The data chunks shown in Fig. 5.1f are created by applying both previously described subdivisions. The order in which the two subdivisions are made is interchangeable, because the resulting data chunks are identical. Of course, it is possible to make further subdivisions, for example to add the *Location* as meta-information. From this, all data columns would be separated into two data chunks resulting in 8 blocks with 2 values.

5.2 Hierarchical Relationship of Data Chunks

The subdivision of the data table is performed for one meta-information at a time. One subdivision step, for example, from the table seen in Fig. 5.1a to the data chunks of

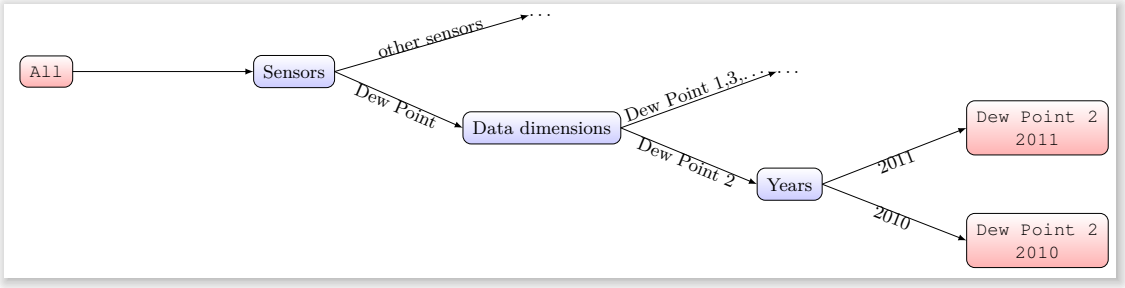

Figure 5.2: The drill-down of the hierarchy to the chunks with three hierarchy levels as a graph representation. The blue nodes indicate the hierarchy levels and the red nodes the data chunks. Figs. 5.3a to 5.3d show the same drill-down with the resulting visualization of the HDO framework.

sensors seen in Fig. 5.1d, is called a partitioning. The consecutive execution of the partitioning leads to the data chunks. As described above, the order of operations is not relevant. However, it becomes important for the user if an interactive drill-down or roll-up of the data chunks is to be used. Fig. 5.2 shows a graph, which shows the subdivision of the raw data table (All) to the data chunks (*Dew Point 2 2011 and Dew Point 2 2012*). The blue nodes in the graph represent three hierarchy levels of partitioning.

One hierarchy level refers to a meta-information and thereby defines the partitioning of an incoming data chunk into multiple data chunks. In this example, the data table is first partitioned into data chunks by their data dimension meta-information *Sensor*, as in the previous examples. This may result in multiple disjunct chunks and all of them are then further separated by their data dimensions. The last hierarchy level partitions the data chunks by the data record meta-information *Years*. The leafs of the hierarchy are actually just a concatenation of these three operations.

The hierarchical structure is a key concept to ensure the scalability (goal G3) of the HDO framework [EF10]. The HDO framework aims to visualize each of these nodes to the user. Besides the visualization of nodes and leaves, the partitioners must also be represented. In order to present this partitioning to a user in an understandable way and to give him the possibility to edit it, a tabular layout was used.

5.3 Hierarchical Tabular Layout

The overview visualization is designed using a hierarchical tabular layout (see Fig. 5.3). This design decision of using a table-oriented display enables an independent visual encoding of different aspects of the data [LGS+14]. Additional previous work showed that the users are familiar with this kind of layout [ASMP17].

The tabular layout consists of two orthogonal parts: rows (Section 5.4) and columns (Section 5.5). A *Row* defines a combination of data chunks and a *Column* is responsible

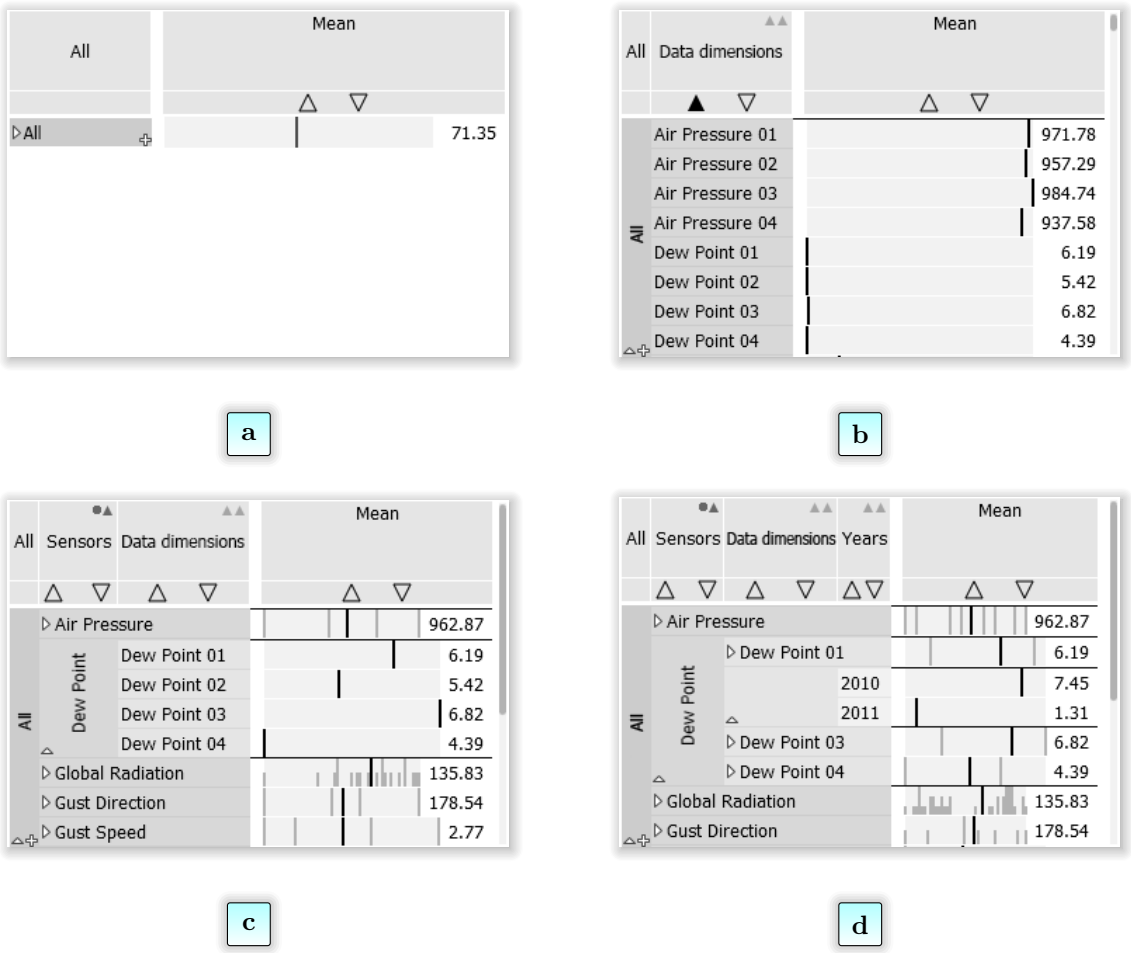

Figure 5.3: The HDO framework is visualized by a tabular layout. The left-hand side defines the rows of the visualization, which are defined by the hierarchy of the data model. The right-hand side visualizes the data chunks of the rows inside columns. The hierarchy can be collapsed (a), expanded by the data dimensions (b), combined by the sensor (c), or refined by a categorical attribute (d). The refinement increases the number of visualized data chunks.

for creating a visual summary of a descriptive quantitative feature of the data chunks (goal G1).

5.4 Table Rows

Visually the first part of the tabular layout of the framework corresponds to the hierarchy of the data chunks. The tabular layout offers the user an interactive definition of the hierarchy (goal G2) by defining one column per hierarchy level.

Fig. 5.2 shows the drill-down in a graph representation. The HDO framework visualizes the partitioning of the data table into a hierarchy of data chunks in a tabular representation. Fig. 5.3 shows this drill-down in four steps. Each step shows the tabular layout after the addition of another hierarchy level.

The headers of the left-hand part of the shown tables show these different hierarchy levels. The first column, marked with *All* in the example, is the root node of the hierarchy (see Fig. 5.3a). This column represents the whole raw data table and does not subdivide it. In this example the root node is the only data chunk and it is equal to the only leaf node of the hierarchy. The second column uses the data dimensions of the original table as a subdivision (see Fig. 5.3b). The next column partitions the table by the data dimension meta-information *Sensor* (see Fig. 5.3c). Fig. 5.3d shows the partitioning by the data record meta-information *Years* as the added fourth column.

One row of the table corresponds to a node of the created hierarchy. In Fig. 5.3b one row corresponds to a single data dimension and also to a single data chunk (for example *Dew Point 03*). In contrast, in Fig. 5.3c the first row shows all data dimensions with the same meta-information of the Sensor *Air Pressure*. That means it is not a leaf of the hierarchy but a node, which contains several (four) leafs. It is possible to drill down the hierarchy to show nodes and leafs side by side. Underneath the previously described row in the same Figure four rows are shown which do not show a node but again leafs.

Additional hierarchy levels refine the data table into data chunks (goal G2). A usual refinement is the partitioning of the data table by data dimensions. Fig. 5.3b shows the table with all assigned data dimensions, which displays a similar layout as compared to the RBFF. The 159 data dimensions and thereby also rows are assigned to the visualization. These are too many to display all of them on the limited screen area.

By combining data chunks, the number of rows can be reduced. In Fig. 5.3c the data dimensions are combined according to their sensor. The number of displayed rows decreases but the number of displayed data chunks increases. For example, the sensor *Air Pressure* contains four dimensions which are plotted in the mean column (see Section 5.5.1). In Fig. 5.3d the hierarchy level *Year* is assigned to the table. As previously described, this level partitions the data records of the data chunks by the categorical meta-information (see Fig. 4.1). As shown in the visualization of the dimension *Dew Point 01*, the number of chunks has increased.

The user is able to control the order of the hierarchy levels, add new levels or remove them (see Section 6.2). Additionally, the visible set of rows can be defined by collapsing and expanding the hierarchy nodes individually. Fig. 5.3d shows the expanded node *Dew*

Point of the hierarchy level *Sensors* with an additional refinement for the dimension *Dew Point 02*.

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

5.5 Table Columns

The orthogonal part to the rows of the tabular layout are the columns to the right hand side. They define which aspects of the data chunks the user wants to observe. This includes statistical properties and visual encoding. To maintain the scalability of the visual complexity (goal G3) of the visual representations of the data chunks the design decision for the visual encoding of a single entry is to select simple and commonly used visualizations. For example in Fig. 5.3 the mean of a data chunk is visualized as a line inside the cell of the column.

This section addresses three aspects of the visualization of data chunks: the visual aggregation, the partitioning, and the combination aspect.

5.5.1 Visual Aggregation of Data Chunks

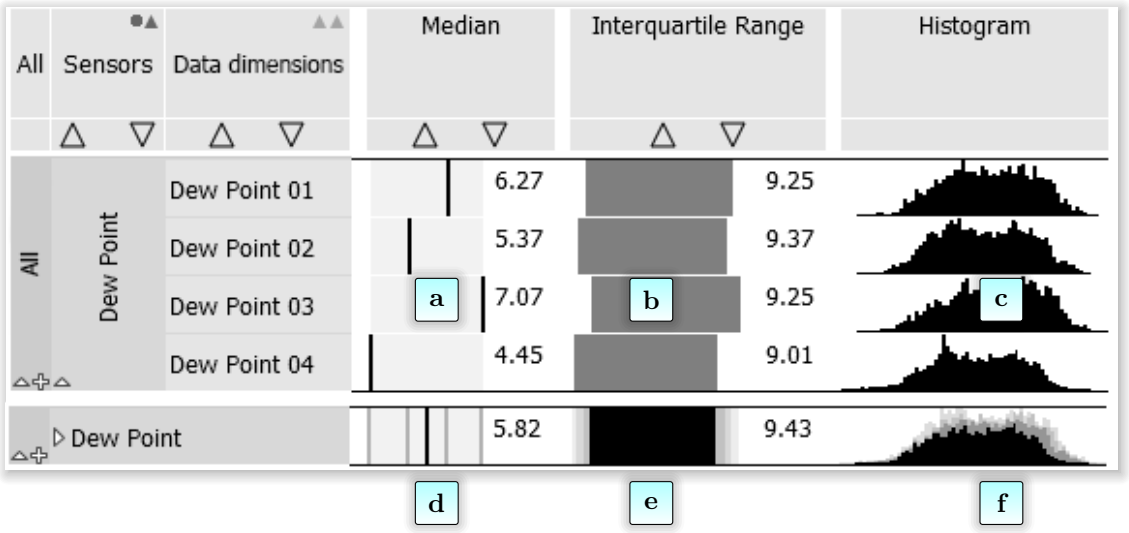

Figure 5.4: The data chunks are visually aggregated in the columns. A cell can visualize one leaf of the hierarchy (a, b, c), but is also able to visualize a node by combining the underlying leafs (d, e, f).

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 (goal G1) [Man07]. The system differentiates between three classes of descriptive statistics. Fig. 5.4 shows the visual encoding of these three

classes: Central Tendency (see Figs. 5.4a and 5.4d), Dispersion (see Figs. 5.4b and 5.4e) and Frequency Distribution (see Figs. 5.4c and 5.4f).

A simple way to visualize a statistic feature is by means of a textual representation. However, text does not encode the position, scattering, and distribution (goal 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 the visual aggregates can be compared intuitively with each other.

In the following paragraphs the visual representation of a data chunk within a cell concerning the three classes of descriptive statistics is introduced in detail. It is also discussed how several visualizations can be combined within a cell. Section 5.5.3 deals with more details of the combination aspect.

Central Tendency

A univariate statistic can describe a point on 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, quantiles). Like other univariate statistics, they can be used for ranking (task T2) but also for other tasks, like finding outliers (task T1). A textual representation of the aggregation, as shown in Fig. 5.4a on the right-hand side of the cell, enables the user to observe the precise numerical 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 of one data chunk with another one (task T1). If multiple data chunks have to be visualized within one cell, the same visual aggregate can be used. The locations of the underlying features are plotted as gray lines, and the combined statistic is shown as a black line (see Fig. 5.4d).

If more than one line is drawn at the same location, the visualization turns into a bar chart as shown in the *Mean* cell of the *Global Radiation* row of Fig. 5.3d. The bin with the most overlaying lines uses the full vertical space of the cell. All other lines start at the lower border of the cell and their height is, depending on the number of overlaying lines, proportionally smaller than the full height.

Dispersion

The second class of univariate statistics is the dispersion. It describes a positive range along an axis. Examples for this class of descriptive statistics are the standard deviation or the Inter-Quartile Range (IQR). Similar to the previous encoding, an area is positioned around a dependent central tendency (the median for the IQR or the mean for the standard deviation) inside the axis of the aggregation. One example visualization is an area with the width of the IQR around the median, which is shown in Fig. 5.4b.

The dispersion of multiple data chunks can also be visualized within a cell. The value that is displayed is usually the dispersion of the combination of all underlying data. In order to show that several chunks are visualized in this cell, the areas of the individual visual elements are drawn on top of each other. Fig. 5.4e shows this over-plotting. As one can see, this cell shows different gray scales. These are created by

histogram equalization [PAA+87]. Histogram equalization is a technique for adjusting image intensities to enhance contrast. The individual visualizations (the black ares) only have a single intensity value in its image which is a fully saturated black $I_{\max}$. Now the n visualizations are drawn adaptively on top of each other, but the intensity value I of the individual images are reduced by

$$I = \frac{I_{\max}}{n}.$$

The intensity values of the resulting visualization are now changed using the histogram equalization method, so that a maximum contrast between the areas is achieved.

Frequency Distribution

An example of a visual representation of the frequency distribution of data is a histogram. A textual representation is no longer suitable, because multiple values have to be encoded. To support details on demand (task T4), specific information of a bin value of a histogram can be displayed as a tool-tip information. In contrast to the previous classes, it is not easily possible to rank the data chunks by frequency distribution.

The binning of the histogram depends on all data chunks inside the axis (see Fig. 5.4c). If the frequency distribution of multiple data chunks has to be visualized within one cell, the histograms of the underlying data chunks are plotted over each other. As in the previous class, histogram equalization is used to show the combination. The darker a part of a bin the more histograms overlap in this position (as shown in Fig. 5.4f).

As identified in Chapter 3, users are also interested in finding structures over time (task T1). In order to observe data chunks over another variable such as time, columns are expanded so that they can be further partitioned.

5.5.2 Partitioning a Column

Another task a user may want to address is the trend 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 on local relationships of the statistic features (goal G1) [MP13].

Fig. 5.5 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 previously introduced three classes that were proposed in Section 5.5.1 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. The vertical position of a not partitioned column had no meaning. For a partitioned column the vertical position of the line (the height) is determined in the same way as the horizontal position for the central tendency of a

not partitioned column. Multiple lines are drawn inside one cell if multiple data chunks belong to one row (see Fig. 5.5a). This representation can be used to analyze the trend of the data, to find outliers, reoccurring patterns, or correlations (task T1).

Dispersion

Similar to the central tendency, lines are drawn as a visual encoding of the dispersion for a partitioned column. The width of the line inside a sub column encodes the dispersion of the data around the dependent first moment. Its upper and lower boundaries are defined by the value of the dispersion. These two boundaries are connected for all partitions and the area between them is filled (see Fig. 5.5b). This representation can be used to characterize the purity of the underlying data (task T3).

Frequency Distribution

The frequency distribution of all data chunks inside a sub-column is calculated. A visualization called Curve Density Estimates [LH11] is used to encode the distributions of all sub-columns (see Fig. 5.5c). Each sub-column is partitioned a second time and each bin is colored with the relative frequency of the data it contains. The darker a point is displayed, the more data points it contains. This representation can be used to analyze the modalities of the underlying data chunks and address the purity of the groups (tasks T1 and T3).

5.5.3 Combining Representations of Data Chunks

All shown examples have used the same scale for creating the visual aggregate. A precondition is, however, that not all data chunks have to share the same scale.

Scales of a Cell

A single linear scale maps a continuous, quantitative input domain to a continuous output range. The output range is defined depending on the used column. For columns without sub-columns the width and for partitioned columns, the height is used.

The input domain is set by the statistical properties of a column. All data chunks that share a scale belong to a set, called a scale group. The assignment to scale groups is the same for all columns, but each column has a different input domain for its scales. The domain is defined by the calculated statistical values of the data chunks. For example, the *Median* column in Fig. 5.4a uses all four Median values of the four data chunks (6.27, 5.37, 7.07, 4.45) and takes their minimum (4.45) and maximum (7.07) values to define the continuous, quantitative input domain. Other columns may use different strategies to define the domain.

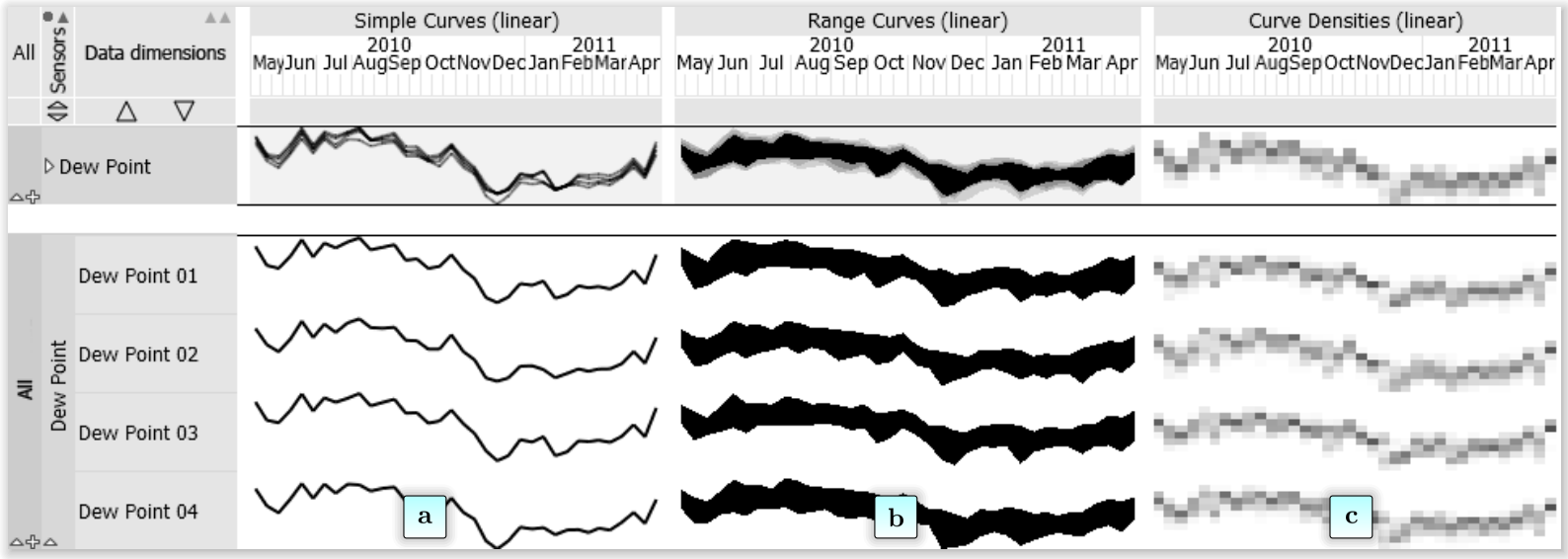

Figure 5.5: Central tendencies are shown as line graphs (a). Dispersions are shown as areas graphs (b). Frequency distributions are shown as heat maps (c).

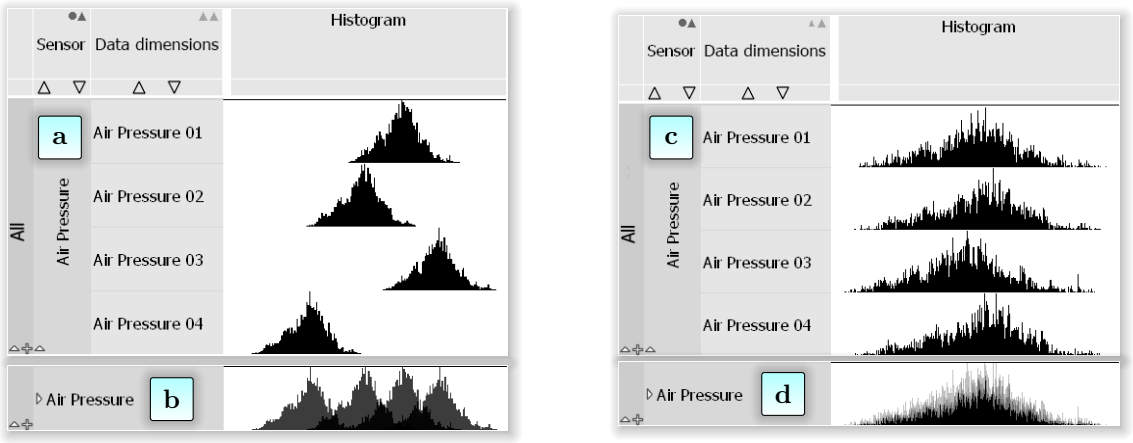

Figure 5.6: The scales of the visualized data chunks can be configured by the hierarchy levels. To compare the position the data chunks, nodes of the level are visualized on a common scale (▲▲) (a, b). To compare the shape of data chunks, every node receives an own scale (▲▲) (c, d).

Scale Affiliation of a Data Chunk

The created hierarchy levels, as described in Section 5.4, can have the property that they separate data into distinctive nodes, where the combination of the data chunks has no useful meaning. This means that the data separated by this level no longer share the same scale. Every chunk with the same meta-information now belongs to the same scale group.

In the visualization, this is indicated by displaying the icon ●▲ ("not combinable") in the header of the hierarchy level (see Fig. 5.6). An example of not combinable nodes is the partitioning of the data by the sensors of the underlying dimension, where it makes no sense to plot a temperature and a voltage value in the same coordinate frame (see Figs. 5.7a and 5.7b). If the hierarchy levels does not separate the data into different scale groups the icon ▲▲ ("common scale") is shown.

However, a level can also separate 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 ▲▲ ("no common scale") in the header (see Fig. 5.7d). The data chunks are still separated into different scale groups for each meta-information. However, data chunks with different scale groups that do not have a common scale, but are combinable (not set to ●▲), can be displayed in the same cell.

Displaying Data Chunks in a Single Cell

The visual representation of a data chunk is always drawable inside a cell if it is the only one. If multiple chunks have to be shown in the same cell, multiple cases need to be addressed:

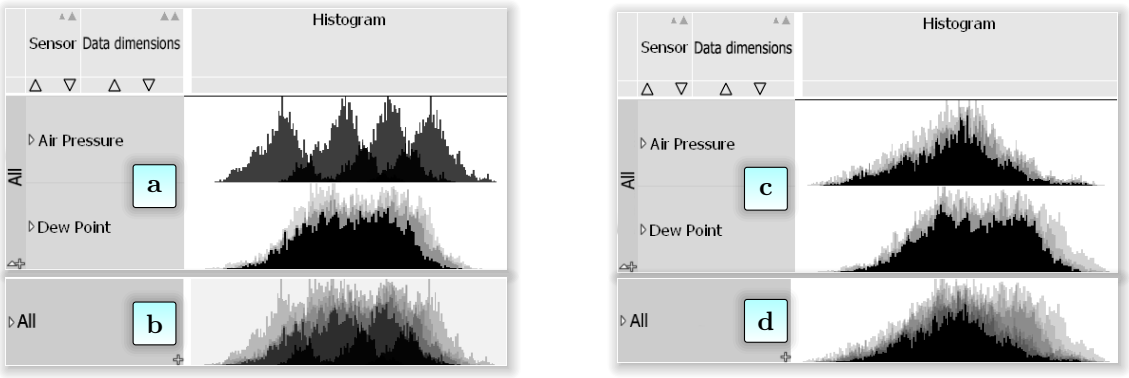

Figure 5.7: The rows a and c show the data chunks of two sensors. The rows b and d show the combined cells. The hierarchy level *Data dimensions* is set to ▲▲ on the left to compare the central tendencies and to ▲▲ on the right to compare the shape of the distributions.

Figure 5.8: The same data as in Fig. 5.7 is shown. If the nodes of hierarchy level are not combinable (●▲) no visualization can be plotted (a). If the level is set to ▲▲ the position can be observed (b).

If two levels of the hierarchy are set to not comparable (●▲), they cannot share a common axis and no visualization can be drawn (see Fig. 5.8a). The common idiom "*comparing apples and oranges*" states that a non suitable comparison would indicate a false analogy (see Fig. 5.7b). To avoid this the cell just displays "not combinable". In comparison, Fig. 5.8b shows when you display data chunks with different scales on one scale. The superimposed histograms show two deflections (task T3).

The already shown case is that all data chunks share the same axis and are combinable. All hierarchy levels that created the chunks have to be set to ▲▲. Hence only one axis exists and the input domain and the output range are the same for all statistical properties of the data. With this setting, it is possible to compare the position of the underlying data chunks. In Figs. 5.6a and 5.6b the hierarchy level *Data dimensions* is set to ▲▲. It is possible to observe that the distributions of the *Air Pressure* measurements are similar, but the central tendency differs. As opposed to this, it is not as easy to compare the shape of frequency distributions of the different data chunks (task T1).

To be able to compare the shape, the hierarchy level may be set to "no common scale" (▲▲). Each created node of the level defines its own scale. It is not possible to use the

same input domain for multiple data chunks with different scales, that are combinable (not set to ●▲) and displayed inside the same cell. The output range of all these scales is still the same (the dimensions of the cell), but the input domain of every scale may now be different. The goal of switching the scales is to compare the shape of the visual encodings. The visualizations of different scales are drawn independently and are then put on top of each other. Figs. 5.6c and 5.6d show that the shapes of the frequency distributions of the different data chunks can now be compared easily.

In Figs. 5.6a to 5.6d the hierarchy level *Sensor* is always set to ●▲. Also, only *data dimensions* from the same sensor are shown. In Fig. 5.7a four more data dimensions of the type *Dew Point* were added. One can see the histograms of their data chunks in juxtaposition to each other. It is important not to compare the position of these distributions, because they do not have a common scale. This becomes clear when the two cells are displayed in combination (Fig. 5.7b).

However, if the user changes the combination type of the level *Sensor* from ●▲ (see Fig. 5.8a) to ▲▲ (see Fig. 5.7b), he will see the *Air Pressure* and *Dew Point* representations of Fig. 5.7a superimposed. This combination is hard to interpret and illustrates why "*oranges should not be compared to apples*". The user must know the meta-informations of his data and configure the hierarchy levels accordingly.

For completeness, Fig. 5.8b shows the same data on the same scale. Here one can see that the central tendencies of the two sensors are very different.

CHAPTER 6

Addressing the Explorative Overview Tasks

This chapter describes the used direct interaction techniques to configure the HDO framework and address the explorative overview tasks.

6.1 Direct Interaction

Direct interaction is the involvement of "a dialog with feedback and control throughout performance of the task" [Dix09]. Direct interaction uses visual elements to guide the user to the desired action. The action is achieved by pointing on a visual representation. Furthermore, this action has to be rapid, incremental, and reversible [AS94]. Well known examples of direct interaction controls are user interface controls (e.g., buttons) inside a user interface design, which are constantly visible in the screen space.

Another class of direct interaction controls are only shown on demand. Fig. 6.1 shows a well known on-demand interaction including a popup menu (Fig. 6.1b) and a popup window (Fig. 6.1c) [RBH+12]. The HDO framework focuses on interaction techniques that appear on-demand. The advantages of making this design decision are [Gal07]:

Saving screen space On demand controls enable a more compact design of visualizations, while maintaining the same functionality, by leaving out screen space filling controls that encode redundant information.

Within the context of the task Since the control elements appear where they are able to fulfill an action, the user can reach his goal faster. For example popup menus as shown in Fig. 6.1b are often also available from other control elements like complex menu structures.

Reduce redundant information Direct interaction controls, that are always visible in the interface design, may encode redundant information. An example are input

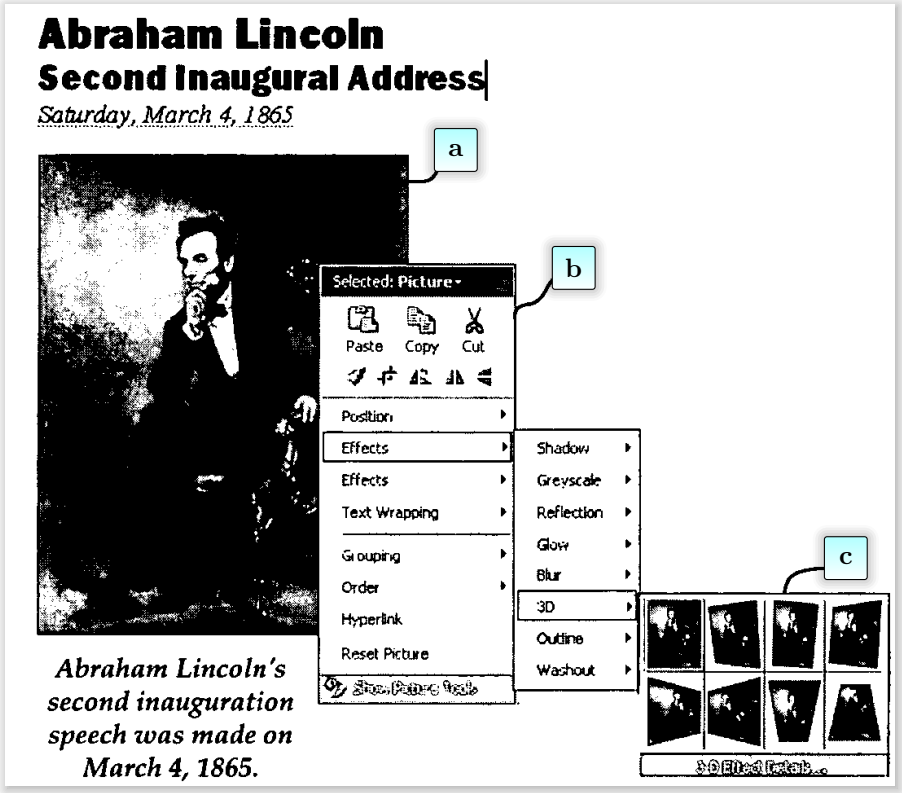

Figure 6.1: The figure (a) inside the text editing program Microsoft Word can be edited on demand by opening a context specific popup menu (b) and a popup window (c) [RBH+12]

fields that show values, which are also visible in the interface. These redundancy is not desired when exporting and publishing the interface design (e.g., as a screen capture). The use of on-demand controls only encodes this redundancy when displayed, which solves this problem [Few09].

Disadvantages of on demand direct interaction controls are on the one hand the shallow learning curve, which is caused by the fact that the user has to remember the location and the trigger action of the controls. On the other hand it is possible that the user triggers them by accident or popup controls occlude important parts of the screen working area [Pfa15].

To be able to perform the identified tasks of the users (see Chapter 3), the HDO framework uses direct interaction controls. To reduce the disadvantages of direct interaction controls, it tries to keep the access to the popups consistent. Whenever a user hovers the mouse over the title area of a column, she or he gets all relevant configuration options for that column (see Section 6.2). To configure individual cells of a column more precisely or to learn details, the user receives control elements when hovering over a cell.

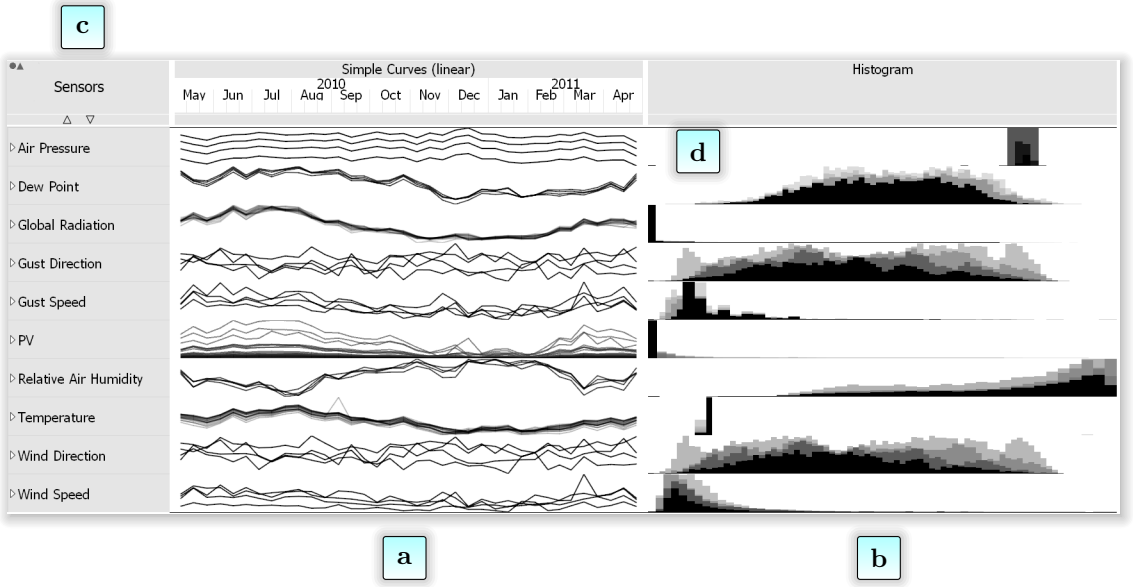

Figure 6.2: The HDO framework visualizes the time curves (a) and the distributions (b) of all 163 data dimensions combined by their sensors (c). (d) shows an outlier in the *Air Pressure* data

6.2 Explorative Overview Tasks

As a precondition, it is assumed that the domain expert knows what data types, sensors, and meta-information one can expect from the data. Typically a user does not start his exploration task with no initial hierarchy, as shown in Fig. 5.3a. A pre-defined set of hierarchy levels that are relevant in the domain is assigned. Fig. 6.2 visualizes all 163 data dimensions of the sensor data set from the guiding example (see Section 1.1). The initial hierarchy is defined by the sensors, and the underlying data dimensions, which are collapsed so that the user is able to get an overview of the data.

To address the goal G1, the user is able to find structures by looking at the visual summaries (task T1). Example structures that can be observed in Fig. 6.2 are: the yearly trend of meteorological quantities like the *Temperature* and the related *PV*; the modality of the *Gust Direction*, which is different for every sensor; the outliers of the data for example the null value in the *Air Pressure* distribution and the purity of some groups like the *Global Radiation* (task T3).

To support the user in his different tasks with the visual analysis of the data, the following configuration options are offered for the HDO framework.

- Adding and Removing Columns (Section 6.2.1)
- Drill-down and roll-up concerning Rows (Section 6.2.2)
- Restricting the shown range (Section 6.2.3)

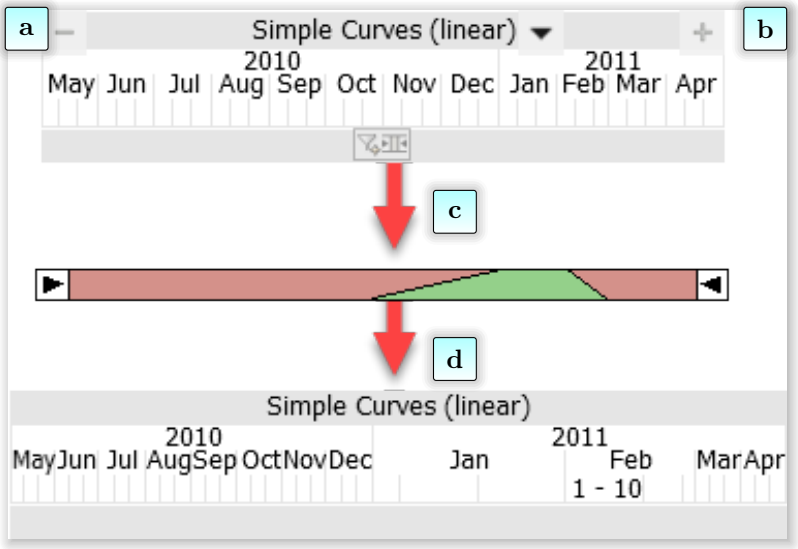

Figure 6.3: Interaction controls are displayed on demand in the column header. It is possible to remove (a) or add (b) columns and open controls to configure the column (c). One configuration concerns the changing of the mapping of the displayed interval (d).

- Reordering the tabular layout (Section 6.2.4)

6.2.1 Adding and Removing Columns

A column in the tabular layout represents either a hierarchy level or visualizes statistical properties of the rows of the table (see Section 5.5). The more columns being used, the more accurately a user can go into individual details in the data. For example in Fig. 5.4 the user is able to analyze the central tendency (median values), the dispersion (IQR), and the frequency distribution (histograms) of the data chunks. This increases the visual complexity, but also enables the user to find structures, like the modality of the data chunks or outliers. Furthermore, the displayable space is limited by the screen. Therefore it must be possible to add and remove columns on demand.

The direct interaction controls to add or remove the columns are shown on demand at the header of a column. Fig. 6.3 depicts the displayed on demand controls. Figs. 6.3a and 6.3b are the buttons to add and remove a column, indicated with a “plus” and “minus”-icon.

Another already introduced concept is the subdivision of columns into sub-columns as described in Section 5.5.2 and shown in Fig. 5.5. This enables the user to analyze the trend and the modality of the data chunks. This subdivision of a column must also be changeable by the user, which is described in Section 6.2.3.

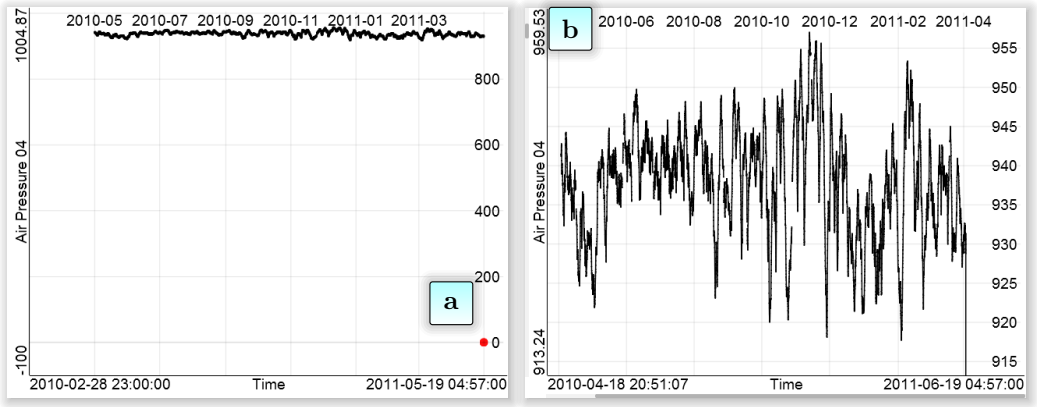

Figure 6.4: In some cases it is not suitable to display all data records in a visualization. (a) shows a single outlier in the data, which changes the domain range of the data. By restricting the shown range to a smaller interval (b) details can be observed.

6.2.2 Drill-Down and Roll-Up

To address goal G2, the framework supports the drill-down and roll-up of the hierarchy nodes as described in Section 5.4 and shown in Fig. 5.3. Whilst exploring the visualizations, a user may observe that the variance in some displayed nodes is very high, and wants to drill-down the node to see if the purity of the underlying nodes in the next hierarchy level increases (task T3). This is achieved by clicking on the arrow inside of the hierarchy node. Fig. 5.3d shows this drill-down of multiple hierarchy nodes to a detailed representation.

The direction of an arrow indicates to the user if the node is collapsed or expanded. This is the reason why it is not only displayed on demand, but permanently in the interface.

Additionally the user is able to refine the hierarchy further, i.e., by adding more hierarchy levels, by clicking on the plus sign in the lower left corner (Fig. 6.2). The further refining of the hierarchy may increase the purity of the newly partitioned data chunks.

6.2.3 Restricting the Shown Range

To address task T4 further, every column can be configured individually. Adjusting the scaling of a continuous axis to a certain interval is a common interaction technique to explore the data [YaKS07].

As the visual information seeking mantra proposes, it is necessary to show the user an overview of the available data. However, after the overview is presented, the user may be interested in a small part of the axis. Fig. 6.4 illustrates the restriction of the shown range of a time series. Due to an outlier (Fig. 6.4a, dot in red), the time series is visualized with visual clutter and it is hard to identify details. By reducing the shown range (Fig. 6.4b), it is possible to explore the data further and identify hidden structures (task T1).

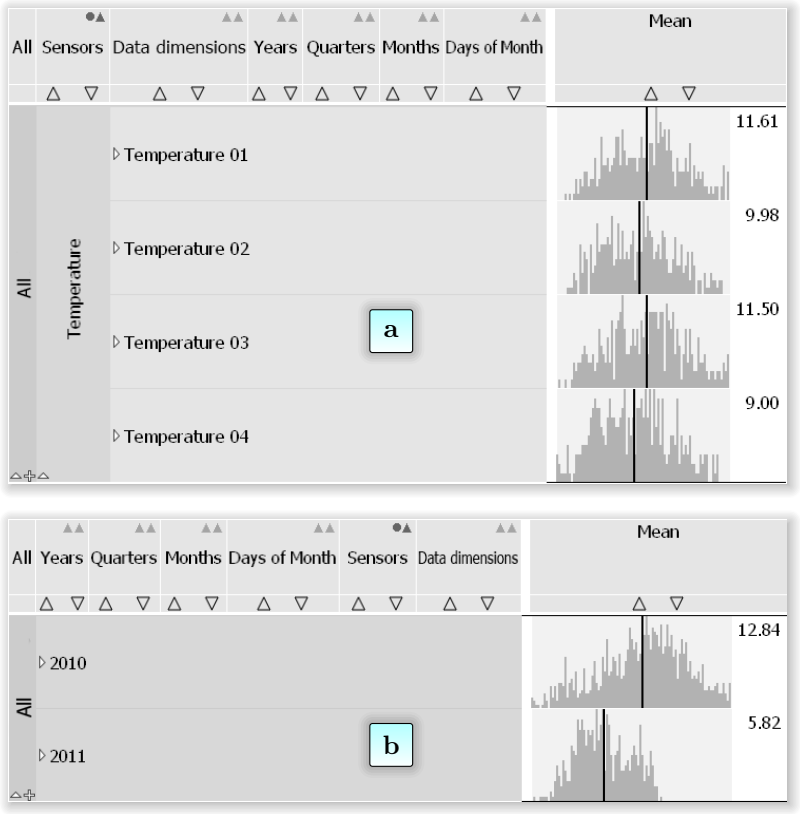

Figure 6.5: The rearrangement of the order of the hierarchy levels from comparison of *Data dimensions* (a) to the comparison of the *Years* (b) enables the user to analyze different data chunks in the cells of the data columns.

The HDO framework supports direct interaction techniques that enable the user to restrict the shown range. The user is able to modify the number of displayed sub-columns and the shown range of the scale of combinable data-chunks.

Fig. 6.3 shows the header of a column with sub-columns, which displays information about the shown range of their partitions. In this case the range is a temporal axis from May 2010 to April 2011. On demand, controls are shown to configure the column. Configuration options are for example: the number of displayed sub-columns, the omission of data by using a data filter, or the displayed range of a temporally partitioned column. The last point is shown in Figs. 6.3c and 6.3d, where the displayed range is changed by zooming.

To restrict the shown intervals of the axis of combinable data-chunks (see Section 5.5.3), the user may point to a cell of the tabular layout and the controls similar to the previous example appear.

6.2.4 Reordering the Tabular Layout

Tabular layouts are an intuitive way to compare and lookup values [LGS+14]. This allows the user to connect and relate the displayed visual representations as described in Section 2.4.

For a human, it is easier to compare representations that are closer to each other [Shn96]. This makes it necessary to interactively rearrange the ordering of the displayed rows and columns to support the user with this task. The interaction technique, which replaces the need for additional controls is called *Drag and Drop*. The user clicks on the header of a column and holds down the mouse button, while dragging the mouse cursor to the new position of the column, which is indicated with a blue line as visual feedback of the new position.

Columns that contain sortable data allow the user a reordering of the rows to a sorted ascending or descending column sequence. This action is performed by clicking on the arrows in the header of the column. If an arrow is filled in black, it indicates that this column is sorted. These controls enable the user to use the HDO framework as an RBFF, by sorting the columns that compute the desired features of the underlying data chunks.

Additionally the user may change the order of the hierarchy levels, by dragging the headers to another position. Fig. 6.5 shows the rearrangement of the hierarchy levels. The comparison of the individual *Temperature* sensors does not reveal specific structures in the data chunks (see Fig. 6.5a). By dragging the columns *Sensors* and *Data dimensions* to the end of the hierarchy, the comparison of *Years* of the underlying sensor data is possible and a new structure in the data is revealed (Fig. 6.5b).

CHAPTER 7

Implementation

This chapter describes the implementation process of the HDO framework, from initial sketches to a working implementation. The final framework was implemented in C++ and uses OpenGL for rendering. It was implemented as a plugin into an existing framework called .

7.1 Initial Sketch and Prototypes

The nine-stage design study methodology framework [SMM12], as described in Section 1.4, suggests designing the visualization before implementing it. For the first discussions of the used visualization design, sketches were made to communicate the intended results. Fig. 7.1 shows an early design sketch. The sketch already shows the tabular layout and some visual encodings that were used in the visualization. This section analyzes which design decisions did not change and which did change in the development process.

7.1.1 Design Decisions Regarding the Tabular Layout

The design decision of the tabular layout and the drill-down hierarchy (Fig. 7.1a) are elements that did not change the this early design decision. The shading of nodes on the same hierarchy level and controls were added for a better overview. Especially if some rows are collapsed and others are expanded, the shade of the row is an indication of the depth of their hierarchy level, although for hierarchies with a lot of levels the shade is no longer a suitable distinguishing feature.

Although other designs were also considered and tested in the early design process, the design decision of using an icicle-plot like representation of the hierarchy [MR10] as used in the sketch is also in the final implementation. The visualization design study Visplause [ASMP17] uses a similar tabular layout and the authors evaluated both the icicle-plot and the indented layout of hierarchies. The users prefer the indented layout of hierarchies. The advantage of an indented layout is that the context to the parent node

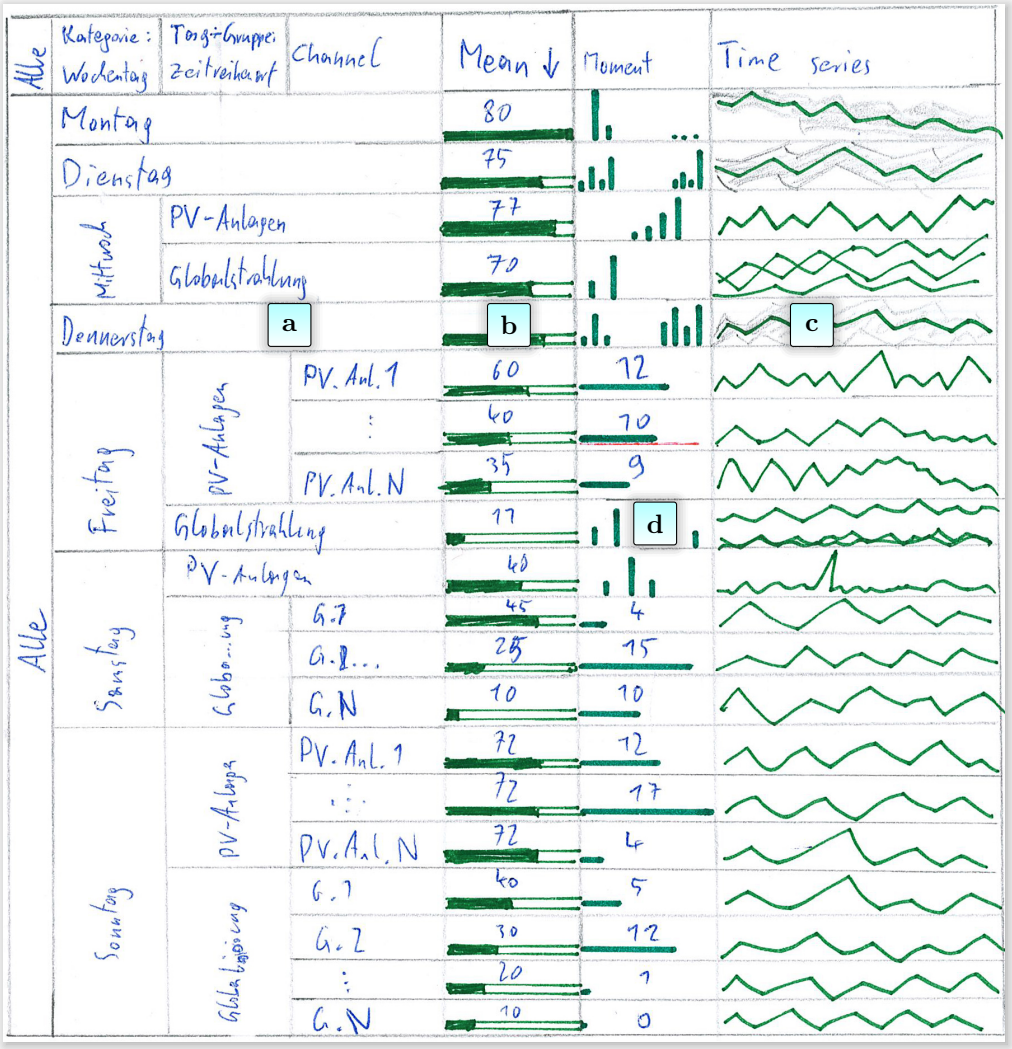

Figure 7.1: A sketch of an early design iteration. It shows the data from the guided example data set (see Section 1.1).

is not lost when expanding a node and that users are familiar with this encoding of a hierarchy [LGS+14]. An example of this is the folder hierarchy in file browsers.

Fig. 7.2 shows a screenshot of a prototype using an indented layout. The sub-total row *Dew Points* encodes the same information as the expanded rows, with an additional aggregated information. As this row consumes more vertical space than in an icicle-plot like representation and the additional information was not relevant while exploring the hierarchy, this design was not used.

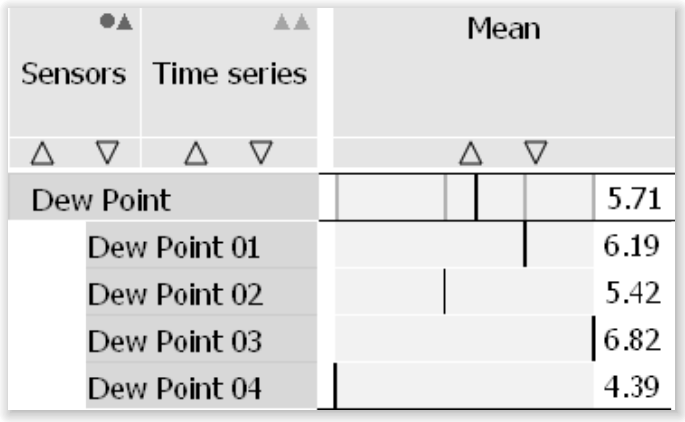

Figure 7.2: A prototype of the HDO framework used an indented layout of the table. This encoded redundant information and used more vertical space.

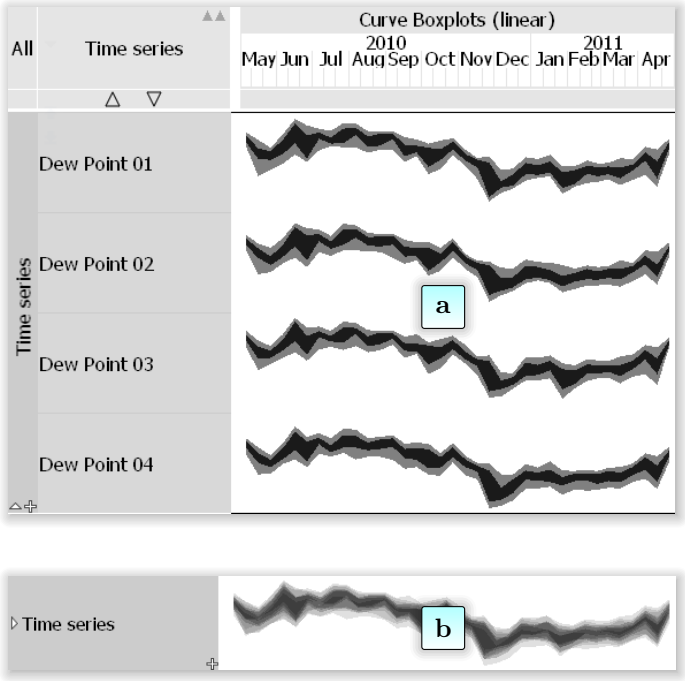

Figure 7.3: The concept of curve box plots [MWK14] is implemented as a column in the HDO framework for a single node (a) and a combined node (b).

7.1.2 Design Decisions Regarding the Visual Encodings

For example, the use of bars to indicate a central tendency as shown in Fig. 7.1b. Despite bars being a common technique to visualize values in a table [Mun14], lines are used for a more intuitive combination in one column. Which is already shown in Fig. 7.1d, where visual representations are mixed inside one column. In this case vertical bar charts for

combined data chunks and a horizontal bar for a single data chunk are shown.

The design decision of not mixing visual encodings also affected another considered visual representation. The concept of curve box plots [MWK14] was implemented as a column in the HDO framework. Whereby, in one sub-column the IQR is encoded as a black area and the 0.1 — 0.9 inter percentile range as a gray area. Additionally, the median is encoded as one line, similar to the line graphs shown in Fig. 5.5a. The mix of line and area graphs in one visualization may work for a single node, but the concept of combining multiple nodes (see Section 5.5.3) lead to multiple lines and areas with different shades.

In the simple case the gray shade of an area (black or gray) encoded a specific range, but it is no longer possible to visually interpret the grayscale in a combined node and assign it to a displayed value. By combining several surfaces by superimposition, one can recognize the boundaries of the underlying surfaces. The previously black areas of the IQR become gray. This results in losing the overview of which surface represents an IQR and which the gray 0.1 — 0.9 inter percentile range. Fig. 7.3 shows the implemented visual representation in a column of the HDO visualization, without the encoding of the median as a line. It shows that the combination of more than a single shade in one node (Fig. 7.3a) represents a problem when combining multiple nodes (Fig. 7.3b).

The initial sketch used a color to visualize an aggregated value in the combined cell and the underlying lines were grayed out (see Fig. 7.1c). Due to two reasons, this concept was not implemented. On the one hand, the aggregated representation of the cells in a partitioned column was not realized. This would not be as scalable as the superimposition method that is currently being used to combine the cells, as additional calculations would need to be made and additional memory needs to be used. It is also another representation which, as mentioned above, can be confusing and distracting for the user. On the other hand, color coding in the visual representations were omitted, since a cell can currently only be used for quantitative data dimensions. In future work, however, it is also be conceivable to partition the quantitative data dimensions in a cell using a categorical data dimension. For this, one would have to use color as a visually distinguishing feature. Such a cell with multiple categories is shown in Fig. 2.6.

7.2 Visplore as used Visualization Framework

is a visual exploration and model building system developed by the VRVis Research Center[1]. It supports a set of analysis tools on multivariate data. One basic supported visual analytics tool is multiple coordinated views. Fig. 7.4 shows an example of a visualization setup with multiple views.

The connection and coordination of different visualization techniques is a key concept of visual analytics. Fig. 7.4a shows the implemented HDO framework integrated in a Visplore configuration. The framework is currently not linked with other visualizations, but the advantages of an integration into a visualization system like Visplore are discussed in Section 9.2.2.

[1] VRVis Zentrum für Virtual Reality und Visualisierung Forschungs-GmbH, Vienna, `www.vrvis.at`

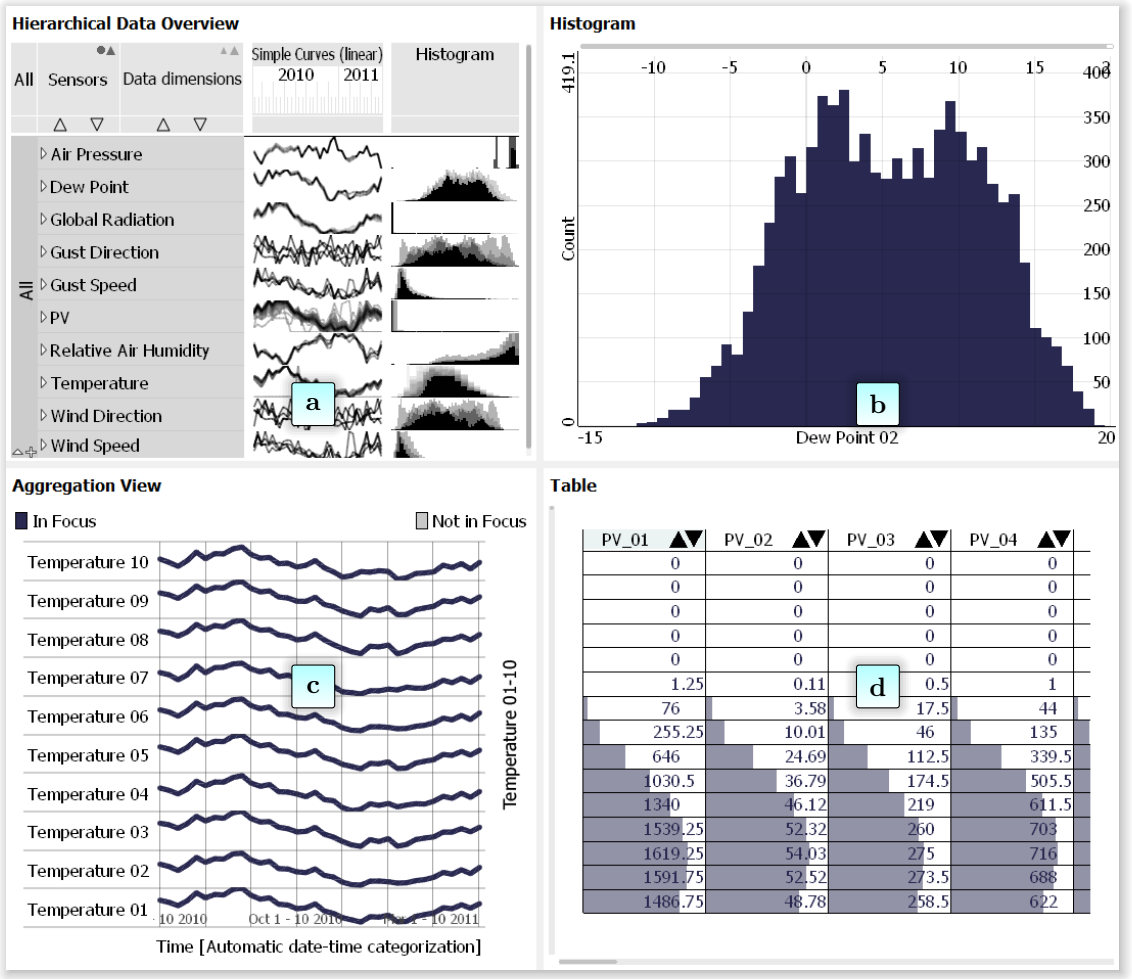

Figure 7.4: An example of multiple connected views in the software VISPLORE. This includes the presented HDO framework (a), Histograms (b), Time series (c) and detailed information (d).

The HDO framework was implemented in , because it implements a multi threading architecture [PTMB09]. This architecture enables the user to interact with the visualization in real-time and to receive instant visual feedback.

7.3 Real-Time Data Exploration

The visual feedback of the visualization is in real-time to help the user to explore the data rapidly and thereby support him to make decisions faster. However, the computational costs of computing statistical features on high-dimensional data are high. Several possibilities to achieve real-time capability are utilized. The hierarchy is employed to compute the results of the measurements on higher levels of it by a bottom-up approach.

This is, for example, possible for the statistic means of values. Unfortunately, not all statistics can be calculated by reusing intermediate results from lower levels. For example, the quantiles of multiple data chunks in an intermediate node of the hierarchy need to be recalculated by using the data of the combined data chunks.

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

7.4 System Architecture

The visualizations of Visplore implement the previously mentioned multi-threading architecture. In addition to a main thread, which waits for user input and should therefore carry out as few calculations as possible, there is a second thread. It performs the time-consuming calculations and places the resulting visual encoding on the monitor [PTMB09].

Since the calculations of the HDO framework can be very costly, because it is written for high-dimensional data, it uses another thread. Fig. 7.5 shows the three threads used:

7.4.1 Main Thread

As mentioned above, the main thread (see Fig. 7.5a) waits for input from the user. It knows which hierarchy is currently set and which data is to be calculated and forwards it to the appropriate threads.

The example in Fig. 7.5 shows a resize operation on the visualization by the user (Fig. 7.5d). This new state is passed from the main thread to the data model thread to recompute the partitioning and to the visualization thread to render the resized table. These threads are stopped, the new state is set, the dependent calculations are invalidated and the thread is forced to restart.

7.4.2 Data Model Thread

The data model thread (see Fig. 7.5c) is responsible for the complex calculations of the HDO framework. As Fig. 7.5 shows, important parts of it are:

- Creating the hierarchy (Fig. 7.5e). This includes the creation and deletion of hierarchy nodes, depending on the user defined hierarchy.
- Dividing the data into data chunks (Fig. 7.5f).
- Partitioning the data chunks for individual columns of the visualization, like the histogram or the time series column (see Fig. 7.5g and Section 5.5.2).
- Calculating the statistics (see Fig. 7.5h and Section 5.5.1).
- Creating the cells (Fig. 7.5i) for visualization, while taking the user defined combination aspects and visual mappings into consideration (see Section 5.5.3).

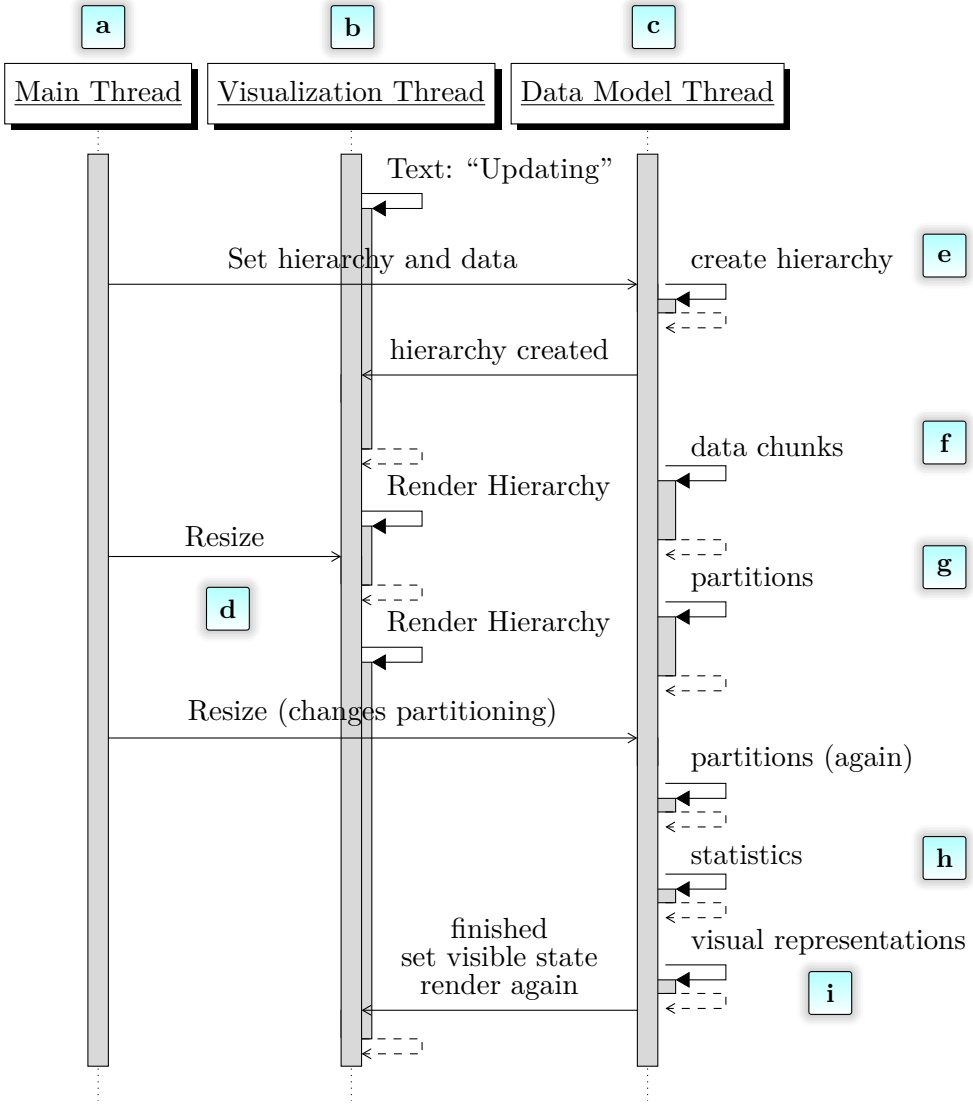

Figure 7.5: Sequence diagram of the three used threads of the HDO visualization. The main thread (a) waits for the user input (d). The visualization thread (b) renders the state. The data model thread (c) performs heavy computations (e–i).

Every part depends on all previous calculations. If a part is invalidated by user interaction, as for the resize operation in example of Fig. 7.5d, all interdependent parts are invalidated as well. After the thread starts again, the invalidated parts are recomputed.

As soon as the data model thread calculates a part that can be displayed in the visualization, this state is forwarded to the visualization thread.

7.4.3 Visualization Thread

To ensure that the displayed information corresponds to the current user input, there is another thread. The visualization thread (see Fig. 7.5b) takes care of drawing the visual representations. The elements to be represented are the table, the hierarchy, the cells, and the direct interaction controls.

This thread uses the current state of the data model to draw these elements. As described in Section 7.4.2, the data model thread forwards changes as a message to the visualization thread. As soon as a new message arrives, the current drawing operations are terminated as quickly as possible, the state is updated and the drawing operations are restarted.

This allows operations that trigger frequent changes to expensive operations, such as drawing cells, to create fluid animations. Resizing the hierarchy, as shown in Fig. 7.5, is an example.

CHAPTER 8

Evaluation

To show that the design process of the HDO framework was successful and the identified goals were achieved, a usage scenario was written. First, the data set selected for the identified user group is presented. Then the possible users are listed and the data is analyzed using several scenarios.

8.1 Evaluation Data Set — SmartMeter Energy Consumption Data in London Households

The design process was motivated by the tasks in the energy sector. Grid operators have two major driving forces to monitor. The first is the power supply from different sources, which was already presented by the first data set. The possible fields of application were already pointed out in the thesis (see Section 1.1). The second influencing variable is the power consumption of the consumers. In order to also deal with this driving forces and the resulting questions, a further data set from the area of electricity consumption is presented here. The data set "SmartMeter Energy Consumption Data in London Households" contains energy consumption measurements from 5,567 London households. They participated in the UK Power Networks led Low Carbon London project between November 2011 and February 2014. This open data set was published by the London City Council and can be downloaded from its homepage [Net15].

Each household was assigned to a A Classification Of Residential Neighbourhoods (ACORN) group in 2010. The ACORN structure categorizes households into different social groups based on different factors. Table 8.1 shows the categorizations, which are used in the data set. ACORN also subdivides the households furthermore into 62 types, but this level of detail is not available in the data. The data of these analyzed factors include property prices, company directors, consumer data from lifestyle surveys and many more. In order to represent the greater London population well, customers were selected from all groups in a balanced way.

Table 8.1: The CACI ACORN classifies households into 6 categories and 18 groups.

ACORN Category	**ACORN Group**
Affluent Achievers	**A** Lavish Lifestyles **B** Executive Wealth **C** Mature Money
Rising Prosperity	**D** City Sophisticates **E** Career Climbers
Comfortable Communities	**F** Countryside Communities **G** Successful Suburbs **H** Steady Neighborhoods **I** Comfortable Seniors **J** Starting Out
Financially Stretched	**K** Student Life **L** Modest Means **M** Striving Families **N** Poorer Pensioners
Urban Adversity	**O** Young Hardship **P** Struggling Estates **Q** Difficult Circumstances
Not private Households	**R** Active and Inactive communal population, without resident

The data set contains the energy consumption in kiloWatt hour (kWh), the unique household identification, date and time as well as the CACI ACORN group. The measured values were determined every half hour and contain about 167 million rows.

Within the data set there are two groups of customers. The first is a subgroup of around 1100 customers who were exposed to Dynamic Time of Use (dToU) energy prices throughout the 2013 calendar year. The tariff prices were transmitted one day in advance via the Smart Meter In Home Display (IHD) or an SMS to the mobile phone. Customers were notified of high (67.20 pence/kWh), low (3.99 pence/kWh), or normal (11.76 pence/kWh) price signals and the relevant times of the day. The data/times and the price signal plan are available as part of this data set.

The signals given were designed to be representative of the type of information that can be used in the future for high renewable generation operation (supply tracking). Electricity generation from renewable energy sources has a disadvantage for the electricity supply system. It can vary greatly. This means that the balance between electricity supply and demand can be disturbed more quickly. Additionally the signals were designed for testing the potential to use high-price signals to relieve local distribution networks in times of stress.

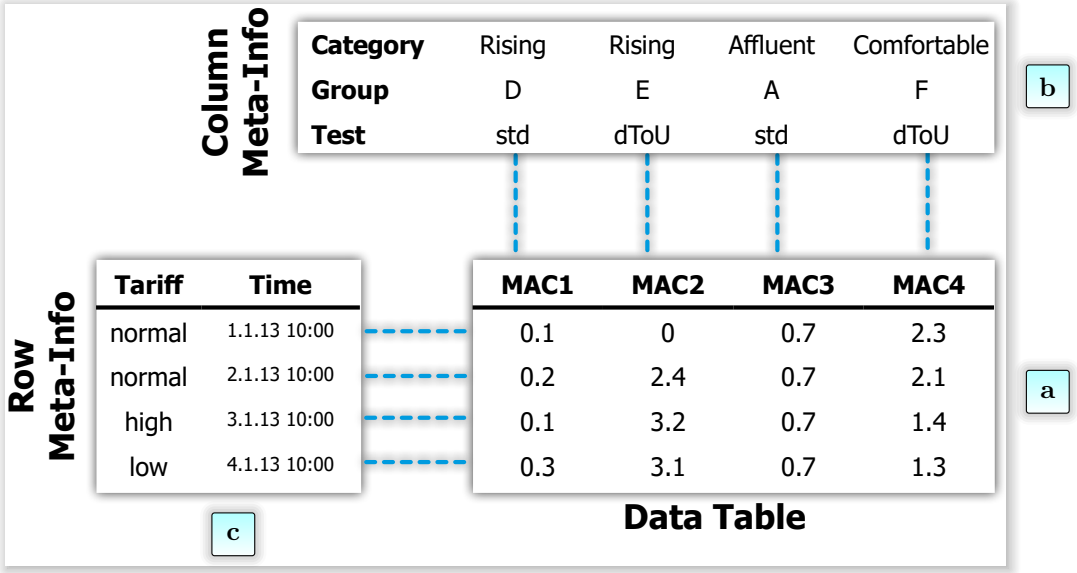

Figure 8.1: The shown data table (a) contains the power consumption values of London households. Meta-information is present on the columns (b) and on the rows (c)

The remaining sample of approximately 4500 customer energy consumption measurements did not fall under the dToU tariff. These customers received a flat rate of 14,228 pence/kWh. They are further noted as the Standard (std) group.

The transformation into the data model of the HDO framework is easy to perform. Fig. 8.1 shows a small data table as an example. The different household identifications define the columns (see Fig. 8.1b) of the data table and thus the data dimensions. These receive two meta-informations to be able to group them. On the one hand they can be grouped by the CACI ACORN categorization, which has two levels of detail. On the lower level of detail the category and more specifically the group (see Table 8.1) can be used for grouping. On the other hand they can be grouped whether the household belongs to the dToU test group or to the standard group. One row of the table contains the power consumption measurement of half an hour for all households in kWh (see Fig. 8.1a). Also meta-informations per data record are stored in the data. The tariff price category (high, low or normal) shows the current pricing for the customer. Additionally the acquisition date of the measurement is stored (see Fig. 8.1c).

8.2 Usage Scenario — Visual Analysis with the Hierarchical Data Overview Visualization

When analyzing data, we assume that the user already has prior knowledge. In addition, the data has to be already in the data model required by the framework. In the case of the data in Section 8.1, a data transformation had to be applied to convert the data into the used data model.

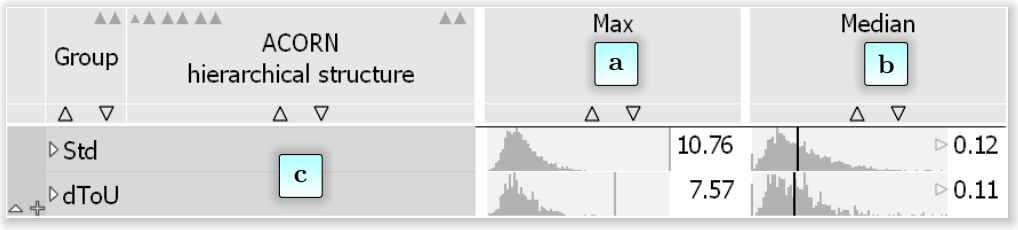

Figure 8.2: The HDO visualization is used to give an overview of the smart meter dataset. The columns visualize the maximum (Fig. 8.2a) and the Median (Fig. 8.2b) of the power consumption of the households, which are grouped by the rows (Fig. 8.2c) into the dToU group and the remaining households (std).

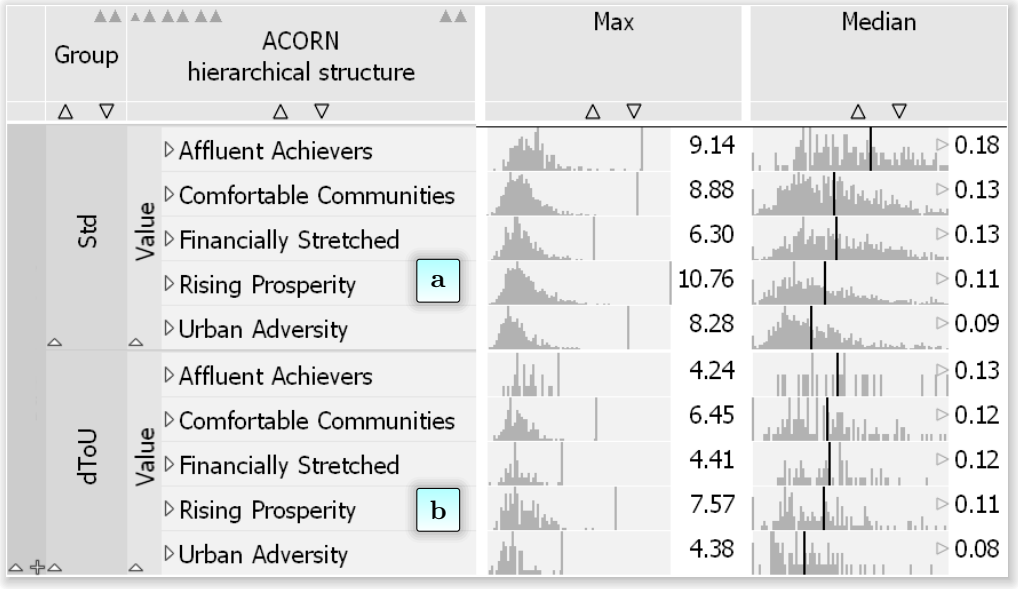

Figure 8.3: The hierarchy from Fig. 8.2 expanded for the ACORN categories (compare with Table 8.1). In both figures it can be seen that the households from the dToU group have a lower median power consumption and lower maximum peaks.

The first question the network operator would like to have answered about this data set could be: *Does switching to Smart Meter devices relieve my network?*

This question is relevant for the grid operator, since peak power is the most critical factor for the power supply. The power grid needs to be dimensioned for this peak power consumption.

As the visual information seeking mantra makes clear, a good starting point for the analysis of the table is to get an overview of the households. The visualization with the HDO framework in Fig. 8.2 shows the households of the dToU group and the remaining households (std, Fig. 8.2c) in the two lines. In the cells of the table, aggregated statistical factors of each household are visually encoded. The first column shows the maximum values (Max, Fig. 8.2a) and the second column shows the medians (Fig. 8.2b). The data

is mapped from a default domain range to the cell area as described in Section 5.5. This default domain range is defined to map from the minimum, to the maximum value of the underlying data. The small white arrows on the right side of the cells of the median column indicate that the displayed target range was reduced (see Section 6.2.3). This reduction was applied to omit outliers in the data and limit the visualization to the distribution of the majority of the data.

At a glance, the grid operator can see that the electricity peaks of the households in the dToU group are by a quarter lower than those of the remaining households (7.57 vs 10.76 kWh). Also the central value of consumption in the test group was reduced from 0.12 to 0.11 kWh. This visual summary of all data dimensions indicates that the goal G1 was met.

The next question the network operator may now be interested in is: *How much is the power consumption for the different user groups?*

To address this question, the hierarchy can quickly be expanded. In Fig. 8.3 the hierarchy is expanded for the ACORN categories (compare with Table 8.1). Again, the user can easily see that in all categories the test group had lower maximum values and medians. The rows Fig. 8.3a and Fig. 8.3b show that the category "Rising Prosperity" has the same median.

Furthermore, the analyst might wonder why the median value for the "Rising Prosperity" group in the dToU group is lower than that of the ACORN categories "Comfortable Communities" and "Financially Stretched". To analyze this more closely, the analyst reduces the displayed data dimensions to the dToU group and adds two more statistics and thereby two columns. In Fig. 8.4 the columns show the median (as before), the IQR, and the histogram. It is noticeable that the dispersion of the category "Rising Prosperity" is particularly high (Fig. 8.4a, 0.18 kWh) and also the histogram indicates multi-modal behavior in comparison to the other categories (Fig. 8.4b). To analyze this category more closely, the user can expand the hierarchy further and get an overview of the two ACORN groups "D — City Sophisticates" and "E — Career Climbers" (see Fig. 8.4c). Some insights can be gained from the visualizations shown. By the smaller number of lines in the upper cell, one sees that the sample size of group D is smaller than that of group E. The histogram of the upper row have several modalities. These are rather outliers, which also produce high dispersion (0.20 kWh). The median of group E is 0.03 kWh lower than the median of group D. From the histograms and the IQRs can also see that the households have very different distributions. By further breaking down the hierarchy, the individual households can now be analyzed more precisely. This shows that the goal G2 was achieved.

Further analysis could be done by adding one or more columns with a temporal partitioning. This has been discussed in more detail in Section 5.5.2. In Fig. 8.5c one sees the aggregated power consumption of the households for each hour of the day. In Fig. 8.5d one sees the dispersion of the daily curves. Furthermore, it is relevant to mention that in Fig. 8.5a the separation of data chunks by ACORN categories and groups were switched from a common scale (▲▲) to no common scale (▲▲) (see Section 5.5.3). This allows the analyst to compare the curves in the cells with each other, but no longer

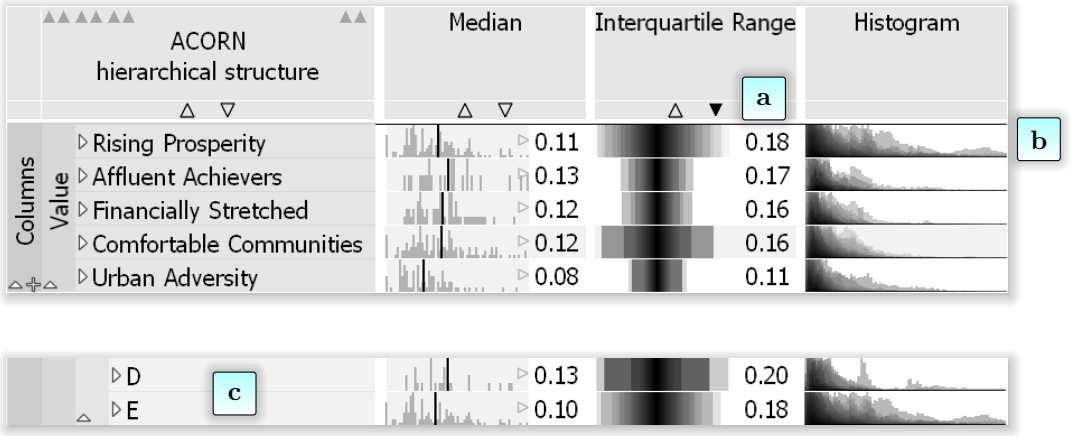

Figure 8.4: Statistical properties of the power consumption of the households from the dToU group. The rows are grouped by the ACORN categories and sorted by the IQR (a). The category "Rising Prosperity" shows a broad distribution (b) and in (c) the category is split up into the ACORN groups D and E.

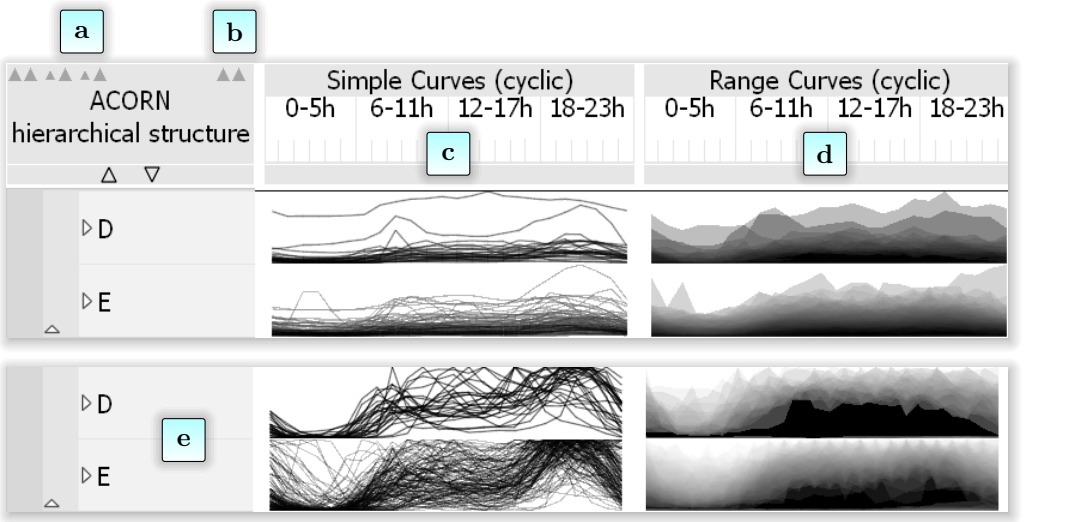

Figure 8.5: The columns of the HDO visualization are partitioned by the hours of a day creating day curves (c) and (d). The curves in the upper Figure are combined with no common scale (▲▲) for the ACORN categories and groups (see (a)). The curves of the lower figures (e) have their own scale

to compare the cells with each other. Again it can be seen that the curves describe very different power consumptions. One also sees recurring patterns regarding the time of day. If each curve is displayed with its own scaling (Fig. 8.5e), one can clearly see hourly patterns in the consumption curves:

- Overnight from 0–5 a.m. there is a phase in which little energy is consumed (with

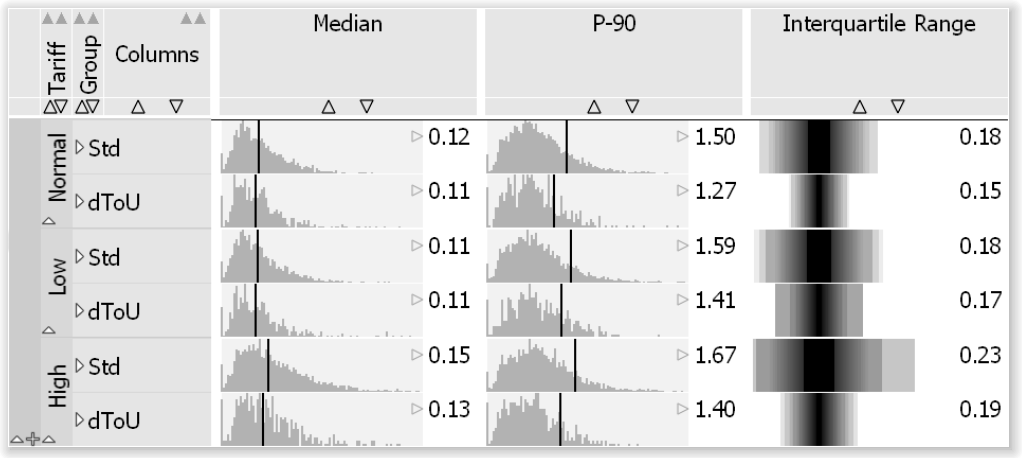

Figure 8.6: The power consumption of the households are divided by the tariff and then grouped by the dToU and std group. One can observe that the IQR and the P-90 are smaller for all tariffs for the dToU group

a few exceptions in Group E).

- In the morning there are some peaks.
- During the day diverse patterns occur.
- In the evening from 18 p.m. the largest consumption can be measured.

Another question a network operator would like to have answered is: *What influence does the tariff have on the dToU test group?*

As shown in Section 8.1 the tariff is a meta-information of the data records. Therefore, another hierarchy level can be added, which divides the data dimensions again into data chunks. In Fig. 8.6 the newly added partitioner is shown in the first column. Again, the network operator can quickly get an overview of the data via the visualizations in the cells. For example, it is possible to see that there is a reduction of the median for the two tariffs "Normal" and "High", but for the tariff "Low" the median consumption stays at 0.11 kWh. The next column computes the 90-Percentile (P-90) of the data chunks. This is a more robust statistical property than the maximum, because outliers less influence the value. As one can see, the dToU group has a lower P-90 value for all three tariffs, which indicates that the power consumption peaks are lower for all tariffs for the dToU as compared to the std group. This can also be seen in the IQR-column.

CHAPTER 9

Discussion

This chapter concludes the analysis phase (see Section 1.4.3) of this paper. First, the evaluation and implementation are critically reviewed. Then, possible extensions are presented.

9.1 Reflection

9.1.1 Reflection of the Goals

To determine whether the HDO framework has achieved its purpose, the defined goals for the design (see Section 3.3) were evaluated in a usage scenario (see Chapter 8). This section looks back at the scenario and draws conclusions whether the goals were achieved.

G1 — Visual Summaries of Groups

In order to determine whether the visual summaries of groups (G1) were meaningful enough, one must first ask oneself whether the definition of related groups is sufficient. In the HDO framework these groups were defined by the meta-informations of the data via the hierarchy levels. Thus, different subdivisions of data chunks could be created in the usage scenario:

- whether the households are in the dToU group or not (in the std group)
- according to the ACORN hierarchy.
- according to the tariff

In addition, all combinations of these groupings can be achieved. This gives enough flexibility to create data chunks. Even though, the meta-information has to be present in the data, if this is not the case, one has to manually define it, which can be cumbersome. Possible extensions are presented in Section 9.2.1.

The visual summaries of the statistical properties of the data chunks depend on the statistics used and whether a partitioned column (e.g., time curves, Fig. 8.5e) is considered. The visualizations can be further adapted by direct interactive configuration possibilities of the representations. The usage scenario gives an overview of the visualizations through the Figs. 8.2 to 8.6 shown. In addition, the scenario shows that questions can be solved by the visual summaries of the groups.

G2 — Flexible Drill-Down and Roll-Up

How the data table of the usage scenario is divided into data chunks and the resulting flexibility of exploring them, has already been shown in the discussion of the previous goal. Section 6.2.2 describes the interaction possibilities to expand or collapse a node in the hierarchy. This principle is used in Fig. 8.4 to analyze a node with a particularly wide distribution more precisely. The node is drilled-down and the user is able to get an insight into the distribution of the data chunk (task T3).

Furthermore, one can see in Fig. 8.4a that the rows of the table are sorted by IQRs. By the possibility of sorting, nodes with a certain feature can be found faster (task T2). Since both tasks can be fulfilled by the design of the HDO framework, one sees that the goal G2 was achieved.

G3 — Scalability

Section 1.1 introduces the first data set of PV time series, which has a raw data table with 163 columns and 8756 rows. The computation time of the shown calculations of the statistics and rendering the visualizations is lower than one second. This shows that the HDO framework is usable for data with multiple data dimensions. The guiding example data set is rather small in contrast to the evaluation data set (Section 8.1). To show the scalability of the framework the performance of this data is evaluated.

The data set from the usage scenario has 167 million rows and 5.567 columns. The raw data table of this record is over 1.4 million times larger than the first one. Since the calculations on this data are much more expensive, a user has to wait up to ten seconds for calculation results. In order to give the user feedback as fast as possible and to enable further interactions, the calculations are executed in a thread. This architecture was introduced in Section 7.4 and the usage scenario showed that it also works for a large data set.

While computing the statistics of the data chunks the visualization shows the current progression of the operation as a percentage. To give the user even more feedback than just the current progress, the calculations could be extended. One possibility would be to progressively visualize every finished data chunk and not to wait for all calculations first. If this extension still takes too long, it would be possible not to use all the chunk's data for the statistical calculations, but only a subset of it. Only when a first preview visualization is available, the complete calculation will be done. Thus, the user receives faster feedback about which results he can expect and can change his configuration early on.

9.1.2 Extended Evaluations

The evaluation method used is called usage scenario. It describes a real-world example of how one would interact with the system. It discusses the steps, events, and/or actions that occur during the interaction. Although this technique is synthetic and involves only a hypothetical user, it helps to evaluate the HDO framework.

Empirical studies were proposed for the evaluation of information visualizations [Car08, LBI+12]. A more detailed evaluation of an information visualization can be conducted for example by case studies or controlled experiments. In case studies domain experts use the visualization and then answer questions about it. Due to the complexity and open-endedness of the analysis of data, however, the evaluation of these studies is difficult. In controlled experiments one focuses on a few aspects of the analysis, these are then often made comparable by means of metrics. Such metrics include [WHA07]:

- The number of revisits
- The number of unique discoveries
- Subjective preferences based on log data.

Using such metrics also makes it difficult to obtain quantitative evaluation results. This is mainly due to the fact that the number of domain-specific users is low. Therefore, a good extension of the existing evaluation would be a qualitative study like a *Case Study*.

9.1.3 Abstractions to other Domains

The HDO framework was designed specifically for time-series data from the energy sector. However, this concept is by no means limited to this area of application. Not only analysts in the energy sector need to analyze high-dimensional time-series data. Meta information is also collected in other domains and can therefore be used flexibly for the framework.

Experiences with time-series data, to which this framework could be applied, have already been made with simulation data in the automotive sector, with patient data in the health sector, and with process data from production engineering through many years of cooperation with research partners of VRVis. The defined tasks and goals (see Chapter 3) from this thesis are also present in these domains.

9.2 Future Work

9.2.1 Automatic Separation of Data Chunks

The hierarchy levels of the partitioning of the data table into data chunks used meta-information on the data dimensions and data records (see Section 5.4). It is possible that the data set you want to analyze is not sufficient or does not contain any meta-information at all. This results in losing the ability to subdivide the data table, which is an important

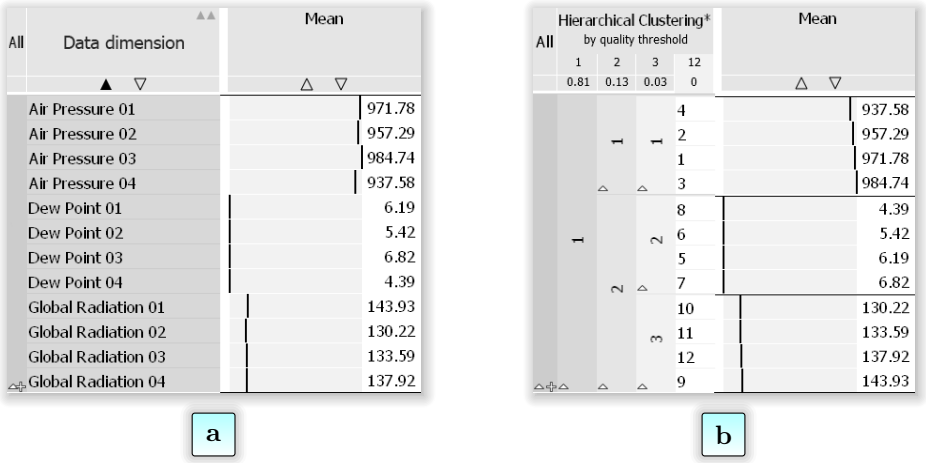

Figure 9.1: A prototype for the automatic separation of data chunks

feature of the HDO framework. In this case automatic analysis of the data chunks would be a possibility to support further separation of the data.

Fig. 9.1 shows a prototype of this concept. In Fig. 9.1a the data chunks of three different sensors are assigned to the HDO framework without a hierarchy level to combine the data dimensions from the same sensor. Fig. 9.1b shows the same data with a hierarchy level that separates the data chunks by a hierarchical agglomeration clustering, using the values of the mean column as a feature vector [JMF99].

9.2.2 Integration in Visplore

In the introduction (see Section 2.2) Keims's visual information seeking mantra [Kei05] was presented. The conceptual approach of the HDO framework follows this mantra: We *analyze first* by subdividing the data into data chunks *show the important* by focusing on visualizing descriptive statistics of the chunks, support to *zoom, filter, and analyze further* via the hierarchy and various columns. To display *details on demand* the visualization currently supports tooltips showing detailed information while hovering the visual representations. By integrating the framework into a visual analytics software Visplore, more methods for more detailed analysis can be applied.

The HDO framework has been developed as a plugin for Visplore as described in Section 7.2. However, the presented framework is currently not fully connected to methods that the visual analytics software support. This will be added as future work, because the integration with these methods will support the user tasks. The integration into an existing software allows users to use known visual analytics patterns, like brushing and linking. Additionally the analyst has access to already implemented and well known information visualizations techniques (e.g., histograms).

Brushing and Linking

Brushing and linking are interaction techniques used to combine different visualization methods. The selections (brushing) of data subsets of the single visualizations are reflected in all other visualizations (linking) [Kei02]. In the context of the HDO framework, a user may select features if he is interested in a separate visualization. Then he can observe the computed and visualized descriptive statistics for the selected data subset in the HDO visualization. Alternatively, a user may select data chunks in the hierarchy to analyze the data chunks in other visualizations.

Details on Demand

The last part of Keims's visual analytics mantra is called "Detail on Demand". The cells of the table in the HDO framework visualize the data in a concise way. Because of limited screen space, some information has to be omitted. Visualization elements like axes, tics, and labels may be useful for some cases, but visualizing every detail can be impractical [Shn96].

To create the balancing act between displayed information and visual clutter in the limited available screen space, details are only shown when they are needed. To apply this pattern to the HDO framework's tabular display, VISPLORE would need to display additional information for special interactions. One way to achieve this would be to click on a cell in the table and display an extended visualization of the cell next to the table as a linked view [Kei02]. Examples of such fully configured extended visualizations can be seen in Fig. 7.4: The expanded histogram cell can be seen in Fig. 7.4b. The extensions include a bigger display area than the cell, interaction possibilities like filtering and further details on demand like tooltips. This figure shows the distribution of only one time series (*Dew Point 02*). It would also be conceivable to combine several time series of a combined cell.

Combined cells could also be split and displayed side by side (see Fig. 7.4c). In this figure ten *Temperature* time series are shown, just as with the cells, the individual points of the time series are aggregated into partitions and only the aggregated values are connected to lines. Another possibility to display time series would be to visualize all single values.

An exploration of the single values is the finest granular detail. In addition to a time-series representation, another possibility would be the representation as a table. Fig. 7.4d shows the single values of four data dimensions in a table.

CHAPTER 10

Conclusion

This thesis describes an HDO framework to efficiently inspect and compare high-dimensional data. Motivated by the tasks of domain experts in the energy domain, three design goals are defined for this framework: Visual summaries, flexible interaction, and the scalability for high dimensional data.

Based on these goals the HDO framework is described, 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. The interactive refinement of the displayed rows and the flexible configuration of the columns of the tabular layout support the interactive exploration tasks of domain experts.

A usage scenario evaluates the design of the framework with a data set of the target domain in the energy sector. The scenario shows that the approach is appropriate to address the identified goals. While the design aspects of the HDO framework are considered as the main contribution, the conceptual approach also follows Keim's visual analytics mantra [Kei05]: We *analyze first* by subdividing the data into data chunks *show the important* by focusing on visualizing descriptive statistics of the chunks, support to *zoom, filter, and analyze further* via the hierarchy and various columns, and show *details on demand* by integrating the visualization into the visual analytics software VISPLORE.

List of Figures

List of Tables

Acronyms

PB petabyte. 1

TB terabyte. 1

kB kilobyte. 1

kW kiloWatt. 31, 32

kWh kiloWatt hour. 64, 65, 67, 69

ACORN A Classification Of Residential Neighbourhoods. 63–68, 71

CAVE recursive acronym (CAVE Automatic Virtual Environment). 16

dToU Dynamic Time of Use. 64–69, 71

HDO Hierarchical Data Overview. 3–5, 23, 26, 30, 31, 33, 35–37, 47–49, 52, 53, 55, 57–61, 63, 65, 66, 68, 71–75, 77, 79, 85

IHD In Home Display. 64

IQR Inter-Quartile Range. 39, 50, 58, 67–69, 72

MDS Multidimensional Scaling. 17

OLAP On-Line Analytical Processing. 20

P-90 90-Percentile. 69

PCA Principle Component Analysis. 17

PV Production Values. 2, 3, 18, 30, 32, 33, 49, 72

RBFF Rank By Feature Framework. 4, 18, 25, 37, 53

RDS Resource Description Framework. 32

SOM Self Organizing Maps. 17

std Standard. 65, 66, 69, 71

TDTT Task by Data Type Taxonomy. 9, 11, 12

VHDR Visual Hierarchical Dimension Reduction. 18

VRVis VRVis Research Center[1]. vii, 23, 58, 73, 86

[1]VRVis Zentrum für Virtual Reality und Visualisierung Forschungs-GmbH, Vienna, `www.vrvis.at`

Glossary

box plots A box plot displays the minimum, first quartile, median, third quartile, and maximum [Yu77] of a set of data. 20, 57, 58

brushing Selecting data records from a visual representation with the intention to highlight them. 9, 86, *see also* linking

CACI CACI International Inc is an American multinational professional services and information technology company headquartered in Arlington, Virginia. The acronym stands for "Consolidated Analysis Center, Incorporated". 64, 65

data chunk A subset of the available data dimensions and data records. 33–41, 43–45, 50–53, 58, 60, 61, 69, 71–75, 77, 85, 86

data dimension A property that can be measured, observed, or logged. Such as a column in a data table. 2, 4, 7, 10–14, 17, 18, 20, 24–26, 29–37, 44, 45, 49, 52, 53, 58, 60, 65, 67, 69, 72–75, 77, 85

data record A discrete individual entity. Such as a row in a data table. 1, 4, 7, 9–11, 17–21, 24, 26, 29–35, 37, 51, 65, 69, 73, 85, 86

data transformation Map the raw data to data tables. 10, 30, 65, *see also* visual mapping & view transformation

flow visualization An application area of scientific visualization, which is used to make flow patterns visible. 8

hierarchy level Subdivides the data table into data chunks. All hierarchy levels define the hierarchy of the HDO framework. 35, 37, 38, 43–45, 49–53, 55, 69, 71, 73, 74, 85, 86

information overload "Information overload occurs when the amount of input to a system exceeds its processing capacity. Decision makers have fairly limited cognitive processing capacity." [SVV99]. 1, 2

information visualization "Information visualization is the communication of abstract data through the use of interactive visual interfaces." [KMSZ06]. 2, 7–10, 12, 14–16, 20, 21, 23, 26, 73, 74

linking The selected data subset from brushing are highlighted in all other visual representations that contain these data records. 9

Moore's law "Moore's Law" predicts the future of integrated circuits. He observed that the number of transistors in a dense integrated circuit doubles approximately every two years [Moo65]. 1

pivot table A pivot table summarizes the data of a larger table. It may be used to reorganize and summarize the larger table, so that the output data may be represented in a condensed, summarized form due to the aggregation used in the data fields. 20

▲▲ The data chunks of the hierarchy level share the same scale. 43, 44, 67, *see also* ●▲ & ▲▲

●▲ The data chunks of the hierarchy level cannot be combined. They do not share the same scale and can not be rendered in the same cell.. 43–45, *see also* ▲▲ & ▲▲

▲▲ The data chunks of the hierarchy level do not share the same scale. 43–45, 67, 68, *see also* ▲▲ & ●▲

scientific visualization "The graphical representation of complex physical phenomena in order to assist scientific investigation and to make inferences that are not apparent in numerical form." [Fod02]. 8, 85

view transformation Create views of the visual structures. 10, *see also* data transformation & visual mapping

VISPLORE is a visual analytics software developed by the VRVis. 5, 13, 14, 21, 22, 55, 58–60, 74, 75, 77

visual analytics "Visual Analytics is the science of analytical reasoning facilitated by interactive visual interfaces." [Tho05]. xi, 2, 7, 8, 12–16, 19, 21, 23, 58, 74, 75, 77, 79, 86

visual information seeking mantra "Overview first, zoom and filter, then details-on-demand" [Shn96]. 9, 14, 19, 26, 51, 66, 74

visual mapping Transform data tables into visual structures. 8, 10, *see also* data transformation & view transformation

Bibliography

[AFG+09] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D Joseph, Randy H Katz, Andrew Konwinski, Gunho Lee, David A Patterson, Ariel Rabkin, Ion Stoica, et al. Above the clouds: A Berkeley view of cloud computing. Technical report, Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, 2009.

[AMST11] Wolfgang Aigner, Silvia Miksch, Heidrun Schumann, and Christian Tominski. *Visualization of time-oriented data.* Springer Science & Business Media, 2011.

[AS94] Christopher Ahlberg and Ben Shneiderman. Visual information seeking: Tight coupling of dynamic query filters with starfield displays. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 313–317. ACM, 1994.

[ASMP17] 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.

[BKC+13] Rita Borgo, Johannes Kehrer, David HS Chung, Eamonn Maguire, Robert S Laramee, Helwig Hauser, Matthew Ward, and Min Chen. Glyph-based visualization: Foundations, design guidelines, techniques and applications. In *Eurographics (STARs)*, pages 39–63, 2013.

[BR07] Mike Botts and Alexandre Robin. OpenGIS sensor model language (SensorML) implementation specification. *OpenGIS Implementation Specification OGC*, 7(000), 2007.

[Bre16] Matthew Michael Brehmer. *Why visualization?: task abstraction for analysis and design.* PhD thesis, University of British Columbia, 2016.

[Bru11] Geoff Brumfiel. High-energy physics: Down the petabyte highway. *Nature News*, 469(7330):282–283, 2011.

[C+85] William S Cleveland et al. *The elements of graphing data.* Wadsworth Advanced Books and Software Monterey, CA, 1985.

[Car08] Sheelagh Carpendale. Evaluating information visualizations. In *Information visualization*, pages 19–45. Springer, 2008.

[CC05] Brock Craft and Paul Cairns. Beyond guidelines: what can we learn from the visual information seeking mantra? In *Ninth International Conference on Information Visualisation (IV'05)*, pages 110–118. IEEE, 2005.

[CCK+83] John M Chambers, William S Cleveland, Beat Kleiner, Paul A Tukey, et al. *Graphical methods for data analysis*, volume 5. Wadsworth Belmont, CA, 1983.

[CCS93] Edgar F Codd, Sharon B Codd, and Clynch T Salley. Providing OLAP (on-line analytical processing) to user-analysts: An IT mandate. *Codd and Date*, 32, 1993.

[Che73] Herman Chernoff. The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, 68(342):361–368, 1973.

[CMS99] Stuart K Card, Jock D Mackinlay, and Ben Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.

[CNSD93] Carolina Cruz-Neira, Daniel J Sandin, and Thomas A DeFanti. Surround-screen projection-based virtual reality: the design and implementation of the CAVE. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 135–142. ACM, 1993.

[DE02] Alan Dix and Geoffrey Ellis. By chance enhancing interaction with large data sets through statistical sampling. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 167–176. ACM, 2002.

[Dix09] Alan Dix. *Human-computer interaction*. Springer, 2009.

[DS14] Norman R Draper and Harry Smith. *Applied regression analysis*. John Wiley & Sons, 2014.

[ED07] Geoffrey Ellis and Alan Dix. A taxonomy of clutter reduction for information visualisation. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1216–1223, 2007.

[EF10] 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.

[EK02] Stephen G Eick and Alan F Karr. Visual scalability. *Journal of Computational and Graphical Statistics*, 11(1):22–43, 2002.

[Few09] Stephen Few. *Now you see it: simple visualization techniques for quantitative analysis*. Analytics Press, 2009.

[FK70] Federico Faggin and Thomas Klein. Silicon gate technology. *Solid-State Electronics*, 13(8):1125IN11131–1130IN21144, 1970.

[Fle01] Arthur Flexer. On the use of self-organizing maps for clustering and visualization. *Intelligent Data Analysis*, 5(5):373–384, 2001.

[Fod02] Imola K Fodor. A survey of dimension reduction techniques. Technical report, Lawrence Livermore National Lab., CA (US), 2002.

[FP02] Jean-Daniel Fekete and Catherine Plaisant. Interactive information visualization of a million items. In *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002.*, pages 117–124. IEEE, 2002.

[FR14] Muriel Foulonneau and Jenn Riley. *Metadata for digital resources: implementation, systems design and interoperability*. Elsevier, 2014.

[Fra13] Steven L Franconeri. The nature and status of visual resources. *Oxford handbook of cognitive psychology*, 8481:147–162, 2013.

[Fri08] Michael Friendly. A brief history of data visualization. In *Handbook of data visualization*, pages 15–56. Springer, 2008.

[Fur86] George W Furnas. *Generalized fisheye views*, volume 17. ACM, 1986.

[FWR99] Ying-Huey Fua, Matthew O Ward, and Elke A Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *Proceedings of the conference on Visualization'99: celebrating ten years*, pages 43–50. IEEE Computer Society Press, 1999.

[Gal07] Wilbert O Galitz. *The essential guide to user interface design: an introduction to GUI design principles and techniques*. John Wiley & Sons, 2007.

[GAW+11] Michael Gleicher, Danielle Albers, Rick Walker, Ilir Jusufi, Charles D Hansen, and Jonathan C Roberts. Visual comparison for information visualization. *Information Visualization*, 10(4):289–309, 2011.

[GEM+13] Prashant Goswami, Fatih Erol, Rahul Mukhi, Renato Pajarola, and Enrico Gobbetti. An efficient multi-resolution framework for high quality interactive rendering of massive point clouds using multi-way kd-trees. *The Visual Computer*, 29(1):69–83, 2013.

[GP01] Nahum Gershon and Ward Page. What storytelling can do for information visualization. *Communications of the ACM*, 44(8):31–37, 2001.

[Han06] Pat Hanrahan. Vizql: a language for query, analysis and visualization. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 721–721. ACM, 2006.

[ID90] Alfred Inselberg and Bernard Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of the 1st conference on Visualization'90*, pages 361–378. IEEE Computer Society Press, 1990.

[Ins97] Alfred Inselberg. Multidimensional detective. In *Proceedings of VIZ'97: Visualization Conference, Information Visualization Symposium and Parallel Rendering Symposium*, pages 100–107. IEEE, 1997.

[JMF99] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[Jol02] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[KAF+08] Daniel Keim, Gennady Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer, and Guy Melançon. Visual analytics: Definition, process, and challenges. In *Information visualization*, pages 154–175. Springer, 2008.

[KDHL08] Robert Kosara, Fritz Drury, Lars Erik Holmquist, and David H Laidlaw. Visualization criticism. *IEEE Computer Graphics and Applications*, 28(3), 2008.

[Kei97] Daniel A Keim. Visual techniques for exploring databases. In *Knowledge Discovery in Databases (KDD'97)*, 1997.

[Kei02] Daniel A Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, 2002.

[Kei05] Daniel A Keim. Scaling visual analytics to very large data sets. In *Workshop on Visual Analytics, Darmstadt*, pages 114–125, 2005.

[KKE10] Daniel A Keim, Jörn Kohlhammer, and Geoffrey Ellis. Mastering the information age: solving problems with visual analytics. *Eurographics Association*, 2010.

[KMSZ06] Daniel A Keim, Florian Mansmann, Jörn Schneidewind, and Hartmut Ziegler. Challenges in visual data analysis. In *Tenth International Conference on Information Visualisation (IV'06)*, pages 9–16. IEEE, 2006.

[Koh90] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[LBI+12] Heidi Lam, Enrico Bertini, Petra Isenberg, Catherine Plaisant, and Sheelagh Carpendale. Empirical studies in information visualization: Seven scenarios. *IEEE Transactions on Visualization and Computer Graphics*, 18(9):1520–1536, 2012.

[LGS+14] 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.

[LH11] O Daae Lampe and Helwig Hauser. Curve density estimates. In *Computer Graphics Forum*, volume 30, pages 633–642. Wiley Online Library, 2011.

[LMW+16] 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*, 23(3):1249–1268, 2016.

[LSS09] Zhicheng Liu, John Stasko, and Timothy Sullivan. Selltrend: Inter-attribute visual analysis of temporal transaction data. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1025–1032, 2009.

[Man07] Prem S Mann. *Introductory statistics*. John Wiley & Sons, 2007.

[Mea92] A Mead. Review of the development of multidimensional scaling methods. *The Statistician*, pages 27–39, 1992.

[MHS07] Jock Mackinlay, Pat Hanrahan, and Chris Stolte. Show me: Automatic presentation for visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 13(6), 2007.

[Moo65] Gordon E Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8), 1965.

[MP13] 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.

[MR10] Michael J McGuffin and Jean-Marc Robert. Quantifying the space-efficiency of 2d graphical representations of trees. *Information Visualization*, 9(2):115–140, 2010.

[Mun14] Tamara Munzner. *Visualization Analysis and Design*. CRC Press, 2014.

[MWK14] Mahsa Mirzargar, Ross T Whitaker, and Robert M Kirby. Curve boxplot: Generalization of boxplot for ensembles of curves. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2654–2663, 2014.

[Net15] UK Power Networks. Smartmeter energy consumption data in london households. `https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households`, 2015. Accessed: 9th October, 2019.

[oEU14] Council of European Union. Council regulation (EU) no 189/2014. http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=SWD:2014:189:FIN, 2014. Accessed: 9th October, 2019.

[PAA+87] Stephen M Pizer, E Philip Amburn, John D Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart ter Haar Romeny, John B Zimmerman, and Karel Zuiderveld. Adaptive histogram equalization and its variations. *Computer vision, graphics, and image processing*, 39(3):355–368, 1987.

[PBH08] 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.

[Pea95] Karl Pearson. Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58:240–242, 1895.

[Pen05] Brian S Penn. Using self-organizing maps to visualize high-dimensional data. *Computers & Geosciences*, 31(5):531–544, 2005.

[Pfa15] David Pfahler. In-place interaction in dashboards. Technical report, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Favoritenstrasse 9-11/186, A-1040 Vienna, Austria, October 2015. Accessed: 9th October, 2019.

[Pir11] Harald Piringer. *Large data scalability in interactive visual analysis*. PhD thesis, Piringer, 2011.

[PTMB09] Harald Piringer, Christian Tominski, Philipp Muigg, and Wolfgang Berger. A multi-threading architecture to support interactive visual exploration. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1113–1120, 2009.

[RBH+12] Roland Radtke, Aaron M Butcher, Jensen M Harris, Catherine R Morrow, and Jesse Clay Satterfield. User interface for displaying selectable software functionality controls that are contextually relevant to a selected object, February 14 2012. US Patent 8,117,542.

[Ril04] Jenn Riley. *Understanding metadata*. National Information Standards Organization, 2004.

[RW73] Horst WJ Rittel and Melvin M Webber. Dilemmas in a general theory of planning. *Policy sciences*, 4(2):155–169, 1973.

[Sch] Schema.org. Schemas for structured data on the internet. Schema.org. Accessed: 9th October, 2019.

[Shn96] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE symposium on visual languages*, pages 336–343. IEEE, 1996.

[SMM12] Michael Sedlmair, Miriah Meyer, and Tamara Munzner. Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2431–2440, 2012.

[SS05] Jinwook Seo and Ben Shneiderman. A rank-by-feature framework for interactive exploration of multidimensional data. *Information visualization*, 4(2):96–113, 2005.

[STH02] 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.

[Sut68] Ivan E Sutherland. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pages 757–764. ACM, 1968.

[SVV99] Cheri Speier, Joseph S Valacich, and Iris Vessey. The influence of task interruption on individual decision making: An information overload perspective. *Decision Sciences*, 30(2):337–360, 1999.

[TB66] Robert C Tryon and Daniel E Bailey. The BC TRY computer system of cluster and factor analysis. *Multivariate Behavioral Research*, 1(1):95–111, 1966.

[TC06] James J Thomas and Kristin A Cook. A visual analytics agenda. *IEEE computer graphics and applications*, 26(1):10–13, 2006.

[Tho05] James J Thomas. *Illuminating the path:[the research and development agenda for visual analytics]*. IEEE Computer Society, 2005.

[TLLH12] 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.

[TM04] Melanie Tory and Torsten Möller. Human factors in visualization research. *IEEE Transactions on Visualization and Computer Graphics*, 10(1):72–84, 2004.

[TSDS96] Lisa Tweedie, Robert Spence, Huw Dawkes, and Hus Su. Externalising abstract mathematical models. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 406–ff. ACM, 1996.

[Tuf06] Edward R Tufte. *Beautiful evidence*, volume 1. Graphics Press Cheshire, CT, 2006.

[WHA07] Wesley Willett, Jeffrey Heer, and Maneesh Agrawala. Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1129–1136, 2007.

[WK06] Martin Wattenberg and Jesse Kriss. Designing for social data analysis. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):549–557, 2006.

[YaKS07] Ji Soo Yi, Youn ah Kang, and John Stasko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1224–1231, 2007.

[YPWR03] 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 *IEEE Symposium on Information Visualization 2003 (IEEE Cat. No. 03TH8714)*, pages 105–112. IEEE, 2003.

[Yu77] Chong Ho Yu. Exploratory data analysis. *Methods*, 2:131–160, 1977.

[YWR02] Jing Yang, Matthew O Ward, and Elke A Rundensteiner. Visual hierarchical dimension reduction for exploration of high dimensional datasets. In *Eurographics/IEEE TCVG Symposium on Visualization*, pages 19–28, 2002.

[YWR03] Jing Yang, Matthew O Ward, and Elke A Rundensteiner. Interactive hierarchical displays: a general framework for visualization and exploration of large multivariate data sets. *Computers & Graphics*, 27(2):265–283, 2003.

[ZFY16] Kaile Zhou, Chao Fu, and Shanlin Yang. Big data driven smart energy management: From big data to big insights. *Renewable and Sustainable Energy Reviews*, 56:215–225, 2016.